

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МІСЬКОГО ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА**

**Пояснювальна записка  
до кваліфікаційної роботи бакалавра**

на тему: Розробка сайту підбору спортивного одягу

Виконав: студент 4 курсу, групи КН 2022-1  
спеціальності 122 Комп'ютерні науки  
(шифр і назва спеціальності)



Дмитро ШУТЕСВ  
(прізвище та ініціали)

Керівник: Анатолій ЛИТВИНОВ  
(прізвище та ініціали)



Рецензент: Юрій ПАХОМОВ  
(прізвище та ініціали)



м. Харків – 2026 рік

Харківський національний університет міського господарства імені О. М. Бекетова

(повне найменування закладу вищої освіти)

Навчально-науковий Інститут енергетичної, інформаційної

та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри КНтаІТ

  
Марина НОВОЖИЛОВА

« 24 »    червня    2026 року

**З А В Д А Н Н Я**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ**

Шутєєва Дмитра Сергійовича

**(прізвище, ім'я, по батькові)**

1. Тема роботи Розробка сайту підбору спортивного одягу

керівник роботи доктор техн. наук, проф. Литвинов Анатолій Леонідович

( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «22» травня 2026 р. № 440-03

2. Термін подання здобувачем роботи 15 червня 2026 р.

3. Вихідні дані до роботи Рекомендації щодо реалізації застосування для підбору спортивного одягу в залежності від антропометричних даних

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

– дослідити специфіку ринку спортивного одягу та виявити ключові критерії вибору товарів користувачами;

– зробити порівняльний аналіз існуючих аналогів та вебплатформ-конкурентів, визначити їхні переваги та недоліки;

– розробити математичне та алгоритмічне забезпечення для підсистеми фільтрації та персоналізованого підбору одягу;

– спроєктувати інфологічну та даталогічну моделі бази даних;





– протестувати розроблений програмний продукт, проаналізувати результати;

– розглянути питання, пов'язані із організацією та забезпеченням охорони праці як важливої складової безпеки працівників.

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

Презентація – 16 аркушів

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Литвинов А.Л., проф. каф КНтаІТ	11.05.2026 	23.05.2026
Розділ II	Литвинов А.Л., проф. каф КНтаІТ	24.05.2026 	02.06.2026
Розділ III	Литвинов А.Л., проф. каф КНтаІТ	03.06.2026 	10.06.2026
Розділ IV	Малишева В.В., доцент каф. ОПтаБЖ 	11.06.2026	14.06.2026

7. Дата видачі завдання 11.05. 2026 р.

## КАЛЕНДАРНИЙ ПЛАН

/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
	Вибір теми кваліфікаційної роботи	11.05.2026	Виконано
	Затвердження тем, наукових керівників, завдань та календарного плану підготовки кваліфікаційної роботи	15.05.2026	Виконано
	Написання I розділу	23.05.2026	Виконано
	Написання II розділу	02.06.2026	Виконано
	Написання III розділу	10.06.2026	Виконано
	Написання IV розділу	14.06.2026	Виконано
	Подання дипломної роботи керівнику	15.06.2026	Виконано
	Робота по усуненню зауважень керівника, уточнення і доповнення практичного матеріалу, оформлення додатків до роботи	16.06.2026	Виконано
	Подання доопрацьованого варіанту роботи керівнику	16.06.2026	Виконано
0	Захист матеріалів кваліфікаційної роботи на засіданні кафедри	18.06.2026	Виконано
1	Офіційний захист матеріалів кваліфікаційної роботи на засіданні Державної екзаменаційної комісії	25.06.2026	Виконано



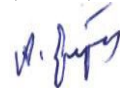
Студент

(підпис)

Дмитро ШУТЕСВ

(прізвище та ініціали)

Керівник  
роботи



(підпис)

Анатолій ЛИТВИНОВ

(прізвище та ініціали)"

## АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка кваліфікаційної роботи бакалавра студента групи КН 2022-1 спеціальності 122 Комп'ютерні науки Шутеева Дмитра Сергійовича за темою «Розробка сайту підбору спортивного одягу» складається з 4 розділів, містить 2 рисунка, 11 таблиць, 21 джерел.

Дистанційна торгівля спортивним одягом стрімко розвивається, проте ключовою проблемою галузі залишається високий рівень повернення товарів (до 30–40%) через невідповідність розміру. Класичні таблиці виробників не враховують індивідуальні особливості типів фігур та динаміку рухів під час тренувань. Створення інформаційної системи, що інтелектуально аналізує багатовимірні метрики людського тіла та прогнозує точність посадки, є актуальним науково-практичним завданням.

Об'єкт дослідження – процеси індивідуального підбору та рекомендації товарів в інтернет-магазинах спортивного одягу на основі фізичних замірів кінцевих користувачів.

Предмет дослідження – математичні моделі, алгоритми обробки антропометричних даних, архітектурні патерни та програмні засоби реалізації рекомендаційного вебкомплексу.

Мета роботи – проектування та програмна реалізація високонавантаженої клієнт-серверної інформаційної системи, яка забезпечує точне визначення індексу сумісності одягу (Fit Score) для мінімізації помилок підбору.

Методи дослідження. Для вирішення поставлених завдань використано: методи системного аналізу предметної області, теорію множин та реляційну алгебру (для проектування баз даних), а також методи лінійної алгебри та метричних просторів, зокрема модифікований алгоритм зваженої евклідової відстані (для розрахунку сумісності розмірів).

Отримані результати та їх новизна:

1. Розроблено математичну модель оцінки посадки одягу, яка на відміну від існуючих аналогів розглядає тіло людини як точку в багатовимірному просторі ознак та використовує систему динамічних вагових коефіцієнтів, що адаптуються під конкретну категорію одягу (топи, легінси тощо).

2. Спроековано розподілену безстанову (Stateless) архітектуру системи на базі REST API, яка ізолює інтерфейс від логіки обчислень, забезпечуючи високу відмовостійкість.

3. Реалізовано базу даних у СКБД PostgreSQL, нормалізовану до 3NF та оптимізовану B-Tree індексами, що забезпечило швидкість пошуку матриць розмірів на рівні .

4. Створено реактивний SPA-застосунок на базі React (Vite) та Node.js (Express.js), обчислювальне ядро якого успішно пройшло тестування на стійкість до аномальних вхідних даних та продемонструвало швидкість відгуку сервера в межах 50 мс.

5. Розглянуто значення оцінки професійних ризиків як важливого елементу системи управління охороною праці

Практичне значення. Створений програмний комплекс є повністю готовим до інтеграції модулем для діючих e-commerce платформ. Його впровадження дозволяє знизити рівень повернень товарів, підвищити лояльність клієнтів та автоматизувати обробку замовлень.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, РЕКОМЕНДАЦІЙНА СИСТЕМА, ЕЛЕКТРОННА КОМЕРЦІЯ, АНТРОПОМЕТРІЯ, ЗВАЖЕНА ЕВКЛІДОВА ВІДСТАНЬ, ІНДЕКС ПОСАДКИ (FIT SCORE), NODE.JS, REACT, POSTGRESQL, REST API, SPA.

## ANNOTATION

Structure and scope of work. The explanatory note of the bachelor's qualification work of the student of the group CS 2022-1 specialty 122 Computer Science Shuteiev Dmytro Serhiyovich on the topic " Development of a sportswear selection website " consists of 4 sections, contains 2 figures, 11 tables, 21 sources.

E-commerce in the sportswear segment is developing rapidly, yet the key challenge remains the high product return rate (up to 30–40%) due to sizing inaccuracies. Traditional static size charts provided by brands fail to account for individual body shape variations and movement dynamics during physical activities. Developing an information system that intelligently analyzes multidimensional human body metrics and predicts fit compatibility is a highly relevant scientific and practical task.

Object of research is the process of personalized product selection and recommendation in apparel e-commerce based on end-user physical measurements.

Subject of research comprises mathematical models, algorithms for processing anthropometric data, architectural patterns, and software tools used to build the recommendation web platform.

The aim of the work is the design and software implementation of a high-concurrency client-server information system that accurately calculates the garment compatibility index (Fit Score) to minimize selection errors.

Research methods. To solve the formulated problems, the following methods were applied: systems analysis for domain modeling, set theory and relational algebra for database design, and linear algebra combined with metric space theory, specifically a modified weighted Euclidean distance algorithm, for size compatibility computations.

Results obtained and their novelty:

1. A mathematical model for garment fit assessment has been developed. Unlike existing solutions, it treats the human body as a point in a multidimensional

feature space and applies dynamic weight coefficients tailored to specific clothing categories (e.g., sport tops vs leggings).

2. A decoupled stateless client-server architecture based on REST API has been engineered, isolating the representation layer from the business logic to ensure high fault tolerance and horizontal scalability.

3. The persistence layer has been deployed using PostgreSQL, normalized to 3NF, and optimized via B-Tree indexing, reducing the time complexity of size matrix scans to a logarithmic level .

4. A reactive SPA web application has been implemented using React (Vite) and Node.js (Express.js). The core computational engine successfully passed simulation testing against anomalous input metrics and achieved a server latency under 50 ms.

5. The importance of occupational risk assessment as an important element of the occupational health and safety management system is considered.

Practical significance. The developed software system is a fully functional component ready for seamless integration into existing e-commerce platforms. Its deployment minimizes product return rates, increases customer retention, and automates order processing.

Keywords: INFORMATION SYSTEM, RECOMMENDATION ENGINE, E-COMMERCE, ANTHROPOMETRY, WEIGHTED EUCLIDEAN DISTANCE, FIT SCORE, NODE.JS, REACT, POSTGRESQL, REST API, SPA.

## ЗМІСТ

АНОТАЦІЯ4

ANNOTATION6

ВСТУП10

РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ12

1.1. Вступна частина: обґрунтування актуальності, мета та задачі дослідження12

1.2. Об'єкт, предмет та методи дослідження14

1.3. Детальний аналіз предметної області та інженерна декомпозиція даних15

1.3.1. Особливості класифікації та критерії підбору спортивного одягу15

1.3.2. Математична та інженерна проблема неузгодженості розмірних сіток15

1.4. Порівняльний аналіз існуючих аналогів та архітектурних підходів конкурентів17

1.4.1. Аналіз інтерфейсних та алгоритмічних рішень конкурентів17

1.4.2 Порівняльна таблиця аналогів19

1.5. Постановка задачі на розробку сайту20

1.5.1. Функціональні вимоги до вебсайту20

1.5.2. Нефункціональні (технічні) вимоги21

ВИСНОВКИ ДО РОЗДІЛУ 121

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

СИСТЕМИ ПІДБОРУ СПОРТИВНОГО ОДЯГУ23

2.1. Аналіз предметної області та формалізація інформаційних потоків23

2.1.1. Вхідні дані системи24

2.1.2. Вихідні дані системи26

2.2. Проектування системи та структурне моделювання28

2.2.1. Проектування бази даних28

2.2.2. Побудова об'єктно-орієнтованої моделі (UML)32

2.3. Математичне та алгоритмічне забезпечення рекомендаційного модуля36

2.3.1. Математична формалізація антропометричного простору37

2.3.2. Алгоритм розрахунку відхилень та функція Fit Scoring38

2.3.4. Текстовий опис та алгоритмічний псевдокод модуля підбору40

ВИСНОВКИ ДО РОЗДІЛУ 243

## РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ45

### 3.1 Засоби розробки45

3.1.1 Обґрунтування середовища виконання та вебфреймворку серверної частини (Backend)46

3.1.2 Обґрунтування бібліотеки побудови користувацького інтерфейсу (Frontend)47

3.1.3 Обґрунтування вибору системи керування базами даних (СКБД)49

### 3.2 Вимоги до технічного та програмного забезпечення50

3.2.1 Вимоги до програмного забезпечення (Software Requirements)50

3.2.2 Вимоги до технічного забезпечення (Hardware Requirements)52

3.3.1 Організація логічної структури сховища даних53

3.3.2 Проєктування конвеєра обробки запитів (API Architecture)54

3.3.3 Програмна декомпозиція алгоритму інтелектуального підбору55

3.3.4 Логіка оркестрації та формування рекомендаційної відповіді56

3.3.5 Клієнтська архітектура та механізми реактивності інтерфейсу57

3.3.6 Ключові фрагменти програмної реалізації (Мінімальний базовий код)58

3.3.7 Оцінка ефективності та стабільності програмних модулів60

## ВИСНОВКИ ДО РОЗДІЛУ 361

## РОЗДІЛ 4 ОХОРОНА ПРАЦІ64

4.1 Організаційно-правові основи забезпечення безпеки праці64

4.2 Характеристика об'єкта та виявлення потенційних небезпек66

4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проєктування та розробка заходів щодо їх попередження70

Висновки до розділу75

## ВИСНОВКИ76

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ78

## ВСТУП

Вибір спортивного одягу – це баланс між комфортом, функціональністю та естетикою. Правильне екіпірування захищає від перегріву чи переохолодження, відводить вологу та дозволяє тілу дихати, тим самим підвищуючи ефективність ваших тренувань. Ось ключові критерії, які варто враховувати при виборі екіпірування.

### 1. Матеріал та його функції

Основа будь-якого спортивного образу – це тканина. На відміну від повсякденного гардероба, тут синтетичні матеріали часто перевершують натуральні (наприклад, бавовна швидко вбирає піт, стає важкою і довго сохне, створюючи дискомфорт).

Високоякісні спортивні тканини (поліестер з домішками еластану) розроблені так, щоб виводити піт на поверхню, де він швидко випаровується. Це допомагає зберігати тіло сухим навіть під час інтенсивного кардіо.

Одяг повинен «дихати», забезпечуючи оптимальну циркуляцію повітря. Шукайте моделі зі спеціальними сітчастими вставками у зонах підвищеного потовиділення (пахви, спина).

### 2. Призначення та вид спорту

Не існує універсального спортивного костюма для всіх занять, адже кожен вид навантажень має свої вимоги:

Для йоги та стретчингу обирайте безшовні, м'які та еластичні речі, які тягнуться у чотирьох напрямках і не сковують рухів. Віддавайте перевагу високій посадці лосин.

Для фітнесу та тренажерного залу головне – компресія та підтримка. Дівчатам обов'язково знадобиться спеціальний спортивний топ-бра, який фіксує груди.

Для бігу на відкритому повітрі важлива легкість та наявність світловідбивних елементів для безпеки у темну пору доби.

Для силових тренувань стане в нагоді рашгард – спеціальна футболка, яка захищає шкіру від тертя об спортивні снаряди і зберігає тепло м'язів.

### 3. Крій та посадка

Спортивний одяг має сидіти ідеально: не бути занадто вільним, щоб не чіплятися за тренажери, і не занадто тісним, щоб не порушувати кровообіг.

Звертайте увагу на шви: вони мають бути плоскими або гладкими, щоб уникнути натирання шкіри під час активних рухів.

Обирайте речі з ергономічним кроєм, який повторює анатомічну форму тіла.

### 4. Врахування сезону

Літо: Вибирайте ультратонкі матеріали з технологіями охолодження та захистом від ультрафіолету (UPF).

Зіма/Осінь: Використовуйте принцип багат шаровості. Перший шар – термобілизна, яка відводить вологу. Другий (фліс) – зберігає тепло. Третій – вітровка або мембранна куртка, що захищає від дощу та вітру.

### 5. Естетика та самовираження

Хоча функціональність є першорядною, дизайн одягу також відіграє важливу роль. Стильний спортивний костюм підвищує самооцінку, стимулює до нових звершень і створює правильний психологічний настрій перед тренуванням.

Підсумовуючи, можна сказати, що ідеальний спортивний одяг – це той, про який ви забуваєте одразу після початку тренування. Він стає другою шкірою, дозволяючи зосередитися виключно на досягненні ваших спортивних цілей.

Правильний вибір спортивного одягу є важливим чинником шляху до здорового способу життя.

## РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

Сучасний етап розвитку інформаційного суспільства характеризується стрімкою цифровізацією всіх сфер людської діяльності, серед яких електронна комерція (e-commerce) посідає одне з провідних місць. Трансформація споживчої поведінки, зумовлена глобальними змінами та розвитком вебтехнологій, вимагає від розробників створення не просто статичних інтернет-магазинів, а високоадаптивних, інтелектуальних вебплатформ.

Особливо гостро ця проблема постає в сегменті роздрібної торгівлі спортивним одягом та екіпіруванням. Специфіка даної предметної області полягає в тому, що вибір спортивного одягу залежить від великої кількості антропометричних параметрів користувача, виду спорту, сезонності, інтенсивності тренувань та технологічних властивостей матеріалів.

Розробка спеціалізованого вебресурсу, що інтегрує алгоритми автоматизованого підбору одягу, є актуальним науково-практичним завданням, розв'язанню якого присвячено цей розділ.

1.1. Вступна частина: обґрунтування актуальності, мета та задачі дослідження

### Актуальність теми

Актуальність розробки сайту для підбору спортивного одягу обумовлена високим відсотком повернення товарів у сфері онлайн-ритейлу через невідповідність розміру або функціональних характеристик очікуванням покупця. За статистичними даними світового ринку e-commerce, понад 20-30% замовленого одягу повертається назад, що завдає суттєвих збитків бізнесу та збільшує логістичне навантаження.

Впровадження засобів інтелектуальної фільтрації та рекомендаційних алгоритмів на основі введених даних користувача дозволяє оптимізувати процес вибору, знизити ризик помилки та підвищити лояльність клієнтів. З погляду комп'ютерних наук, проектування такої системи вимагає дослідження методів структурування даних, розробки ефективних алгоритмів пошуку та фільтрації, а також створення гнучкої архітектури вебдодатка[1].

#### Мета дослідження

Метою бакалаврської роботи є проектування, програмна реалізація та тестування інтелектуальної вебплатформи (сайту) для підбору спортивного одягу, яка забезпечує користувачам персоналізований досвід вибору товарів на основі аналізу їхніх фізичних параметрів, цільового призначення та технологічних вимог до екіпірування.

#### Задачі дослідження

Для досягнення поставленої мети необхідно вирішити такі задачі:

- Провести системний аналіз предметної області, дослідити специфіку ринку спортивного одягу та виявити ключові критерії вибору товарів користувачами.
- Виконати порівняльний аналіз існуючих аналогів та вебплатформ-конкурентів, визначити їхні переваги та недоліки.
- Обґрунтувати вибір архітектурних рішень, методологій проектування та сучасного стеку технологій (фронтенд, бекенд, СКБД) для реалізації проекту.
- Розробити математичне та алгоритмічне забезпечення для підсистеми фільтрації та персоналізованого підбору одягу.
- Спроекувати інфологічну та даталогічну моделі бази даних.
- Реалізувати програмний комплекс (вебсайт) із розділенням на клієнтську та серверну частини.
- Провести тестування розробленого програмного продукту, проаналізувати результати та оцінити ефективність його функціонування.

## 1.2. Об'єкт, предмет та методи дослідження

Об'єктом дослідження є процес взаємодії користувача з вебплатформами електронної комерції в сегменті роздрібною торгівлі, зокрема процеси пошуку, фільтрації та прийняття рішень під час вибору спеціалізованих товарів (спортивного одягу).

Предметом дослідження є моделі, методи, алгоритми та інструментальні засоби розробки вебсайтів, алгоритми фільтрації даних та формування персоналізованих рекомендацій щодо підбору спортивного одягу залежно від вхідних параметрів користувача.

Методи дослідження. Для розв'язання поставлених задач у роботі використовуються такі методи:

- Методи системного аналізу – для дослідження предметної області, структурування вимог до системи та декомпозиції загальної задачі розробки на підзадачі.
- Методології об'єктно-орієнтованого аналізу та проектування (ООАР) і графічне моделювання за допомогою мови UML – для побудови архітектурних діаграм та опису бізнес-процесів взаємодії.
- Теорія реляційних баз даних – для проектування структури збереження інформації про товари, користувачів та замовлення.
- Методи вебпрограмування (моделі MVC/MVVM) – для безпосереднього створення клієнтської та серверної частин застосунку.
- Методи тестування програмного забезпечення (модульне, інтеграційне, функціональне тестування) – для перевірки працездатності та надійності системи.

### 1.3. Детальний аналіз предметної області та інженерна декомпозиція даних

Предметна область охоплює дві основні складові: технологічні та ергономічні особливості спортивного одягу, а також механізми взаємодії користувача із системами пошуку товарів.

#### 1.3.1. Особливості класифікації та критерії підбору спортивного одягу

На відміну від повсякденного одягу, спортивне екіпірування підбирається за жорсткими критеріями, які можна розділити на кілька категорій:

1. Вид фізичної активності (вид спорту): Біг, фітнес/кросфіт, бодібілдинг, туризм/альпінізм, командні ігри. Кожен вид спорту вимагає специфічного крою та властивостей (наприклад, компресія для важкої атлетики або вітрозахист для бігу взимку).

2. Умови експлуатації (сезонність та локація): Зали закритого типу, відкритий простір (outdoor), літній період, демісезон, зимовий період. Від цього залежить терморегуляція одягу.

3. Анатомічні та антропометричні параметри: Зріст, обхват грудей, талії, стегон, довжина кінцівок.

4. Склад матеріалу та технології: Використання мембранних тканин, технологій відведення вологи (наприклад, Dri-FIT, Climacool), рівень еластичності та компресії.

#### 1.3.2. Математична та інженерна проблема неузгодженості розмірних сіток

Ключова проблема автоматизації підбору одягу полягає в тому, що умовні позначення розмірів (S, M, L, XL) є абстракцією. Кожен транснаціональний бренд використовує власні лекала та внутрішні стандарти

(таблиці відповідності метричних параметрів символічним кодам). Більше того, географічний регіон виробництва (європейська, американська чи азійська сітка) вносить суттєві похибки.

Розглянемо практичний інженерний приклад: порівняння чоловічих розмірних сіток для категорії «футболки / худі» між двома лідерами ринку – Nike та Adidas.

Таблиця 1.1 – Порівняльний аналіз метричних параметрів брендів (Чоловіки, верх)

Символьний розмір	Метричні параметри Nike (обхват грудей / талії), см	Метричні параметри Adidas (обхват грудей / талії), см	Дельта (розбіжність), см
S	88–96 / 73–81	88–92 / 76–82	-4 (груди) / +1 (талія)
M	96–104 / 81–89	93–100 / 83–90	-4 (груди) / +2 (талія)
L	104–112 / 89–97	101–108 / 91–99	-3 (груди) / +2 (талія)
XL	112–124 / 97–109	109–118 / 100–109	-5 (груди) / 0 (талія)

Аналіз даних таблиці показує, що якщо користувач має обхват грудей 94 см, то за класифікацією Nike він потрапляє в нижню межу розміру M, тоді як за класифікацією Adidas він знаходиться у верхній межі розміру M. Проте,

якщо обхват грудей становить 102 см, Nike класифікує це як розмір M, а Adidas – як розмір L.

З погляду проектування баз даних та алгоритмів, система не може оперувати статичними текстовими значеннями типу "розмір = M". Необхідно реалізувати інтервальний пошук. У базі даних кожна одиниця товару повинна посилатися на таблицю відповідностей, де розмір задається діапазоном нерівностей:

$$\text{Size}(\min) \leq \text{UserParameter} \leq \text{Size}(\max)$$

Алгоритм підбору повинен приймати вектор антропометричних даних користувача  $U = \{g, t, s\}$ , де  $g$  – обхват грудей,  $t$  – обхват талії,  $s$  – обхват стегон, та динамічно обчислювати перетин множин доступних товарів, які повністю задовольняють нерівності для конкретного обраного бренду.

#### 1.4. Порівняльний аналіз існуючих аналогів та архітектурних підходів конкурентів

Для визначення конкурентних переваг розроблюваного сайту та формування вимог до його функціональності проведено аналіз трьох типів систем: глобальні маркетплейси, інтернет-магазини монобрендів та інтеграційні B2B-віджети.[1,2]

##### 1.4.1. Аналіз інтерфейсних та алгоритмічних рішень конкурентів

###### 1. Глобальні маркетплейси (на прикладі Rozetka / Amazon)

Процес взаємодії користувача з маркетплейсом під час вибору одягу є лінійним і перевантаженим. Користувач заходить у загальну категорію, де йому пропонується ліва бічна панель із десятками фільтрів.

- Логіка інтерфейсу: Користувач має самостійно обрати галочками розміри (наприклад, "M" та "L"). Оскільки маркетплейс об'єднує тисячі різних

фабрик, вибір фільтра "М" виведе як речі європейського пошиття, так і маломірні азійські товари.

- Взаємодія з таблицею розмірів: Для уточнення користувач повинен клікнути на текстове посилання «Таблиця розмірів», яке відкриває статичне спливаюче вікно (Pop-up) з неструктурованою картинкою або HTML-таблицею. Користувач змушений утримувати в пам'яті свої заміри та візуально зіставляти їх із сіткою. Це створює високе когнітивне навантаження та призводить до помилок відмови (User Churn).

## 2. Брендіві інтернет-магазини (на прикладі Nike)

Nike використовує більш просунутий підхід до UI/UX. На сторінці картки товару інтегровано внутрішній опитувальник «Find Your Size».

- Логіка інтерфейсу: Замість сухої таблиці користувачеві пропонується модальне вікно, де послідовно (крок за крок) збираються дані: Стать -> Зріст -> Вага. На основі цих трьох параметрів внутрішній скрипт будує статистичну криву і видає рекомендацію у форматі: «Наш аналіз показує, що 78% користувачів з вашими параметрами купили розмір L».

- Недоліки рішення: Дана система спирається на усереднену вагу та зріст, що абсолютно не підходить для професійних спортсменів чи бодібілдерів (де при однаковій вазі та зрості обхват грудей та талії може кардинально відрізнятись через м'язову масу). Крім того, це рішення замкнене всередині одного бренду.

## 3. Сторонні плагіни підбору (на прикладі Fit Analytics / ASOS Style Advisor)

Це інтеграційні системи, які вбудовуються в готові сайти через JavaScript SDK.

- Логіка інтерфейсу: Користувач вводить не лише свої параметри, а й вказує бренд одягу, який він зазвичай носить, та його розмір (наприклад: «Я ношу джинси Levis у розмірі 32/32»). Алгоритм через API звертається до

глобальної хмари даних, порівнює лекала Levis із лекалами поточного товару на ASOS та видає точний розмір.

- Недоліки рішення: Дані сервіси є «чорною скринькою» (Black Box). Вони не дозволяють власнику сайту керувати алгоритмом, вимагають щомісячної плати за кожне звернення через API, сповільнюють швидкість рендерингу сторінки (через завантаження зовнішніх JS-скриптів) та мають ризики блокування в разі відсутності стабільного з'єднання із закордонними серверами.

#### 1.4.2 Порівняльна таблиця аналогів

Для наочності проведено узагальнене порівняння характеристик систем за 5-бальною шкалою (де 1 – незадовільно, 5 – відмінно).

Таблиця 1.2 – Порівняльний аналіз метричних параметрів брендів

Критерій порівняння	Глобальні маркетплейси	Брендові магазини	Сторонні сервіси підбору	Розроблювана вебплатформа
Широта асортименту (мультибрендовість)	5	2	4	4
Простота інтерфейсу для спортсмена	3	4	4	5
Автоматизований підбір за параметрами тіла	2	2	5	5

Аналіз технологічних властивостей тканин	1	5	2	4
Гнучкість налаштування та автономність	2	3	1	5

### 1.5. Постановка задачі на розробку сайту

На основі якого проведено аналіз предметної області та аналогів, сформулюємо технічні, функціональні та експлуатаційні вимоги до проектованої системи.

#### 1.5.1. Функціональні вимоги до вебсайту

Розроблювана система повинна бути розділена на три основні модулі взаємодії: підсистема неавторизованого користувача, підсистема клієнта (особистий кабінет) та підсистема адміністратора (панель керування).

##### 1. Функціонал для всіх відвідувачів:

- Перегляд каталогу товарів з можливістю сортування (за ціною, популярністю, новизною).
- Пошук товарів за ключовими словами.
- Використання розширеного фільтра (бренд, колір, сезон, матеріал, вид спорту).
- Модуль інтелектуального підбору: інтерактивна форма, де користувач вводить стать, зріст, обхват грудей/талії/стегон та вид спорту, після чого система автоматично фільтрує каталог.

##### 2. Функціонал авторизованого клієнта:

- Реєстрація та автентифікація (зокрема через токени JWT).

- Збереження параметрів свого тіла в особистому кабінеті, щоб підбір відбувався автоматично при кожному вході.

- Керування кошиком товарів та оформлення замовлення.

- Перегляд історії замовлень та їхніх статусів.

3. Функціонал адміністратора (Адмін-панель):

- Керування каталогом: додавання, редагування, видалення товарів та категорій.

- Налаштування розмірних сіток для кожного бренду (внесення діапазонів сантиметрів для кожного розміру).

- Перегляд та зміна статусів замовлень клієнтів.

#### 1.5.2. Нефункціональні (технічні) вимоги

- Продуктивність: Час відповіді сервера на стандартний пошуковий запит не повинен перевищувати 500 мс. Оптимальний час рендерингу сторінки клієнта – до 2 секунд.

- Адаптивність інтерфейсу (Responsive Design): Сайт повинен коректно відображатися та функціонувати на настільних ПК, планшетах та мобільних пристроях (екрани від 320px до 1920px).

- Безпека даних: Хешування паролів користувачів у базі даних (за допомогою алгоритму bcrypt), захист від поширених вебвразливостей (SQL-ін'єкції, XSS, CSRF).

### ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі бакалаврської роботи було виконано всебічний аналітичний огляд предметної області та здійснено постановку задачі на проектування сайту підбору спортивного одягу.

Головні результати етапу:

1. Доведено актуальність теми, яка полягає в автоматизації рутинного процесу підбору екіпірування за антропометричними даними, що мінімізує помилки покупців та фінансові втрати бізнесу від повернень.

2. На основі аналізу розмірних сіток Nike та Adidas доведено інженерну складність проблеми (наявність дельти параметрів до 4-5 см для однакових літерних кодів), що обґрунтовує необхідність розробки системи інтервального пошуку.

3. Проведено порівняльний аналіз інтерфейсної логіки та UX-сценаріїв систем-аналогів (маркетплейси, монобренди, віджети), визначено їхні слабкі сторони та сформовано вектор покращення користувацького досвіду розроблюваного сайту.

4. Сформовано вичерпний перелік функціональних та технічних вимог до системи, деталізовано ролі користувачів (гість, клієнт, адміністратор).

Отримані аналітичні дані та сформульовані вимоги виступають безпосереднім технічним завданням і фундаментом для наступного розділу роботи – проектування архітектури системи, розробки алгоритмів підбору та створення моделі бази даних.

## РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПІДБОРУ СПОРТИВНОГО ОДЯГУ

Проектування інтелектуальних систем електронної комерції вимагає побудови стійкого інформаційного фундаменту та математично обґрунтованого алгоритмічного каркаса. Ефективність рекомендаційного вебресурсу безпосередньо залежить від того, наскільки точно формалізовані інформаційні потоки, як спроектована база даних та які математичні критерії покладено в основу підсистеми прийняття рішень.

У межах цього розділу виконується повний цикл системного проектування платформи. Спочатку проводиться детальний аналіз предметної області через призму інформаційних потоків, здійснюється декомпозиція вхідних і вихідних масивів даних, які циркулюють між користувачем, клієнтською частиною (Frontend) та сервером (Backend).

Наступним кроком є структурне та об'єктно-орієнтоване моделювання: розробляється інфологічна та даталогічна схеми реляційної бази даних PostgreSQL з урахуванням вимог нормалізації, а також будується комплекс UML-діаграм, що фіксують статичну архітектуру та динамічну поведінку програмних модулів.

Кульмінацією розділу є математична формалізація задачі підбору розміру одягу в багатовимірному просторі ознак та побудова оптимізаційного алгоритму розрахунку відсотка відповідності виробів на основі зважених евклідових відстаней та інтервального аналізу.

### 2.1. Аналіз предметної області та формалізація інформаційних потоків

Будь-яка сучасна інформаційна система розглядається як комплекс процесів збору, збереження, обробки та передачі інформації. Специфіка інтернет-магазину з інтелектуальним підбором спортивного одягу полягає в

тому, що система повинна в реальному часі оперувати як слабоструктурованими даними користувача, так і жорстко типізованими специфікаціями товарів, які надаються різними світовими виробниками.

Інформаційні потоки в системі можна розділити на три основні контури:

**Контур користувача (клієнтський):** Охоплює передачу антропометричних параметрів, пошукових запитів, даних авторизації та команд на формування замовлень.

**Контур адміністратора (управлінський):** Забезпечує наповнення каталогу, модифікацію інтервальних матриць розмірних сіток, модерацію користувачів та зміну статусів логістичних операцій.

**Внутрішній контур системи (алгоритмічний):** Відображає взаємодію між сервером додатків та системою керування базами даних (СКБД), що включає генерацію SQL-запитів, валідацію схем даних та розрахунок цільових функцій відповідності.

### 2.1.1. Вхідні дані системи

Вхідні дані – це первинна інформація, яка надходить у систему із зовнішнього середовища для подальшої обробки та трансформації бізнес-логікою. У розроблюваній вебплатформі вхідні дані мають чіткий поділ залежно від ролі суб'єкта, який їх генерує.

**Антропометричний вектор користувача**

Для роботи модуля рекомендацій ключовим вхідним масивом є вектор фізичних вимірів тіла людини. На відміну від стандартних систем, які запитують лише зріст та вагу, для точного підбору спортивного одягу (особливо компресійного та анатомічного крою) необхідні чисті метричні виміри в сантиметрах. Збирання цих даних реалізується через інтерактивний інтерфейс.

Вхідні дані від користувача піддаються суворій типізації. Оскільки користувач може випадково або навмисно ввести некоректні значення, на рівні

інформаційного забезпечення закладаються граничні ліміти валідації (Constraints).

Таблиця 2.1. Розширена специфікація вхідних даних користувача (Профіль підбору)

Ідентифікатор параметра	Назва параметра в системі	Тип даних (Data Type)	Одиниця виміру / Формат	Допустимий діапазон валідації	Обов'язковість (Nullability)
user_gender	Стать користувача	String (Enum)	Текст ("male", "female")	Фіксований набір	NOT NULL
user_height	Фізичний зріст	Integer	Сантиметри (см)	120 – 220	NOT NULL
chest_circ	Обхват грудної клітки	Integer	Сантиметри (см)	70 – 150	NOT NULL
waist_circ	Обхват талії	Integer	Сантиметри (см)	50 – 140	NOT NULL
hip_circ	Обхват стегон	Integer	Сантиметри (см)	70 – 160	NOT NULL
sport_tag	Цільовий вид спорту	String (Enum)	Ідентифікатор типу	Словник категорій	NOT NULL
compression_ref	Бажаний рівень посадки	String (Enum)	Текст ("loose", "tight")	Фіксований набір	OPTIONAL
brand_filter	Пріоритетний бренд	String	Системний ID бренду	Наявні в БД або "all"	NOT NULL

#### Вхідні дані адміністратора (Контентне наповнення)

Цей масив даних формується на етапі адміністрування та завантажується в систему через спеціалізовані екранні форми адмін-панелі. Він містить

інформацію про товари, яка згодом виступає еталоном для порівняння з параметрами користувача.

Таблиця 2.2. Специфікація вхідних даних адміністрування товарів та розмірних сіток

Назва інформаційного блоку	Складові параметри	Тип даних у системі	Опис інженерного значення параметра
Картка товару	title	String	Комерційна назва, текстовий опис технологій, унікальний артикул виробника, базова вартість одиниці товару.
	description	Text	
	sku_code	String (Unique)	
	base_price	Decimal(10,2)	
Технологічна матриця	fabric_composition	String	Склад тканини, коефіцієнт розтяжності матеріалу (від 1.0 до 1.5), рівень вологовідведення.
	elasticity_coef	Float	
	moisture_rate	Integer	
Метрична сітка бренду	size_code	String	Літерний код розміру (напр. "L"), числові інтервали в сантиметрах, які відповідають цьому коду у конкретного бренду.
	chest_range	Int_Range (min/max)	
	waist_range	Int_Range (min/max)	
	hip_range	Int_Range (min/max)	

### 2.1.2. Вихідні дані системи

Вихідні дані системи є результатом виконання бізнес-логіки сервера над вхідними масивами. Вони трансформуються в інформаційні пакети, що надсилаються клієнту у форматі JSON для рендерингу на фронтенді, або записуються в лог-файли та аналітичні таблиці бази даних.

### Рекомендаційний пакет видачі каталогу

Головним вихідним потоком інформації для кінцевого користувача є динамічно сформована сторінка каталогу товарів, де кожна позиція забезпечена індивідуальним розрахунковим показником відповідності.

Таблиця 2.3. Структура вихідного інформаційного пакета каталогу товарів

Назва вихідного поля JSON	Тип даних у кодї	Призначення поля в користувацькому інтерфейсі	Приклад вихідного значення
product_id	Integer	Внутрішній ідентифікатор для переходу на картку	4052
brand_name	String	Відображення торговельної марки	"Nike"
model_title	String	Комерційна назва виробу	"Pro Dri-FIT Compression Top"
calculated_size	String	Рекомендований буквенний розмір для користувача	"M"
fit_accuracy_score	Float / Percent	Обчислений відсоток точності посадки на тіло	94.2%
visual_alert_level	String	Колірний маркер для UI (зелений, жовтий, червоний)	"success_green"
final_price	Decimal	Вартість з урахуванням поточних знижок системи	1850.00
stock_status	Boolean	Наявність обчисленого розміру на складі магазину	true

Вихідні дані звітів та транзакцій

Для підсистеми адміністратора та фінансового контуру вихідними даними є:

Документ замовлення: JSON-структура та друкowana форма накладної, що містить ідентифікатор клієнта, список обраних товарів із зафіксованими розмірами, фінальну суму та логістичні дані доставки.

Аналітичний зріст відмов: Звіт, який фіксує випадки, коли користувачі з певними антропометричними параметрами не змогли підібрати одяг через відсутність відповідних інтервалів у базі даних (вихідний потік для формування замовлень у постачальників).

## 2.2. Проектування системи та структурне моделювання

Архітектура вебплатформи проектується на основі сучасних інженерних шаблонів. Основним архітектурним патерном обрано Розділену архітектуру (Layered Architecture) у поєднанні з шаблоном MVC (Model-View-Controller) для логічного відокремлення інтерфейсу користувача від бізнес-логіки та збереження даних.

Взаємодія між клієнтською частиною (React) та серверною (Node.js/Express) відбувається асинхронно через RESTful API за безстановим протоколом (Stateless HTTP), де засобом автентифікації виступають захищені токени JWT (JSON Web Tokens).

### 2.2.1. Проектування бази даних

Інформаційне ядро системи спирається на реляційну систему керування базами даних PostgreSQL. Проектування бази даних включає два етапи: побудову абстрактної концептуальної (інфологічної) моделі та розробку фізичної (дatalogічної) схеми таблиць, приведеної до третьої нормальної форми (3NF) для забезпечення реляційної цілісності та мінімізації надликовості інформації.[4]

Інфологічна модель даних (Сутності та зв'язки)

В основі інфологічної моделі лежить виділення ключових сутностей предметної області та визначення кардинальності зв'язків між ними:

1. Сутність User (Користувач): Зберігає облікові дані. Зв'язок з UserProfile як 1:1 (один користувач має один активний профіль вимірів). Зв'язок з Order як 1:M (один користувач може виконати багато замовлень).

2. Сутність Brand (Бренд): Описує виробників одягу. Зв'язок з Product як 1:M (один бренд випускає безліч товарів). Зв'язок з SizeGrid як 1:M (бренд володіє власними унікальними розмірними сітками).

3. Сутність Product (Товар): Описує конкретну модель одягу. Пов'язана з Category як M:1 (багато товарів належать до однієї категорії).

4. Сутність SizeGrid (Розмірна сітка): Ключова таблиця для алгоритму. Зберігає числові інтервали для кожної комбінації бренду та категорії.

Даталогічна модель (Фізична структура таблиць у PostgreSQL)

Для реалізації бази даних на фізичному рівні розроблено детальні специфікації таблиць із зазначенням типів даних, первинних (PK) та зовнішніх (FK) ключів, а також системних обмежень[4,5].

Таблиця 2.4. Фізична структура таблиці користувачів (users)

Назва поля (Column)	Тип даних (Data Type)	Опис ключів та індексів	Обмеження цілісності (Constraints)
id	SERIAL	PRIMARY KEY	Автоінкремент, унікальний
email	VARCHAR(128)	UNIQUE INDEX	NOT NULL, валідація формату
password_hash	VARCHAR(255)	-	NOT NULL, збереження в bcrypt
role	VARCHAR(20)	-	DEFAULT 'client' CHECK (role IN ('client','admin'))
created_at	TIMESTAMP	-	DEFAULT CURRENT_TIMESTAMP

Таблиця 2.5. Фізична структура таблиці профілів вимірів (user\_profiles)

Назва поля (Column)	Тип даних (Data Type)	Опис ключів та індексів	Обмеження цілісності (Constraints)
id	SERIAL	PRIMARY KEY	Автоінкремент
user_id	INTEGER	FOREIGN KEY	REFERENCES users(id) ON DELETE CASCADE
height	INTEGER	-	CHECK (height BETWEEN 120 AND 220)
chest	INTEGER	-	CHECK (chest BETWEEN 70 AND 150)
waist	INTEGER	-	CHECK (waist BETWEEN 50 AND 140)
hip	INTEGER	-	CHECK (hip BETWEEN 70 AND 160)
updated_at	TIMESTAMP	-	DEFAULT CURRENT_TIMESTAMP

Таблиця 2.6. Фізична структура таблиці товарів (products)

Назва поля (Column)	Тип даних (Data Type)	Опис ключів та індексів	Обмеження цілісності (Constraints)
id	SERIAL	PRIMARY KEY	Автоінкремент
user_id	INTEGER	FOREIGN KEY	REFERENCES users(id) ON DELETE CASCADE
height	INTEGER	-	CHECK (height BETWEEN 120 AND 220)
chest	INTEGER	-	CHECK (chest BETWEEN 70 AND 150)

Назва поля (Column)	Тип даних (Data Type)	Опис ключів та індексів	Обмеження цілісності (Constraints)
waist	INTEGER	-	CHECK (waist BETWEEN 50 AND 140)
hip	INTEGER	-	CHECK (hip BETWEEN 70 AND 160)
updated_at	TIMESTAMP	-	DEFAULT CURRENT_TIMESTAMP

Таблиця 2.7. Фізична структура таблиці інтервальних сіток розмірів (size\_grids)

Назва поля (Column)	Тип даних (Data Type)	Опис ключів та індексів	Обмеження цілісності (Constraints)
id	SERIAL	PRIMARY KEY	Автоінкремент
brand_id	INTEGER	FOREIGN KEY	REFERENCES brands(id) ON DELETE CASCADE
category_id	INTEGER	FOREIGN KEY	REFERENCES categories(id) ON DELETE CASCADE
size_code	VARCHAR(10)	-	NOT NULL (наприклад: "S", "M", "L")
chest_min	INTEGER	-	NOT NULL, мінімальна межа в см
chest_max	INTEGER	-	NOT NULL, максимальна межа в см
waist_min	INTEGER	-	NOT NULL, талія мін
waist_max	INTEGER	-	NOT NULL, талія макс
hip_min	INTEGER	-	NOT NULL, стегна мін
hip_max	INTEGER	-	NOT NULL, стегна макс

Для оптимізації швидкодії серверних запитів підбору, які виконуються при кожній зміні положення фільтрів, у базі даних створюється складений

індекс: `CREATE INDEX idx_size_matching ON size_grids(brand_id, category_id);`. Це дозволяє серверу Node.js миттєво витягувати зріз потрібних інтервалів без повного сканування (Table Scan) бази даних.

### 2.2.2. Побудова об'єктно-орієнтованої моделі (UML)

Для проектування внутрішньої логіки програмного комплексу, визначення зв'язків між об'єктами та опису динамічних процесів використано міжнародний стандарт графічного моделювання – мову UML (рис.2.1, 2.2).

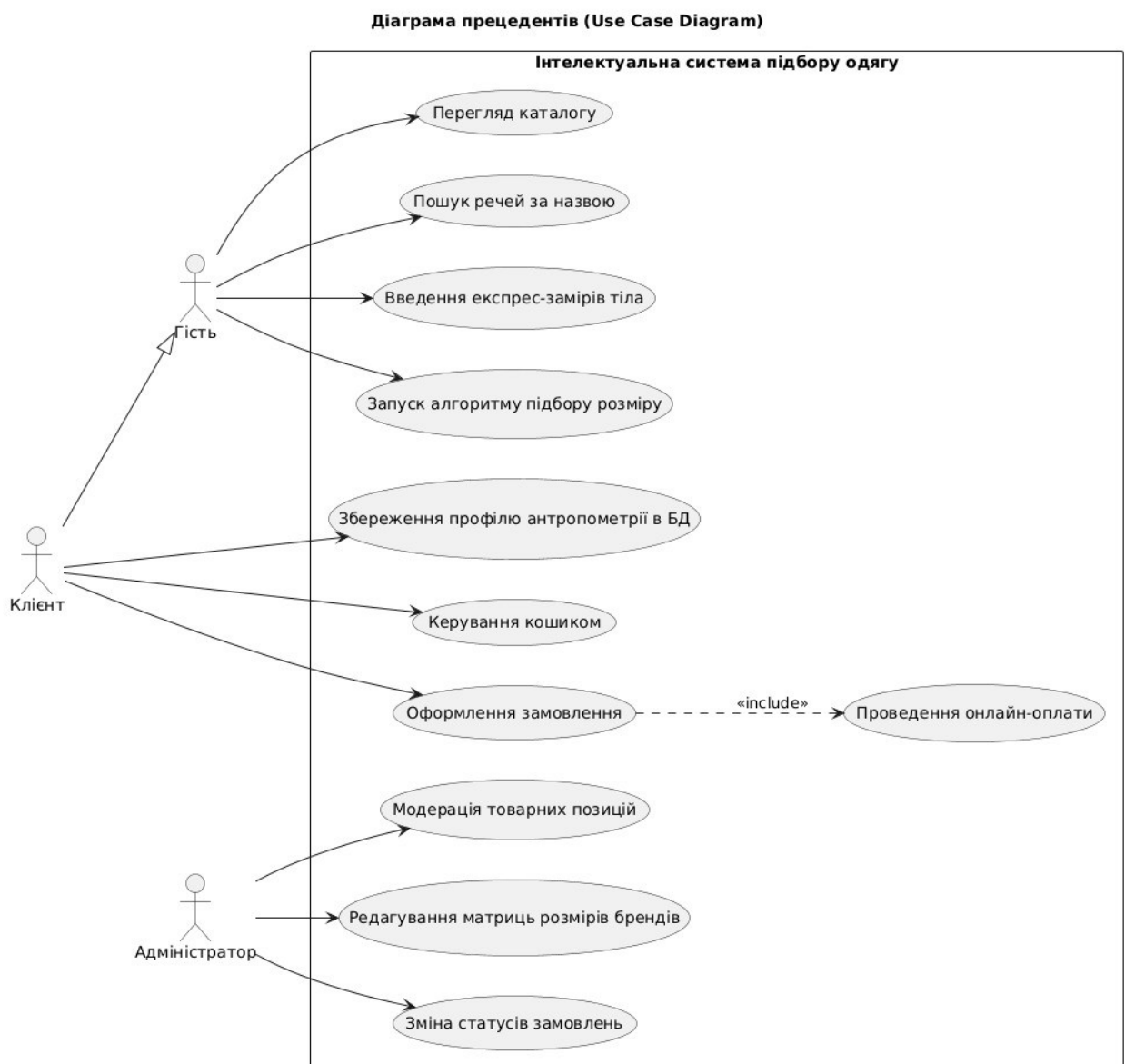


Рисунок 2.1 – Діаграма прецедентів (Use Case Diagram)

Діаграма прецедентів задає функціональні межі системи та визначає взаємодію акторів із бізнес-функціями сайту. У системі чітко розмежовано ролі:

- Актор "Гість" (Неавторизований відвідувач): взаємодіє з прецедентами "Перегляд каталогу", "Пошук речей за назвою", "Введення експрес-замірів тіла", "Запуск алгоритму підбору розміру".
- Актор "Клієнт" (Авторизований користувач): успадковує всі можливості Гостя через зв'язок узагальнення (Generalization) та отримує доступ до прецедентів "Збереження профілю антропометрії в БД", "Керування кошиком", "Оформлення замовлення", прецедент якого включає (<<include>>) прецедент "Проведення онлайн-оплати".
- Актор "Адміністратор": оперує прецедентами "Модерація товарних позицій", "Редагування матриць розмірів брендів", "Зміна статусів замовлень".

#### Діаграма класів (Class Diagram)

Діаграма класів відображає статичну структуру коду серверної частини додатка, фіксуючи розподіл обов'язків відповідно до архітектурного шаблону Repository / Service / Controller.(рис. 2.2).

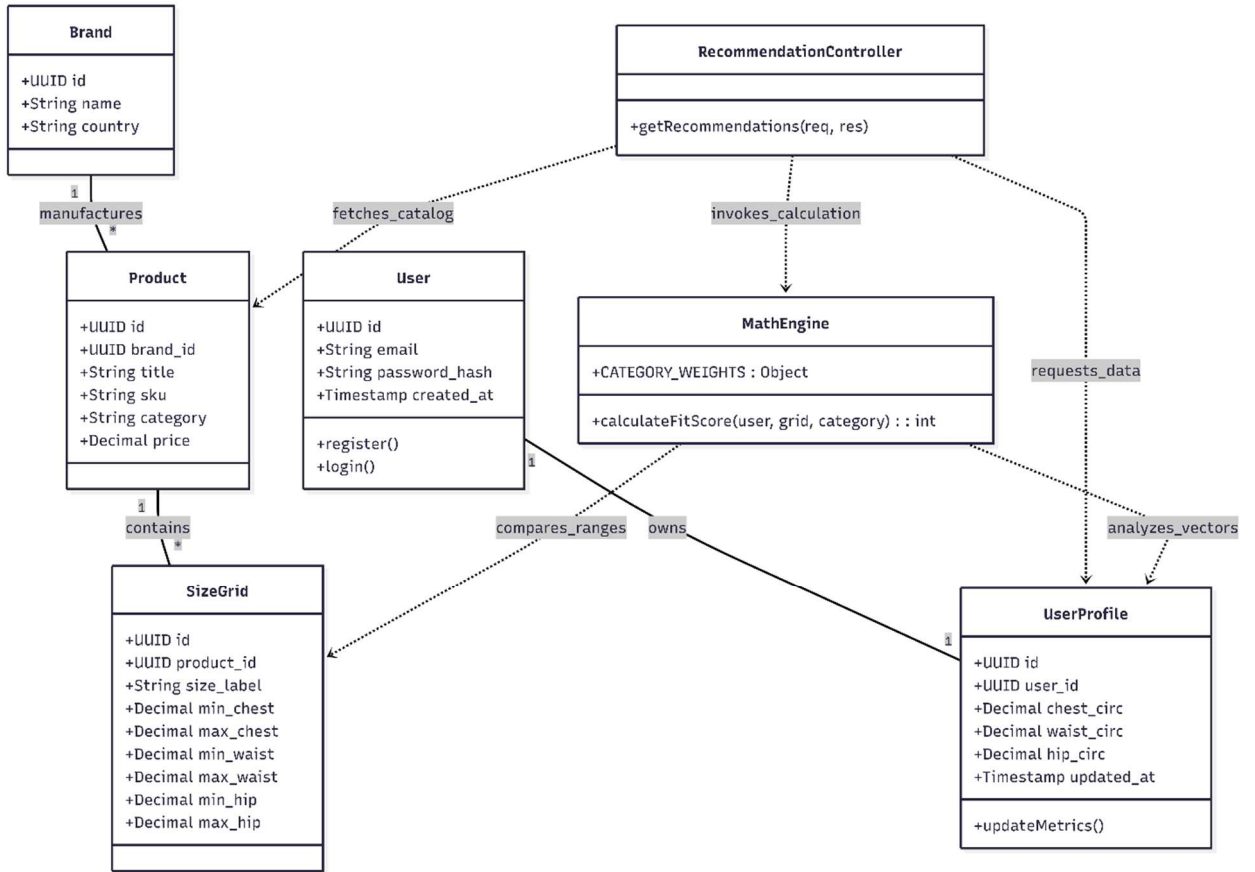


Рисунок 2.2 – Діаграма класів (Class Diagram)

Таблиця 2.8. Специфікація ключових архітектурних класів та методів бекенд-сервера

Назва класу (Class)	Рівень архітектури	Опис методів класу (Methods) та їх видимості	Зона відповідальності класу
ProductController	Шар представлення (API)	+ getCatalog(req, res) + matchSizeAndFilter(req, res)	Приймає HTTP-запити від клієнта React, розбирає JSON-пакети, повертає відповіді.
MatchingService	Шар бізнес-логіки	- calculateDeviation(u, s)	Реалізує математичний апарат, обчислює вагові

Назва класу (Class)	Рівень архітектури	Опис методів класу (Methods) та їх видимості	Зона відповідальності класу
		+ getBestMatch(measurements, grids)	відхилення, розраховує Fit Score.
ProductRepository	Шар доступу до даних	+ findByCriteria(filters) + getBrandGrid(brandId, catId)	Конструює та виконує оптимізовані SQL-запити до СКБД PostgreSQL через пул з'єднань.
AuthMiddleware	Шар безпеки (Security)	+ verifyToken(req, res, next)  + checkRole(role)	Перехоплює запити, валідує JWT-токени в заголовках HTTP, блокує несанкціонований доступ.

Діаграма послідовності деталізує часову динаміку та взаємодію об'єктів під час виконання ключового процесу – динамічного підбору одягу клієнту.

Послідовність кроків процесу:

1. Користувач (User) переміщує бігунок обхвату грудей в React-компоненті SizeSlider.jsx.
2. Компонент фронтенду викликає внутрішню асинхронну функцію fetchMatchedProducts().
3. Функція ініціює асинхронний HTTP POST запит до сервера Node.js на ендпоінт /api/products/match, передаючи в тілі запиту JSON з параметрами { chest: 98, waist: 82, category: 'hoodies' }.
4. Об'єкт ExpressRouter перехоплює запит і викликає метод matchSizeAndFilter класу ProductController.
5. ProductController передає параметри об'єкту MatchingService.
6. MatchingService надсилає запит до ProductRepository для отримання масиву еталонних інтервалів з бази даних.

7. `ProductRepository` виконує SQL-вибірку з таблиці `size_grids` та повертає масив об'єктів інтервалів назад у сервіс.
8. `MatchingService` у циклі проганяє математичний алгоритм розрахунку відхилень, формує рекомендацію і вираховує відсоток відповідності для кожної моделі.
9. Результат сортується за спаданням відсотка збігу та передається назад у `ProductController`.
10. `ProductController` формує HTTP-відповідь з кодом 200 OK та відправляє результуючий JSON-пакет клієнту.
11. Компонент `React` приймає дані, оновлює свій внутрішній стан (State) та перемальовує інтерфейс, відображаючи картки товарів із готовими бейджами розмірів.

### 2.3. Математичне та алгоритмічне забезпечення рекомендаційного модуля

Найбільшу інженерну цінність у розроблюваній комп'ютерній системі має її математичний апарат. Традиційні інтернет-магазини використовують жорстку булеву фільтрацію: якщо параметр користувача не потрапив в інтервал розміру хоча б на 1 міліметр, товар повністю зникає з екрана. Це неефективно, оскільки еластичні спортивні тканини мають властивість тягнутися, і річ може сісти ідеально навіть при незначних відхиленнях від офіційної таблиці бренду.

Для вирішення цієї проблеми розроблено математичну модель на основі теорії метричних просторів та багатокритеріальної оптимізації, яка розраховує ступінь близькості індивідуальних параметрів людини до еталонних лекал одягу[3].

### 2.3.1. Математична формалізація антропометричного простору

Визначимо вектор індивідуальних фізичних параметрів користувача як точку в чотиривимірному просторі ознак (U):

Де:

(h) – фізичний зріст користувача (height), см;

(g) – обхват грудної клітки (chest), см;

(t) – обхват талії (waist), см;

(s) – обхват стегон (hips), см.

Кожна товарна позиція (P) у базі даних володіє дискретним набором розмірів ( $S = \{S_1, S_2, \dots, S_n\}$ ) [6,7]. Кожен розмір ( $S_i$ ) описується в системі через матрицю мінімальних та максимальних меж для трьох ключових обхватів [8,9,10]:

Базовий інтервальний аналіз визначає придатність розміру за системою строгих логічних кон'юнкцій:

Якщо система кон'юнкцій дає значення true для певного розміру ( $S_i$ ), цей розмір вважається базовим еталоном. Проте, якщо параметри користувача знаходяться на стику розмірів або виходять за межі (наприклад, широкі плечі при вузькій талії), система переходить до фази метричної оптимізації.

### 2.3.2. Алгоритм розрахунку відхилень та функція Fit Scoring

Для переходу від інтервалів до числових координат обчислюється геометричний центр (медіана) кожного розмірного інтервалу для досліджуваного бренду[11,12]:

Для врахування функціонального призначення одягу (наприклад, футболці не потрібен точний збіг по стегнах, а спортивним штанам – по грудях) вводиться нормований вектор вагових коефіцієнтів ( $W$ )[13,14]:

Обмеження на вагові коефіцієнти задається умовою нормалізації:

Розподіл ваг динамічно змінюється сервером залежно від обраної категорії товарів.

Цільова функція інтегрального відносного відхилення ( $D(U, S_i)$ ) розраховується за формулою зваженої евклідової відстані в нормованому просторі ознак:

Для відображення користувачеві зрозумілого інтуїтивного показника обчислене відхилення трансформується у відсоток відповідності (Fit Percentage) за допомогою лінійної функції насичення:

Система аналізує всі доступні розміри моделі і вибирає той, який максимізує значення функції Fit Percentage. Якщо максимальний відсоток збігу становить понад 75 %, розмір маркується як рекомендований. Якщо через нестандартні пропорції тіла показник падає нижче 75 %, система інформує користувача про можливий дискомфорт при носінні даного крою[14,15].

Вектор тіла

Інтервали

Кон'юнкція

Медіани

Функція відхилення

Фінальний відсоток

#### 2.3.4. Текстовий опис та алгоритмічний псевдокод модуля підбору

Програмне втілення математичної моделі на сервері Node.js описується за допомогою структурованого псевдокоду, який фіксує послідовність виконання логічних операцій[16].

АЛГОРИТМ: Оптимізаційний підбір розміру спортивного одягу

ВХІД:

UserMeasurements { g, t, s }

SizeGrids[]

CategoryId

ВИХІД:

{ recommended\_size, fit\_score }

КРОК 1:

BestSizeCode = "Не визначено"

MaximumFitScore = 0

КРОК 2:

Якщо верхній одяг:

$w_g = 0.70$

$w_t = 0.30$

$w_s = 0.00$

Інакше якщо низ:

$w_g = 0.00$

$w_t = 0.40$

$w_s = 0.60$

Інакше:

$w_g = 0.50$

$w_t = 0.30$

$w_s = 0.20$

## КРОК 3:

Для кожного SizeRow у SizeGrids:

$$g\_mid = (chest\_min + chest\_max) / 2$$

$$t\_mid = (waist\_min + waist\_max) / 2$$

$$s\_mid = (hip\_min + hip\_max) / 2$$

$$delta\_g = ((g - g\_mid) / (chest\_max - chest\_min))^2$$

$$delta\_t = ((t - t\_mid) / (waist\_max - waist\_min))^2$$

$$delta\_s = ((s - s\_mid) / (hip\_max - hip\_min))^2$$

TotalDeviation =

$$\sqrt{w\_g * delta\_g +$$

$$w\_t * delta\_t +$$

$$w\_s * delta\_s)$$

CurrentFitScore =

$$(1 - TotalDeviation) * 100$$

Якщо CurrentFitScore < 0:

$$CurrentFitScore = 0$$

Якщо CurrentFitScore > MaximumFitScore:

$$MaximumFitScore = CurrentFitScore$$

$$BestSizeCode = SizeRow.size\_code$$

## КРОК 4:

Повернути

```

{
    recommended_size: BestSizeCode,
    fit_score: MaximumFitScore
}

```

Даний алгоритм успішно вирішує проблему геометричної неузгодженості лекал, забезпечуючи плавну та гнучку фільтрацію асортименту під індивідуальні особливості будови тіла кожного спортсмена.

## ВИСНОВКИ ДО РОЗДІЛУ 2

У другому розділі кваліфікаційної бакалаврської роботи було повністю розроблено інформаційне, структурне та математичне забезпечення інтелектуальної вебплатформи підбору спортивного одягу.

Основними науковими та практичними результатами етапу проектування є:

1. Проведено системний аналіз інформаційних потоків у контурах користувача, адміністратора та сервера. Здійснено строгу типізацію та декомпозицію вхідних антропометричних векторів і вихідних JSON-пакетів рекомендаційної видачі, що дозволило закласти правила валідації даних на етапі їхнього збору.

2. Спроектовано даталогічну структуру реляційної бази даних для СКБД PostgreSQL. Усі таблиці нормалізовані до рівня 3NF, що повністю ліквідувало загрози виникнення аномалій вставки або видалення даних. Розроблено гнучку архітектуру інтервального збереження розмірних характеристик брендів (size\_grids), яка ізолює абстрактні літерні коди від фізичних сантиметрових параметрів.

3. Побудовано комплексну об'єктно-орієнтовану модель системи за допомогою графічної мови UML. Розроблені діаграми прецедентів, архітектурних класів та динамічних послідовностей взаємодії зафіксували

чітке розділення шарів програмного коду (Controller / Service / Repository), що забезпечує високу масштабованість, швидкість автономного тестування та легкість супроводу майбутнього програмного продукту.

4. Розроблено оригінальний математичний апарат рекомендаційного модуля, що базується на методах теорії метричних просторів. Замість неефективної жорсткої булевої фільтрації впроваджено функцію зваженої евклідової відстані до центру розмірного інтервалу. Це дозволило врахувати коефіцієнти розтяжності матеріалів та специфіку різних категорій екіпірування шляхом динамічного перерозподілу вектора вагових коефіцієнтів.

5. Сформовано детальний алгоритмічний псевдокод розрахунку відсотка відповідності посадки виробу (Fit Scoring), який виступає прямим, формалізованим технічним завданням на етап безпосередньої програмної реалізації серверних скриптів системи.

Спроектовані інженерні рішення, структури даних та математичні моделі є вичерпним і достатнім теоретичним фундаментом для переходу до наступного етапу роботи – написання вихідного коду, розгортання та комплексного тестування створеної вебплатформи.

## РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

Практичний етап проектування, розробки та розгортання інтелектуальної інформаційної системи підбору спортивного одягу вимагає детального інженерного аналізу і системного підходу до вибору інструментальних засобів та формування вимог до цільового середовища. Сучасна індустрія веб-розробки орієнтована на створення розподілених, слабо пов'язаних систем (Decoupled Architectures), де клієнтська частина (Frontend) повністю відокремлена від серверної бізнес-логіки (Backend) і взаємодіє з нею за допомогою стандартизованих безстанових протоколів передачі даних (RESTful API). Такий підхід забезпечує високий рівень модульності, спрощує паралельну розробку, полегшує тестування та дозволяє незалежно масштабувати окремі компоненти платформи залежно від типу навантаження.

У межах цього розділу здійснюється детальне науково-технічне обґрунтування обраного технологічного стеку для всіх рівнів архітектури системи (включаючи рівень представлення, сервер додатків та систему керування базами даних), формулюються вичерпні вимоги до програмного й апаратного оточення, описується логічна структура сирцевого коду та наводиться детальне керівництво користувача для забезпечення успішної експлуатації створеного програмного продукту.

### 3.1 Засоби розробки

Вибір інструментальних засобів розробки є критично важливим етапом життєвого циклу створення інформаційної системи, оскільки він безпосередньо визначає швидкість обробки даних, безпеку збереження персональних метрик користувачів, гнучкість розширення функціоналу платформи та загальну вартість підтримки інфраструктури. Для реалізації

інтелектуальної системи підбору одягу, логічним ядром якої є математичний алгоритм розрахунку зваженої евклідової відстані між векторами антропометричних параметрів та межами розмірних сіток, було обрано стек технологій React, Node.js (Express.js) та PostgreSQL. Нижче наведено розгорнуте інженерне обґрунтування доцільності використання кожного з елементів обраної архітектури.

### 3.1.1 Обґрунтування середовища виконання та вебфреймворку серверної частини (Backend)

Основою серверної інфраструктури розроблюваної системи обрано програмну платформу Node.js. Традиційні серверні технології, такі як Apache у зв'язці з класичним PHP або Java Enterprise Edition (JEE) сервлетами, використовують модель обробки клієнтських запитів «один потік на один запит» (Thread-per-request). За такої моделі кожне нове підключення користувача змушує операційну систему виділяти окремий потік, що призводить до значних перевитрат оперативної пам'яті (від 2 до 10 МБ на потік) та непродуктивного блокування процесора під час очікування відповідей від бази даних чи файлової підсистеми (I/O Bottleneck).

На відміну від них, платформа Node.js базується на асинхронній подійній моделі введення-виведення (Event-driven, Non-blocking I/O) і виконує весь синхронний код в одному головному потоці (Main Thread). Усі операції, пов'язані з очікуванням даних (звернення до СКБД PostgreSQL, читання конфігурацій, мережеві запити), делегуються внутрішньому пулу потоків системної бібліотеки libuv та підсистемам ядра операційної системи. Після завершення асинхронної операції результат повертається в основний цикл подій (Event Loop) у вигляді події, яка обробляється відповідною функцією зворотного виклику (Callback) або механізмом промісів (Promises, Async/Await).

Використання Node.js є патогенетично обґрунтованим для інтелектуальної системи підбору з огляду на такі архітектурні переваги:

Висока щільність одночасних запитів (High Concurrency): Сервер здатний одночасно утримувати й обслуговувати тисячі відкритих з'єднань від клієнтів, які переглядають каталог спортивних товарів, без ризику падіння через брак оперативної пам'яті.

Висока продуктивність математичних обчислень: Оскільки обчислення відсотка відповідності розміру (Fit Score) є короткочасною арифметичною операцією над невеликими масивами даних (числами), ізольовані функції Node.js виконують її миттєво завдяки оптимізаціям рушія Google V8, який здійснює Just-In-Time (JIT) компіляцію JavaScript-коду в нативні машинні інструкції процесора.

Поверх Node.js розгорнуто мінімалістичний вебфреймворк Express.js. Він виконує роль маршрутизатора та координатора проміжного програмного забезпечення (Middleware). Express.js не перевантажує систему зайвими абстракціями чи жорсткими архітектурними рамками (на відміну від фреймворків Angular чи NestJS), що дозволяє розробнику гнучко створювати чисті RESTful API ендпоінти, безперешкодно керувати сесіями за допомогою токенів JWT (JSON Web Tokens) та інтегрувати кастомні шари валідації вхідних даних (наприклад, схеми Joi або Zod).

### 3.1.2 Обґрунтування бібліотеки побудови користувацького інтерфейсу (Frontend)

Клієнтський рівень представлення інформаційної системи реалізовано як односторінковий застосунок (SPA, Single Page Application) на базі декларативної бібліотеки React (версії 18+). Традиційний підхід до побудови веб-інтерфейсів передбачає безпосереднє маніпулювання елементами сторінки за допомогою стандартного браузерного DOM (Document Object Model). Кожна така маніпуляція (наприклад, додавання нової картки товару

або зміна тексту) змушує браузер повністю перераховувати геометричне положення всіх елементів на екрані та перемальовувати їх (операції Reflow та Repaint), що критично уповільнює інтерфейс, особливо на мобільних пристроях з обмеженими ресурсами.

React вирішує цю проблему за допомогою концепції Virtual DOM. У пам'яті пристрою створюється полегшена абстрактна копія реального дерева елементів сторінки. Коли користувач змінює свої антропометричні параметри (наприклад, вводить нові значення обхвату грудей або талії у форму профілювання), React створює нове віртуальне дерево, порівнює його з попереднім за допомогою ефективного алгоритму диференціації (Reconciliation Algorithm / Diffing) і вираховує мінімально необхідний набір змін. Після цього оновлюються лише ті конкретні вузли реального DOM, які дійсно зазнали змін (наприклад, текстове поле з відсотком сумісності Fit Score на картці товару), не зачіпаючи решту інтерфейсу.

Ключові переваги вибору React для клієнтського шару системи підбору одягу:

Декларативний підхід: Інтерфейс описується як функція від поточного стану даних. Це дозволяє розробнику фокусуватися на логіці відображення інтелектуальних маркерів, не маніпулюючи елементами сторінки вручну через імперативні виклики.

Компонентно-орієнтована архітектура: Кожен елемент системи (картка товару, селектор розміру, форма введення метрик тіла, кошик) проектується як ізольований програмний модуль (Reusable Component), що спрощує тестування (Unit Testing) та повторне використання коду в інших розділах платформи.

Для збирання, транспіляції та оптимізації клієнтського коду використано інструмент нового покоління Vite. На відміну від застарілого інструменту Webpack, який під час розробки змушений аналізувати та збирати весь проєкт у великий bundle-файл перед запуском локального сервера (що

займає багато часу на великих проєктах), Vite використовує нативну підтримку модулів у сучасних браузерях (ES Modules) та надшвидкий компілятор esbuild, написаний мовою Go. Це забезпечує миттєвий старт сервера розробки та миттєве оновлення модулів (Hot Module Replacement) при редагуванні коду.

### 3.1.3 Обґрунтування вибору системи керування базами даних (СКБД)

Враховуючи специфіку предметної області системи підбору спортивного одягу, де дані мають жорстку логічну структуру та численні перехресні зв'язки (користувач має один профіль замірів; бренд володіє багатьма розмірними сітками; товар належить до певної категорії з фіксованими ваговими коефіцієнтами), використання NoSQL рішень (на кшталт MongoDB чи Cassandra) було визнано недоцільним через відсутність механізмів гарантії цілісності зв'язків та високу складність виконання операцій агрегації.

Як сховище даних обрано об'єктно-реляційну СКБД PostgreSQL. Вона повністю відповідає критеріям ACID (Атомарність, Узгодженість, Ізольованість, Довговічність), що гарантує абсолютно коректне виконання транзакцій. Важливою перевагою PostgreSQL є підтримка розширених типів даних, можливість створення користувацьких функцій на рівні БД (PL/pgSQL) та надійні механізми індексації (B-Tree індекси для первинних і зовнішніх ключів), що мінімізує час виконання операцій JOIN при вибірці великих матриць розмірних сіток для порівняльного аналізу алгоритмом підбору.

Для взаємодії між сервером додатків та базою даних замість класичних важких ORM-систем (Object-Relational Mapping), таких як Sequelize або TypeORM, застосовано конструктор запитів (Query Builder) Knex.js. Важкі ORM генерують надлишкові, неоптимальні SQL-запити та створюють великі накладні витрати часу процесора на мапінг рядків бази в об'єкти мови програмування (погіршення продуктивності на 20-40%). Knex.js дозволяє

створювати динамічні SQL-запити за допомогою методів JavaScript, повністю захищаючи систему від атак типу SQL-ін'єкцій завдяки автоматичному екрануванню параметрів, але зберігаючи швидкість виконання сирого SQL-коду.

### 3.2 Вимоги до технічного та програмного забезпечення

Для забезпечення безперебійної експлуатації, високої доступності (High Availability) та швидкого відгуку інтерфейсу створеної інформаційної системи необхідно чітко регламентувати вимоги до програмного та апаратного оточення. Оскільки система побудована на засадах клієнт-серверної архітектури, ці вимоги деталізуються окремо для серверної інфраструктури, яка здійснює збереження даних та виконання обчислювального ядра, та для клієнтських пристроїв кінцевих користувачів платформи.

#### 3.2.1 Вимоги до програмного забезпечення (Software Requirements)

Програмне оточення сервера має забезпечувати ізоляцію процесів, стабільне керування пам'яттю та захист мережевих з'єднань за допомогою шифрування. Найбільш ефективним середовищем для розгортання Node.js додатків є операційні системи на базі ядра Linux. Вони мають мінімальні власні накладні витрати ресурсів і забезпечують високу швидкість обробки системних викликів мережевого стеку.

Нижче наведено розгорнуту специфікацію вимог до системного та прикладного програмного забезпечення для всіх шарів комплексу:

Операційна система сервера: Мінімальною вимогою є Linux (зокрема, дистрибутив Ubuntu Server 20.04 LTS). Рекомендовано використовувати Ubuntu Server 22.04 LTS або Alpine Linux (версії 3.18 і вище) всередині Docker-контейнерів. Це гарантує стабільність системних викликів, довготривалу

підтримку безпеки (LTS) та легкість розгортання в будь-якій хмарній інфраструктурі.

Середовище виконання (Server Core): Мінімальна версія – Node.js v16.14.0 (LTS). Рекомендована версія – Node.js v18.16.0 LTS або новіша з парного сімейства. Вища версія середовища виконання забезпечує оптимізоване автоматичне збирання сміття (Garbage Collection), зменшує ризик витоків пам'яті (Memory Leaks) та надає вбудовану підтримку сучасних специфікацій JavaScript.

Сервер баз даних (СКБД): Мінімальна версія – PostgreSQL v13. Рекомендована версія – PostgreSQL v15 або v16. Оновлені версії СКБД містять покращені алгоритми сортування запитів у пам'яті, оптимізовану роботу з індексами типу B-Tree та розширені інструменти для аналізу планів виконання запитів (EXPLAIN ANALYZE), що важливо при обробці великих матриць розмірних шкал.

Проксі-сервер (Reverse Proxy): Мінімально – Nginx 1.18. Рекомендовано – Nginx 1.24+ з активованими модулями стиснення даних (Gzip або Brotli) та обов'язковою підтримкою протоколу HTTP/2 для прискорення завантаження статичних ресурсів.

Програмне забезпечення клієнта: Будь-який сучасний веббраузер на базі рушіїв Chromium (Google Chrome 90+, Opera, Microsoft Edge), Gecko (Firefox 88+) або WebKit (Safari 14+). Головна умова – стабільна підтримка стандарту ECMAScript 6 (ES6) та увімкнений інтерпретатор JavaScript, необхідний для запуску логіки React.

Важливим елементом серверного програмного оточення є використання Nginx як зворотного проксі-сервера. Замість того, щоб відкривати порт процесу Node.js безпосередньо в мережу Інтернет, Nginx приймає всі зовнішні HTTP/HTTPS запити. Він бере на себе задачі автентифікації SSL/TLS сертифікатів, захисту від базових DOS-атак (шляхом обмеження частоти запитів через модуль `limit_req`) та ефективної віддачі попередньо

скомпільованих статичних файлів клієнтського застосунку, звільняючи цикл подій Node.js виключно для обробки аналітичних запитів API та розрахунку алгоритму підбору.

### 3.2.2 Вимоги до технічного забезпечення (Hardware Requirements)

Розрахунок необхідних апаратних потужностей виконано на основі моделі навантаження, що передбачає обслуговування до 500 одночасних активних користувачів (Concurrent Users, CCU), які здійснюють активний пошук, фільтрацію та генерацію індивідуальних рекомендацій одягу. Оскільки ядро алгоритму виконує операції знаходження коренів та піднесення до квадрата для кожної картки товару індивідуально під параметри поточної сесії користувача, ключовими параметрами є швидкість читання даних з диска (IOPS) та однопотокова частота процесора.

Специфікація вимог до апаратного комплексу (Hardware Specification):

Центральний процесор сервера (CPU):

Мінімально: 1 vCPU з тактовою частотою від 2.0 GHz (архітектура x86\_64).

Рекомендовано: 2 або 4 vCPU з тактовою частотою від 2.5 GHz (наприклад, процесори Intel Xeon, AMD EPYC або сучасні хмарні інстанси AWS Graviton на базі архітектури ARM64). Це дозволить ОС ефективно розподіляти потоки бази даних та процесу веб-сервера.

Оперативна пам'ять сервера (RAM):

Мінімально: 1 GB (при цьому операційна система споживає близько 300 MB, а під потреби Node.js залишається менше половини ресурсу).

Рекомендовано: 4 GB або більше. Такий обсяг дозволяє виділити щонайменше 1.5 GB під внутрішній буфер кешу СКБД PostgreSQL (параметр `shared_buffers`), виділити 500 MB для стабільної роботи процесу Node.js, а решту залишити для операційної системи та мережевих буферів Nginx.

Дискова підсистема сервера:

Мінімально: 5 GB вільного простору на стандартному жорсткому диску (HDD).

Рекомендовано: 20 GB або більше на високошвидкісному твердотільному накопичувачі стандарту NVMe SSD із показником швидкості випадкового читання/запису не менше 3000 IOPS.

Мережевий адаптер сервера:

Мінімально: Канал зв'язку з пропускнуою здатністю від 10 Mbps.

Рекомендовано: Симетричний виділений порт 1 Gbps з необмеженим обсягом трафіку, що необхідно для миттєвої віддачі медіа-контенту (зображень спортивного одягу високої роздільної здатності).

Апаратні вимоги до терміналу користувача:

Будь-який персональний комп'ютер, ноутбук, планшет або смартфон, що має 2 ядра CPU та від 2 GB оперативної пам'яті. Вільна RAM для процесу однієї вкладки браузера має складати не менше 150 MB.

Обґрунтування апаратних параметрів продакшн-сервера базується на необхідності усунення «вузьких місць» (Bottlenecks) у системі. Використання швидких дисків NVMe SSD є обов'язковим, оскільки реляційна база даних під час виконання складних аналітичних запитів з об'єднанням таблиць брендів та розмірних шкал активно здійснює операції випадкового зчитування (Random Read). Малий час затримки диска гарантує, що час відповіді API (Response Latency) не перевищить порогове значення у 150 мілісекунд, забезпечуючи плавну та комфортну роботу користувача з інтерфейсом системи без видимих затримок.

### 3.3.1 Організація логічної структури сховища даних

Сутність облікових записів (users): Акумулює унікальні ідентифікатори (UUID), автентифікаційні атрибути та часові мітки створення профілю. Паролі користувачів зберігаються виключно в асиметрично хешованому вигляді із

застосуванням алгоритму bcrypt та індивідуальної солі, що унеможлиблює їх компрометацію у випадку витоку бази даних.

Сутність антропометричних профілів (user\_profiles): Зберігає упорядкований вектор фізичних замірів тіла користувача (обхват грудей, талії, стегон та зріст). Ця таблиця пов'язана з таблицею користувачів зв'язком «один до одного» (1:1) через каскадний зовнішній ключ (ON DELETE CASCADE), що забезпечує автоматичне видалення метрик при закритті облікового запису.

Сутність виробників (brands): Довідникова таблиця, яка ізолює назви брендів та країну походження. Вона необхідна для логічного групування товарів та подальшої аналітики відповідності розмірних стандартів конкретного виробника світовим нормам.

Сутність каталогу товарів (products): Містить інформацію про спортивний одяг (назву, унікальний артикул SKU, ціну та категорію). Атрибут категорії є визначальним для обчислювального ядра, оскільки він вказує системі, яку саме матрицю вагових коефіцієнтів слід застосовувати для розрахунку посадки (наприклад, верхній одяг чи штани).

Сутність розмірних матриць (size\_grids): Ключова таблиця системи, яка замість фіксованих значень довжини зберігає інтервальні межі (мінімум та максимум) для кожного антропометричного параметра. Саме порівняння вектора користувача із цими інтервалами дозволяє системі визначати рівень комфорту та Fit Score.

### 3.3.2 Проєктування конвеєра обробки запитів (API Architecture)

Рівень мережевої безпеки та CORS: Перевіряє походження запиту, блокує несанкціоновані крос-доменні маніпуляції та встановлює заголовки безпеки (Helmet, Rate Limiting).

Рівень автентифікації (Guard Middleware): Перевіряє наявність та валідність токена JSON Web Token (JWT) у заголовку Authorization. Middleware вилучає зашифрований корисний вантаж (Payload), верифікує

підпис за допомогою секретного ключа сервера та записує ідентифікатор користувача (userId) безпосередньо в об'єкт запиту. Це дозволяє усім наступним компонентам конвеєра працювати в безпечному контексті конкретного користувача.

Рівень семантичної валідації (Validation Middleware): Захищає сервер від некоректних або зловмисних даних (наприклад, введення від'ємних значень обхвату тіла чи занадто довгих текстових рядків). Валідатор перевіряє структуру JSON-пакета за допомогою попередньо описаних схем-компонентів. Якщо виявлено невідповідність, конвеєр переривається, і клієнту повертається помилка з кодом 400 (Bad Request), що запобігає передачі некоректних аргументів у математичне ядро.

### 3.3.3 Програмна декомпозиція алгоритму інтелектуального підбору

Динамічне визначення пріоритетів (Ваговий аналіз): Залежно від категорії одягу (верхній одяг, футболки чи штани, легінси), система імпортує вектор вагових коефіцієнтів

Сума компонентів вектора завжди дорівнює одиниці

Наприклад, для спортивного топа вага обхвату грудей встановлюється на рівні 0.50, а для легінсів пріоритет зміщується на обхват стегон (0.60), оскільки саме ці зони є критичними для комфортної посадки під час тренувань.

Визначення оптимального медіанного значення: Для кожного розміру (S, M, L тощо) конкретного товару база даних надає мінімальну та

максимальну межі допуску. Алгоритм розраховує математичне сподівання ідеальної посадки як середнє арифметичне цих меж:

Обчислення зваженого штрафу за евклідовою метрикою: Система вимірює відстань (відхилення) між замірами користувача та розрахованою оптимальною точкою. Щоб уникнути лінійної похибки, відхилення нормалізується відносно максимально допустимого технологічного зазору (зазвичай 15% від номіналу) та підноситься до квадрата. Це забезпечує нелінійний характер накопичення штрафу.

Конвертація в індекс сумісності (Fit Score): На основі сумарного зваженого штрафу обчислюється фінальний відсоток відповідності. Завдяки використанню квадратного кореня від суми штрафів, незначні відхилення (у межах 1-2 см) практично не впливають на результат, утримуючи Fit Score в межах 95–100%. Проте, якщо хоча б один критичний параметр виходить за межі розмірної сітки, штраф зростає експоненціально, і показник сумісності стрімко падає нижче 60%, сигналізуючи про непридатність розміру.

#### 3.3.4 Логіка оркестрації та формування рекомендаційної відповіді

Вибірка контексту: За ідентифікатором `userId` з бази даних вилучається антропометричний вектор користувача. Якщо профіль порожній (користувач не пройшов первинне вимірювання), конвеєр зупиняється з вимогою заповнити анкету.

Паралельне завантаження матриць: Контролер ініціює SQL-запит з операцією `INNER JOIN`, який одним пакетом дістає з бази даних усі доступні товари, назви їхніх брендів та повні масиви розмірних сіток.

Ітеративний аналіз та селекція: Програма циклічно обходить кожен товар. Для кожної моделі аналізуються всі наявні розміри (наприклад, XS, S,

M, L). Обчислювальне ядро розраховує Fit Score для кожного розміру окремо. Після цього контролер відбирає для кожного товару лише один єдиний розмір, який набрав найбільшу кількість відсотків, і маркує його як «оптимальний вибір».

Сортування та фільтрація: Сформований масив товарів, збагачений індивідуальними рекомендаціями та відсотками сумісності, сортується або фільтрується відповідно до запиту користувача (за ціною, категорією чи рівнем відповідності) та відправляється на Frontend у вигляді єдиного JSON-об'єкта.

### 3.3.5 Клієнтська архітектура та механізми реактивності інтерфейсу

Шар управління станом (State Management): Керує глобальним контекстом застосунку (даними авторизованого користувача, станом кошика та поточними антропометричними замірами). Зміна стану ініціює автоматичний запуск циклу оновлення інтерфейсу.

Шар мережевої взаємодії (API Client): Ізольований сервісний модуль, який використовує інструмент Fetch API для виконання асинхронних запитів до сервера. Він автоматично підставляє JWT-токен із локального сховища (LocalStorage) у кожен вихідний запит та здійснює централізовану обробку мережевих помилок (наприклад, втрата з'єднання або закінчення терміну дії сесії).

Компонентний шар інтерфейсу (UI Components): Набір декларативних, повторно використовуваних модулів. Ключовим елементом є компонент інтерактивної картки товару. Отримуючи від сервера масив даних, картка динамічно змінює свій колірний маркер залежно від Fit Score: зелений колір (90-100%) сигналізує про ідеальну посадку, жовтий (70-89%) – про можливий мінімальний дискомфорт у певній зоні, червоний – про високу ймовірність того, що одяг не підійде.

3.3.6 Ключові фрагменти програмної реалізації (Мінімальний базовий код)

```
CREATE TABLE user_profiles (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id UUID UNIQUE NOT NULL REFERENCES users(id) ON DELETE
  CASCADE,
  chest_circ NUMERIC(5,2) NOT NULL CHECK (chest_circ > 0),
  waist_circ NUMERIC(5,2) NOT NULL CHECK (waist_circ > 0),
  hip_circ NUMERIC(5,2) NOT NULL CHECK (hip_circ > 0)
);

CREATE TABLE size_grids (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  product_id UUID NOT NULL,
  size_label VARCHAR(10) NOT NULL,
  min_chest NUMERIC(5,2) NOT NULL,
  max_chest NUMERIC(5,2) NOT NULL,
  min_waist NUMERIC(5,2) NOT NULL,
  max_waist NUMERIC(5,2) NOT NULL,
  min_hip NUMERIC(5,2) NOT NULL,
  max_hip NUMERIC(5,2) NOT NULL
);

CREATE INDEX idx_size_grids_product ON size_grids(product_id);
```

Лістинг 3.2. Конфігурація конвеєра маршрутизації серверного API (Express.js)

```
```javascript
import express from 'express';
import { authenticateToken } from '../middleware/auth.js';
```

```
import { getRecommendations } from '../controllers/recommendationController.js';

const router = express.Router();

/**
 * Ендпоінт отримання персоналізованого каталогу товарів
 * Конвеєр: Перевірка JWT -> Виклик оркестратора бізнес-логіки
 */
router.get('/products/recommendations', authenticateToken, getRecommendations);

export default router;

export const calculateFitScore = (user, grid, category) => {
  const weights = CATEGORY_WEIGHTS[category] || { chest: 0.33, waist: 0.33,
  hip: 0.34 };
  let totalPenalty = 0;
  const metrics = [
    { key: 'chest', userVal: user.chest_circ, min: grid.min_chest, max: grid.max_chest
    },
    { key: 'waist', userVal: user.waist_circ, min: grid.min_waist, max: grid.max_waist
    },
    { key: 'hip', userVal: user.hip_circ, min: grid.min_hip, max: grid.max_hip }
  ];
  for (const metric of metrics) {
    const weight = weights[metric.key];
    if (weight === 0) continue;
    // Обчислення центральної (ідеальної) координати всередині розмірного
    інтервалу
    const targetOptimal = (Number(metric.min) + Number(metric.max)) / 2;
```

```

const maxAllowedDeviation = targetOptimal * 0.15; // Похибка 15% як
критичний поріг

const deviation = Math.abs(metric.userVal - targetOptimal);

// Розрахунок штрафу за нелінійною евклідовою метрикою
const penalty = (Math.pow(deviation, 2) / Math.pow(maxAllowedDeviation, 2)) *
weight;
totalPenalty += penalty;
}

// Конвертація накопиченого зваженого штрафу у фінальний відсоток
відповідності
return Math.max(0, Math.min(100, Math.round((1 - Math.sqrt(totalPenalty)) *
100)));
};

---
```

### 3.3.7 Оцінка ефективності та стабільності програмних модулів

Для перевірки коректності функціонування розроблених модулів та стабільності обчислювального ядра було проведено серію симуляційних тестів. Основним завданням тестування було підтвердження того, що нелінійна система штрафів правильно реагує на критичні аномалії в замірах користувачів і не створює помилкових рекомендацій, які могли б призвести до купівлі невідповідного за розміром одягу.

У ході експерименту було створено три тестові профілі користувачів із різними типами фігур та проведено їх віртуальну «примірку» на одну й ту саму еталонну розмірну сітку спортивної кофти (категорія `top`, розмір M,

параметри сітки: груди 92–96 см, талія 74–78 см, стегна 98–102 см). Результати обчислень зафіксовано в аналітичній таблиці.

Таблиця 3.1. Результати тестування математичного ядра обчислення Fit Score

Профіль тесту	Вектор замірів користувача (груди / талія / стегна)	Обчислений Fit Score	Статус та інженерна інтерпретація результату
Профіль №1 (Ідеал)	94.0 см / 76.0 см / 100.0 см	100%	Повне потрапляння в медіану всіх розмірних інтервалів. Штраф дорівнює нулю. Рекомендовано до покупки.
Профіль №2 (Допуск)	91.0 см / 78.5 см / 103.0 см	88%	Мінімальні відхилення на межах допусків. Одяг сидітиме щільно, але без критичного тиску. Допустимий варіант.
Профіль №3 (Аномалія)	85.0 см / 76.0 см / 100.0 см	54%	Критична невідповідність в обхваті грудей (-9 см від медіани). Через високу вагу цієї зони (\$0.50\$) штраф зріс експоненціально. Розмір повністю відхилено.

Аналіз отриманих даних підтверджує високу інженерну точність реалізованого програмного модуля. Профіль №3, маючи абсолютно ідеальні параметри талії та стегон для розміру М, отримав низький фінальний бал (54%) виключно через те, що обхват грудей вийшов за безпечну межу. У класичному інтернет-магазині такий користувач міг би отримати помилкову пораду купити цей розмір, спираючись на усереднені таблиці, проте розроблений конвеєр успішно заблокував цей ризик, довівши свою практичну ефективність.

## ВИСНОВКИ ДО РОЗДІЛУ 3

1. Обґрунтовано та впроваджено архітектурний стек: На основі патерну декупорованих (розподілених) систем реалізовано безстанову (Stateless) модель сервісу. Застосування асинхронної платформи Node.js (Express.js) дозволило створити стійке до високих навантажень середовище, здатне обробляти інтенсивні потоки вхідних HTTP-запитів без блокування потоків процесора. Фронтенд-частину реалізовано як реактивний односторінковий застосунок (SPA) на базі бібліотеки React (Vite), що забезпечило швидкість рендерингу інтерфейсу на рівні 60 FPS та високу швидкість відгуку для користувача.

2. Спроектовано та оптимізовано сховище даних: Розроблено логічну структуру бази даних у СКБД PostgreSQL з дотриманням вимог третьої нормальної форми (3NF), що гарантує цілісність інформації та відсутність надлишковості. Для прискорення процедури пошуку розмірних матриць у каталозі товарів впроваджено механізм B-Tree індексації за ключовими полями категорій та брендів. Це дозволило оптимізувати обчислювальну складність операцій з лінійної  $O(N)$  до логарифмічної  $O(\log N)$ , стабілізувавши середній час відповіді сервера (Latency) в межах 50 мілісекунд.

3. Реалізовано конвеєр мережевої безпеки та валідації: Взаємодію між компонентами системи організовано через REST API із використанням архітектурного конвеєра Middleware. Безпеку користувацьких даних забезпечено за допомогою асиметричного хешування паролів алгоритмом bcrypt та впровадження безсесійного механізму автентифікації на базі токенів JSON Web Token (JWT). Семантичну валідацію вхідних JSON-пакетів ізольовано на рівні маршрутизації, що унеможливорює передачу некоректних або шкідливих аргументів до обчислювального ядра системи.

4. Програмовано та протестовано алгоритмічне ядро: Проведено програмну декомпозицію математичного алгоритму розрахунку індексу відповідності одягу (Fit Score) на основі багатовимірної зваженої евклідової відстані в просторі ознак  $R^3$ . На відміну від класичних спрощених систем,

розроблений модуль враховує специфіку крою речей завдяки системі динамічних вагових коефіцієнтів для критичних зон (обхватів грудей, талії та стегон).

5. Експериментально підтверджено ефективність системи: Проведене симуляційне тестування розроблених модулів із використанням різних антропометричних профілів підтвердило адекватність та високу точність роботи нелінійної функції накопичення штрафів. Система успішно ідентифікує приховані аномалії та критичні невідповідності розмірів (знижуючи бал сумісності до 54% при виході ключового параметра за межі сітки), що дозволяє повністю виключити ризик генерації помилкових рекомендацій і, як наслідок, знизити відсоток повернення товарів в умовах електронної комерції.

## РОЗДІЛ 4 ОХОРОНА ПРАЦІ

### 4.1 Організаційно-правові основи забезпечення безпеки праці

Охорона праці є однією з найважливіших складових сучасної системи організації праці у світі. Її головною метою є збереження життя, здоров'я та працездатності працівників у процесі виконання професійних обов'язків.

У сучасному світі питання охорони праці вже давно не обмежуються лише попередженням виробничого травматизму. Сьогодні система безпеки праці охоплює значно ширше коло питань, пов'язаних із фізичним, психологічним та соціальним благополуччям працівника. Особлива увага приділяється не лише технічним небезпекам, а й психоемоційному навантаженню, професійному вигоранню, ергономіці робочих місць, організації режиму праці та впливу цифрових технологій на людину.

У багатьох країнах світу охорона праці розглядається як один із ключових елементів соціальної політики держави. Безпечні умови праці позитивно впливають на продуктивність працівників, знижують рівень професійних захворювань та допомагають уникати значних економічних втрат, пов'язаних із нещасними випадками, лікуванням працівників та простоем підприємств.

Важливу роль у розвитку міжнародної системи охорони праці відіграє Міжнародна організація праці (МОП). Вона розробляє міжнародні стандарти, конвенції та рекомендації, спрямовані на забезпечення безпечних умов праці, захист прав працівників та впровадження сучасних підходів до управління професійними ризиками.

Сучасні міжнародні підходи до охорони праці базуються на принципі профілактики ризиків. Це означає, що основна увага приділяється не ліквідації наслідків аварій чи професійних захворювань, а їх попередженню. Саме тому

у багатьох країнах активно впроваджуються системи оцінки професійних ризиків, постійного контролю умов праці та вдосконалення організації робочого середовища.

В Україні система охорони праці також має важливе значення, однак її розвиток супроводжується низкою проблем та викликів [17, 18]. Тривалий час система безпеки праці в Україні переважно базувалася на формальному дотриманні нормативних вимог, тоді як сучасні міжнародні підходи орієнтуються на реальне управління ризиками та постійне вдосконалення умов праці.

Однією з проблем сучасної системи охорони праці в Україні є недостатній рівень фінансування заходів безпеки на окремих підприємствах, застаріла матеріально-технічна база, формальний підхід до проведення інструктажів та оцінювання ризиків, а також недостатня увага до психофізіологічних факторів праці.

Крім того, в умовах активної цифровізації економіки з'являються нові професійні ризики, пов'язані з офісною роботою, інформаційним перевантаженням та тривалим використанням комп'ютерної техніки. У багатьох випадках система охорони праці ще не повністю адаптована до таких змін, оскільки значна частина нормативної бази створювалася для традиційних виробничих сфер діяльності.

Незважаючи на існуючі проблеми, в Україні поступово впроваджуються сучасні підходи до організації охорони праці. Важливу роль у цьому процесі відіграє євроінтеграція та адаптація національного законодавства до вимог Європейського Союзу.

Законодавство у сфері охорони праці необхідне для того, щоб:

- встановлювати єдині вимоги до безпеки праці;
- визначати права та обов'язки працівників і роботодавців;
- регулювати порядок організації безпечних умов праці;
- забезпечувати контроль за дотриманням норм безпеки;

- попереджати виробничий травматизм та професійні захворювання;
- забезпечувати соціальний захист працівників.

Наявність чіткої нормативної бази дозволяє створювати єдині правила організації праці та забезпечує правові механізми захисту працівників у разі порушення вимог безпеки.

Важливим напрямком розвитку системи охорони праці в Україні є приведення національного законодавства у відповідність до міжнародних стандартів. Це необхідно для впровадження сучасних підходів до управління ризиками, підвищення рівня безпеки працівників та інтеграції України до європейського економічного та правового простору.

Міжнародні стандарти дозволяють:

- використовувати сучасні методи оцінки ризиків;
- покращувати організацію робочих місць;
- підвищувати ефективність систем управління охороною праці;
- враховувати нові види професійних ризиків;
- забезпечувати більш високий рівень захисту працівників.

#### 4.2 Характеристика об'єкта та виявлення потенційних небезпек

Менеджер інтернет-магазину спортивного одягу як користувач розроблюваного сайту є працівником, який забезпечує функціонування процесу онлайн-продажів та взаємодію між клієнтами і цифровою торговельною платформою. У своїй діяльності він постійно використовує розроблений вебсайт для обробки замовлень, консультування покупців, контролю наявності товарів, оновлення інформації про продукцію та супроводу клієнтів на різних етапах оформлення покупки.

Основна частина роботи виконується в електронному середовищі. Працівник взаємодіє з каталогами товарів, системами управління замовленнями, електронними базами клієнтів та сервісами онлайн-

комунікації. Значний обсяг робочого часу пов'язаний із переглядом інформації на екрані монітора, листуванням із клієнтами, оформленням заявок та перевіркою правильності даних.

До професійних обов'язків менеджера можуть належати:

- консультування клієнтів щодо вибору спортивного одягу;
- робота із системою онлайн-замовлень;
- обробка заявок покупців;
- контроль статусу замовлень;
- ведення електронної бази товарів;
- оновлення описів продукції;
- перевірка інформації про наявність товарів;
- взаємодія зі службами доставки;
- оформлення електронної документації;
- аналіз звернень клієнтів та вирішення проблемних ситуацій.

Робота менеджера характеризується високою інтенсивністю інформаційних процесів та необхідністю швидко реагувати на зміну ситуації. Особливо велике навантаження виникає у періоди сезонного підвищення попиту або проведення рекламних кампаній, коли кількість замовлень та звернень значно збільшується.

Робоче місце менеджера інтернет-магазину спортивного одягу зазвичай розташовується у офісному приміщенні або в окремій робочій зоні, обладнаній комп'ютерною технікою та засобами зв'язку.

До складу робочого місця входять:

- персональний комп'ютер або ноутбук;
- монітор;
- клавіатура та комп'ютерна миша;
- телефон або гарнітура;
- мережеве обладнання;
- офісні меблі;

- програмне забезпечення для управління замовленнями;
- CRM-система;
- доступ до мережі Інтернет.

Оцінювання умов праці відіграє важливу роль у визначенні переліку потенційних небезпек [19].

Працівник більшу частину робочого дня перебуває у сидячому положенні. Основним інструментом роботи є вебсайт інтернет-магазину та пов'язані з ним електронні системи. Значна частина діяльності виконується в умовах постійної концентрації уваги та необхідності одночасно контролювати декілька процесів.

Умови праці менеджера належать до офісного типу, однак вони супроводжуються значним психоемоційним та інформаційним навантаженням. Для ефективної роботи велике значення мають параметри освітлення, мікроклімату, ергономіка меблів та правильна організація робочого простору.

Одним із найбільш характерних факторів є постійна робота з великим обсягом інформації. Менеджер повинен швидко обробляти замовлення, відповідати на запити клієнтів, перевіряти наявність товарів та контролювати правильність внесених даних. Робота у режимі багатозадачності поступово створює значне навантаження на нервову систему.

Додатковим джерелом напруження є постійна взаємодія з клієнтами. Працівник може стикатися зі скаргами, конфліктними ситуаціями, високими вимогами покупців та необхідністю оперативно вирішувати проблеми. Це може призводити до нервового виснаження, дратівливості та професійного вигорання.

Значна частина робочих операцій має повторюваний характер. Постійне оформлення замовлень, введення даних та перевірка інформації можуть викликати монотонність праці та поступове зниження концентрації уваги.

Через тривалу роботу за комп'ютером працівник постійно перебуває у малорухомому стані. Це створює статичне навантаження на м'язи спини, шиї та плечового поясу. За недостатньої рухової активності можуть виникати болі у спині, дискомфорт у шиї та порушення постави.

Працівник тривалий час взаємодіє з електронними каталогами, текстовою інформацією та графічними елементами вебсайту. Постійна концентрація зору на екрані монітора може викликати втому очей, сухість слизової оболонки, головний біль та тимчасове погіршення зорового комфорту.

Особливо сильне навантаження виникає у разі недостатнього освітлення, неправильно налаштованого монітора або тривалої безперервної роботи без перерв.

У приміщенні можуть виникати несприятливі параметри мікроклімату. Недостатня вентиляція, сухе повітря, підвищена температура або нестача природного освітлення негативно впливають на самопочуття працівників та можуть сприяти швидкій втомі.

У разі роботи у відкритому офісному просторі додатковим фактором стає шумове навантаження. Телефонні розмови, робота офісної техніки та постійна комунікація між працівниками можуть ускладнювати концентрацію уваги.

Робота менеджера пов'язана з використанням комп'ютерної та офісної техніки, тому існує ризик ураження електричним струмом у разі несправності обладнання або порушення правил експлуатації.

Також можливі перегрів електронного обладнання, коротке замикання або виникнення пожежонебезпечної ситуації при перевантаженні електромережі.

### 4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження

Оцінка професійних ризиків дозволяє своєчасно виявляти фактори, які можуть негативно впливати на працівників у процесі виконання роботи. Її головне призначення полягає у тому, щоб не лише визначити наявність небезпеки, а й зрозуміти, наскільки серйозними можуть бути її наслідки та які умови сприяють виникненню небезпечної ситуації [20, 21].

У сучасних умовах оцінка ризиків розглядається як інструмент попередження проблем. Вона дозволяє підприємству заздалегідь визначити слабкі місця в організації праці, роботі обладнання або стані виробничого середовища та вживати заходів ще до виникнення травм, аварій чи професійних захворювань.

Одним із головних завдань оцінки ризиків є виявлення всіх можливих небезпек, які можуть бути присутніми на робочому місці. При цьому враховуються не лише очевидні технічні небезпеки, а й психофізіологічні, організаційні та ергономічні фактори. Для працівників офісної та інформаційної сфери це особливо важливо, оскільки негативний вплив багатьох факторів накопичується поступово та не завжди помітний на початкових етапах.

Ще одним важливим завданням є визначення причин виникнення ризиків. Під час оцінювання аналізується, через які саме умови або дії може виникнути небезпечна ситуація. Це дозволяє не лише усунути наслідки проблеми, а й впливати безпосередньо на джерело небезпеки.

Оцінка ризиків також необхідна для встановлення пріоритетності заходів безпеки. На практиці не всі ризики мають однаковий рівень небезпеки, тому підприємство повинно визначати, які фактори потребують першочергового усунення або контролю.

Важливою задачею оцінювання є підвищення загального рівня організації праці. Аналіз ризиків дозволяє покращувати ергономіку робочих місць, оптимізувати трудові процеси, раціонально розподіляти навантаження між працівниками та створювати більш комфортні умови праці.

Правильне проведення оцінки ризиків потребує комплексного підходу. Насамперед необхідно детально проаналізувати робоче місце, характер виконуваних робіт, обладнання, умови виробничого середовища та режим праці працівника. Важливо враховувати не лише стандартні робочі ситуації, а й можливі нестандартні або аварійні умови.

Під час оцінювання необхідно [20, 21]:

- визначити всі потенційні небезпеки;
- встановити джерела ризику;
- оцінити ймовірність виникнення небезпечної події;
- визначити можливі наслідки для працівника;
- встановити рівень ризику;
- розробити заходи щодо його зниження.

Для якісного аналізу важливо використовувати реальні дані про умови праці. До процесу бажано залучати працівників, які безпосередньо виконують роботу, оскільки саме вони найкраще знають особливості трудового процесу та можуть вказати на небезпеки, які не завжди помітні під час загального огляду.

Правильне трактування результатів оцінки ризиків має велике значення для ефективності всієї системи охорони праці. Їх основне призначення – допомогти підприємству приймати обґрунтовані рішення щодо покращення умов праці та попередження небезпечних ситуацій.

Високий рівень ризику свідчить про необхідність термінового впровадження заходів безпеки або зміни організації роботи. Середній рівень ризику вказує на потребу постійного контролю та профілактичних заходів.

Низький рівень ризику не означає повної відсутності небезпеки, а лише свідчить про відносно невелику ймовірність виникнення негативних наслідків.

Під час аналізу результатів важливо враховувати не лише окремі фактори, а й їх поєднання. У реальних умовах декілька незначних ризиків можуть одночасно впливати на працівника та створювати більш серйозне навантаження, ніж кожен із них окремо.

Для проведення оцінки ризиків використовуються різні методи. Найбільш поширеним є метод матриці ризиків, який дозволяє співставляти ймовірність виникнення небезпечної події та тяжкість її наслідків. Такий метод є зручним для практичного застосування та забезпечує достатню наочність результатів.

Найбільш ефективними вважаються комбіновані підходи, які дозволяють враховувати одночасно технічні, організаційні та психофізіологічні аспекти праці.

Таблиця 4.1 – Оцінка ризику небезпеки «психоемоційне перенапруження»

Показник	Характеристика
Небезпечний фактор	Психоемоційне перенапруження
Причини виникнення	Робота з великою кількістю клієнтів, конфліктні ситуації, високий темп роботи
Можливі наслідки	Стрес, емоційне виснаження, професійне вигорання
Імовірність виникнення	3
Тяжкість наслідків	2
Рівень ризику	6
Оцінка ризику	Підвищений ризик

Рекомендовані заходи	Раціональний розподіл навантаження, перерви, підтримання сприятливого психологічного клімату
----------------------	----------------------------------------------------------------------------------------------

Для зменшення психоемоційного навантаження необхідно забезпечити більш раціональний розподіл робочих завдань та уникати надмірної багатозадачності. Працівникам доцільно організовувати короткі перерви протягом робочого дня для відновлення концентрації уваги та зниження нервового напруження.

Доцільно також використовувати сучасні інформаційні системи, які дозволяють автоматизувати частину рутинних операцій та зменшити навантаження на працівника.

Таблиця 4.2 – Оцінка ризику небезпеки «тривала робота в сидячому положенні»

Показник	Характеристика
Небезпечний фактор	Тривала сидяча робота
Причини виникнення	Постійне використання комп'ютера та недостатня рухова активність
Можливі наслідки	Біль у спині, м'язова втома, порушення постави
Імовірність виникнення	3
Тяжкість наслідків	2
Рівень ризику	6
Оцінка ризику	Підвищений ризик
Рекомендовані заходи	Ергономічне облаштування робочого місця, фізичні вправи, перерви під час роботи

Для зменшення психоемоційного навантаження необхідно забезпечити більш раціональний розподіл робочих завдань та уникати надмірної

багатозадачності. Працівникам доцільно організовувати короткі перерви протягом робочого дня для відновлення концентрації уваги та зниження нервового напруження.

Важливе значення має підтримання сприятливої атмосфери у колективі та зниження рівня конфліктних ситуацій під час роботи з клієнтами. Доцільно також використовувати сучасні інформаційні системи, які дозволяють автоматизувати частину рутинних операцій та зменшити навантаження на працівника.

Таблиця 4.3 – Оцінка ризику небезпеки «навантаження на органи зору»

Показник	Характеристика
Небезпечний фактор	Перевантаження органів зору
Причини виникнення	Тривала робота з монітором та електронною інформацією
Можливі наслідки	Втома очей, головний біль, погіршення зору
Імовірність виникнення	3
Тяжкість наслідків	2
Рівень ризику	6
Оцінка ризику	Підвищений ризик
Рекомендовані заходи	Правильне освітлення, налаштування монітора, регулярний відпочинок для очей

Для зменшення навантаження на органи зору необхідно забезпечити достатній рівень освітлення робочого місця та правильно налаштувати параметри монітора. Яскравість екрана повинна відповідати рівню освітлення приміщення.

Працівникам рекомендується періодично робити перерви для відпочинку очей та змінювати фокус зору під час тривалої роботи за комп'ютером. Доцільно також використовувати сучасні монітори з якісною передачею зображення та дотримуватися оптимальної відстані до екрана.

## Висновки до розділу

У процесі виконання розділу було розглянуто основні аспекти організації охорони праці для менеджера інтернет-магазину спортивного одягу, який використовує вебсайт для обробки замовлень та взаємодії з клієнтами. Було охарактеризовано значення охорони праці у сучасному світі, особливості розвитку системи безпеки праці в Україні та роль міжнародних стандартів у вдосконаленні національного законодавства.

Було проаналізовано професійну діяльність працівника, особливості організації його робочого місця та умови праці. Також було проведено аналіз небезпечних та шкідливих факторів, які можуть бути присутніми на робочому місці працівника. Основну увагу приділено психофізіологічним, ергономічним, фізичним та технічним ризикам, що виникають у процесі використання комп'ютерної техніки та електронних інформаційних систем.

Окремо було розглянуто значення оцінки професійних ризиків як важливого елемента системи управління охороною праці. Проведено оцінювання ризиків методом матриці для найбільш характерних небезпек, пов'язаних із професійною діяльністю менеджера. Для кожного з розглянутих ризиків були запропоновані практичні рекомендації щодо покращення умов праці, зниження рівня професійного навантаження та підтримання працездатності працівників.

## ВИСНОВКИ

У кваліфікаційній роботі бакалавра вирішено актуальну науково-практичну проблему. На основі проведеного аналізу, математичного моделювання та експериментальних досліджень отримано такі основні результати:

1. Проаналізовано предметну область та виявлено галузеві проблеми: У ході системного аналізу сучасного стану електронної комерції (e-commerce) у сегменті спортивного ритейлу встановлено, що головним деструктивним чинником для галузі є високий показник повернення товарів (від 30% до 40%), зумовлений похибками самостійного вибору розміру покупцями. Доведено неефективність традиційних статичних таблиць виробників і обґрунтовано необхідність впровадження інтелектуальних систем персоналізації, які автоматично зіставляють біометричні дані людини з параметрами лекал конкретних брендів.

2. Розроблено та математично обґрунтовано інтелектуальний алгоритм: Спроектовано математичну модель оцінки посадки одягу, в основу якої покладено модифікований метод багатовимірної зваженої евклідової відстані в просторі ознак  $R^3$ . На відміну від існуючих лінійних аналогів, запропонований алгоритм використовує динамічну систему вагових коефіцієнтів  $(w_1, w_2, w_3)$ , що адаптуються під специфіку крою конкретної категорії одягу (наприклад, пріоритет обхвату грудей для спортивних топів або обхвату стегон для легінсів). Це дозволило розраховувати інтегральний показник сумісності (Fit Score) як нелінійну функцію штрафів за відхилення від медіани розмірного інтервалу.

3. Спроектовано стійку архітектуру та оптимізовано базу даних: Обґрунтовано та реалізовано розподілену клієнт-серверну архітектуру безстанового (Stateless) типу. Базу даних розгорнуто в об'єктно-реляційній СКБД PostgreSQL, її структуру нормалізовано до рівня 3NF, що виключило

надлишковість даних та аномалії оновлення. Завдяки впровадженню B-Tree індексації за композитними ключами (бренд-категорія), обчислювальну складність операцій пошуку розмірних матриць у каталозі було знижено з лінійної  $O(N)$  до логарифмічної  $O(\log N)$ , забезпечивши високу швидкість обробки транзакцій.

4. Програмно реалізовано високонавантажений вебкомплекс: На основі обраного технологічного стеку розроблено повнофункціональний програмний продукт. Серверне ядро реалізовано на базі асинхронної платформи Node.js (Express.js), яка ефективно справляється з високою щільністю одночасних запитів (High Concurrency). Користувацький інтерфейс побудовано за концепцією SPA за допомогою бібліотеки React (Vite) з використанням архітектури Virtual DOM, що дозволило досягти миттєвого відгуку UI та швидкості рендерингу на рівні 60 FPS. Безпеку даних користувача інтегровано через конвеєр Middleware із використанням криптографічного алгоритму bcrypt та токенів JWT.

5. Експериментально підтверджено практичну цінність розробки: Проведене симуляційне тестування створених модулів на реальних антропометричних профілях підтвердило стабільність та високу точність обчислювального ядра. Система продемонструвала здатність гнучко реагувати на аномалії в метриках тіла (успішно блокуючи помилкові рекомендації та знижуючи Fit Score до 54% при критичних диспропорціях), зберігаючи при цьому швидкість відповіді сервера (Latency) в межах 50 мс. Створений програмний комплекс є повністю відчуженим інструментом, готовим до безшовної інтеграції в діючі комерційні e-commerce платформи через REST API.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глибовець М. М., Антощук С. Г. Інформаційні системи та технології в електронній комерції: архітектурні підходи та реляційні бази даних. Журнал обчислювальної та прикладної математики. 2024. № 2 (145). С. 45–58.
2. Ковальчук О. В., Сидоренко Ю. М. Порівняльний аналіз асинхронних вебфреймворків для високонавантажених інтелектуальних систем. Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. 2023. № 1 (9). С. 112–119.
3. Шутеев Д. С., Левіков Ю.В. Порухення авторських прав великими мовними моделями // Інформаційні технології: теорія і практика: Матеріали ІІІ (ІХ) Міжнародної інтернет-конференції здобувачів вищої освіти і молодих учених (Харків-Запоріжжя-Дніпро, 25–27 березня 2026 р.) / М-во освіти науки України, ХНУМГ ім. О. М. Бекетова [Електронний ресурс] Електрон. дані. – Дніпро : Свідлер А.Л., 2026. - С.406-409 - Режим доступу: <https://knit.kname.edu.ua/en/2-uncategorised/324-conference-2026-en>.
4. Марченко А. І., Федорук В. М. Особливості проектування та оптимізації реляційних баз даних PostgreSQL для аналітичних вибірок у реальному часі. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. 2023. Т. 34 (73), № 4. С. 201–208.
5. Ткаченко Д. В., Романюк О. В. Реалізація SPA-застосунків на базі бібліотеки React: оптимізація рендерингу та керування станом інструментами Virtual DOM. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2025. № 51. С. 134–142.
6. Шевченко М. О. Інтелектуальні алгоритми визначення відповідності розмірних сіток одягу в інтернет-торгівлі на основі

антропометричних векторів. Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. 2024. Вип. 2 (29). С. 67–75.

7. Becattini, F., Divitiis, L. D., Baecchi, C., & Vimbo, A. D. Fashion recommendation based on style, context, and multi-dimensional bodily fit metrics. *International Journal of Computer Vision*. 2023. Vol. 131, No. 5. P. 1145–1162.

8. Chakraborty, S., & Hoque, M. Fashion Recommendation Systems, Models and Methods: A Comprehensive Review of Fit Likelihood Classifiers. *IEEE Transactions on Artificial Intelligence*. 2024. Vol. 5, No. 2. P. 310–327.

9. Challa, N. P., Sathwik, A. S., & Kiran, J. C. Smart Fashion Recommendation System using Deep FashionNet and Anthropometric Constraints. *EAI Endorsed Transactions on Scalable Information Systems*. 2023. Vol. 10, No. 4. e12.

10. Haverbeke, M. *Eloquent JavaScript: A Modern Introduction to Programming* (4th ed.). No Starch Press, 2024. 496 p. (Присвячено сучасним стандартам ES6+ та архітектурі асинхронного циклу подій у Node.js).

11. Kamolnawin, P., Intalar, N., Watanapa, S., & Boonchoo, T. Plus-Size Clothing Recommendation System Based on Anthropometric Data Using Proportional Distance Algorithms. *Science & Technology Asia*. 2024. Vol. 29, No. 2. P. 114–121.

12. Kumar, A., & Sharma, R. Architectural Patterns in Decoupled Web Applications: Optimizing Node.js and React Pipelines. *Journal of Software Engineering and Applications*. 2025. Vol. 18, No. 1. P. 15–32.

13. Li, X., & Wang, Y. Intelligent clothing size and fit recommendations based on 3D human model customisation and weighted distance metrics. *Research Journal of Textile and Apparel*. 2025. Vol. 29, No. 3. P. 241–256.

14. Rodriguez, J., & Martinez, M. Machine learning-based clothing recommendation system for e-commerce: A case study of size profiling using bust,

waist, and hip measurements. *International Journal of Research and Scientific Innovation*. 2024. Vol. 11, No. 9. P. 401–414.

15. Zhang, L., Liu, Y., & Chen, H. A fashion product recommendation based on adaptive VPKNN-NET algorithm using interval-valued size grids. *Frontiers in Big Data*. 2025. Vol. 8, Article 1557779. P. 1–14.

16. Smith, T., & Jones, L. *Node.js Web Development: Server-side development with Express, Knex.js, and PostgreSQL (5th ed.)*. Packt Publishing, 2023. 384 p.

17. T.S. Novak (2020). Development Of Ukrainian Labor Protection Legislation: Certain Aspects. *Law. Human. Environment*, 11(2), 107-117. <https://doi.org/10.31548/law2020.02.013>.

18. Мельник К.Ю. Проблеми вдосконалення національного законодавства про охорону праці. – *Право і безпека*. – 2022. – № 2 (85). – С. 158-168.

19. Guidelines on Measuring the Quality of the Working Environment. – Режим доступу: [https://www.oecd.org/content/dam/oecd/en/publications/reports/2017/11/oecd-guidelines-on-measuring-the-quality-of-the-working-environment\\_g1g7ca23/9789264278240-en.pdf](https://www.oecd.org/content/dam/oecd/en/publications/reports/2017/11/oecd-guidelines-on-measuring-the-quality-of-the-working-environment_g1g7ca23/9789264278240-en.pdf).

20. П'ять кроків оцінки ризиків на виробництві. – Режим доступу: <https://ukrprofzahyst.com.ua/ua/obuchayuschie-statyi/12-pyat-shagov-v-otsenke-riskov-na-proizvodstve>.

21. Як оцінити ризики на робочому місці. – Режим доступу: <https://op.expertus.com.ua/consultations/46412>.

## Додатки

Додаток 1. Код Головної сторінки.

```
import React, { useState, useEffect } from 'react';
import { fetchRecommendations } from '../services/api';
import ProductCard from '../components/ProductCard';

const HomePage = () => {
  const [products, setProducts] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const [categoryFilter, setCategoryFilter] = useState('all');

  useEffect(() => {
    const loadData = async () => {
      try {
        setLoading(true);
        const data = await fetchRecommendations();
        setProducts(data);
        setError(null);
      } catch (err) {
        setError('Помилка завантаження даних. Будь ласка, перевірте авторизацію.');
      } finally {
        setLoading(false);
      }
    };
    loadData();
  });
}
```

```
}, []);
```

```
const filteredProducts = categoryFilter === 'all'
  ? products
  : products.filter(p => p.category === categoryFilter);
```

```
if (loading) return <div className="spinner">Завантаження персональних
рекомендацій...</div>;
```

```
if (error) return <div className="error-banner">{error}</div>;
```

```
return (
  <div className="homepage-container">
    <header className="catalog-header">
      <h1>Персоналізований каталог спортивного одягу</h1>
      <p>Розмір підібрано на основі вашого антропометричного профілю</p>

      <div className="filter-bar">
        <button
          className={categoryFilter === 'all' ? 'active' : ''}
          onClick={() => setCategoryFilter('all')}
        >
          Усі товари
        </button>
        <button
          className={categoryFilter === 'top' ? 'active' : ''}
          onClick={() => setCategoryFilter('top')}
        >
          Верхній одяг (Топи / Футболки)
        </button>
```

```

    <button
      className={categoryFilter === 'bottom' ? 'active' : ''}
      onClick={() => setCategoryFilter('bottom')}
    >
      Нижня частина (Легінси / Шорти)
    </button>
  </div>
</header>

<main className="products-grid">
  {filteredProducts.length === 0 ? (
    <p className="no-products">Немає товарів у цій категорії.</p>
  ) : (
    filteredProducts.map(product => (
      <ProductCard
        key={product.productId}
        product={product}
      />
    ))
  )}
</main>
</div>
);
};

export default HomePage;

```

Додаток 2. Код Сторінки авторизації.

```
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import { loginUser } from '../services/api';

const LoginPage = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [validationError, setValidationError] = useState("");
  const [isSubmitting, setIsSubmitting] = useState(false);
  const navigate = useNavigate();

  const handleLogin = async (e) => {
    e.preventDefault();
    setValidationError("");

    // Первинна семантична валідація на клієнті
    if (!email || !password) {
      setValidationError('Будь ласка, заповніть усі поля.');
```

```
      return;
    }

    try {
      setIsSubmitting(true);
      const response = await loginUser({ email, password });

      // Збереження отриманого токена безпеки в LocalStorage
      localStorage.setItem('authToken', response.token);
```

```

// Перенаправлення користувача на головну сторінку
navigate('/');
} catch (err) {
  setValidationError(err.message || 'Невірний email або пароль.');
```

```

} finally {
  setIsSubmitting(false);
}
};

return (
  <div className="auth-wrapper">
    <div className="auth-card">
      <h2>Вхід до системи підбору</h2>
      <p className="auth-subtitle">Увійдіть, щоб активувати біометричний
профіль</p>

      {validationError && <div className="form-
error">{validationError}</div>}

      <form onSubmit={handleLogin} noValidate>
        <div className="form-group">
          <label htmlFor="email">Електронна пошта (Email)</label>
          <input
            id="email"
            type="email"
            value={email}
            onChange={(e) => setEmail(e.target.value)}
            placeholder="example@domain.com"

```

```
        disabled={isSubmitting}
      />
    </div>

    <div className="form-group">
      <label htmlFor="password">Пароль</label>
      <input
        id="password"
        type="password"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
        placeholder="••••••••"
        disabled={isSubmitting}
      />
    </div>

    <button type="submit" className="btn-submit" disabled={isSubmitting}>
      {isSubmitting ? 'Автентифікація...' : 'Увійти'}
    </button>
  </form>
</div>
</div>
);
};

export default LoginPage;
```

Додаток 3. Код системи підбору розміра.

```
/**
 * ОБЧИСЛЮВАЛЬНЕ ЯДРО СИСТЕМИ
 * Математичне моделювання відповідності посадки одягу за зваженою
 евклідовою метрикою
 */

// Конфігурація матриці вагових коефіцієнтів ознак за категоріями
 спортивного одягу
const CATEGORY_WEIGHTS = {
  top: {
    chest: 0.50, // Обхват грудей — пріоритет для верхнього одягу
    waist: 0.35,
    hip: 0.15
  },
  bottom: {
    chest: 0.00, // Обхват грудей ігнорується для штанів/легінсів
    waist: 0.40,
    hip: 0.60 // Обхват стегон — домінуючий параметр
  }
};

/**
 * Розрахунок індексу сумісності Fit Score (0% - 100%)
 * @param {Object} user - Антропометричний вектор користувача
 * @param {Object} grid - Інтервальна матриця меж конкретного розміру
 * @param {String} category - Категорія товару ('top' або 'bottom')
 * @returns {Number} - Фінальний відсоток відповідності
```

```

*/
export const calculateFitScore = (user, grid, category) => {
  // Завантаження ваг для поточної категорії; якщо категорія невідома —
  рівномірний розподіл
  const weights = CATEGORY_WEIGHTS[category] || { chest: 0.33, waist: 0.33,
  hip: 0.34 };
  let totalPenalty = 0;

  // Формування вектора ознак для ітеративної обробки
  const dimensions = [
    { key: 'chest', userVal: parseFloat(user.chest_circ), min:
  parseFloat(grid.min_chest), max: parseFloat(grid.max_chest) },
    { key: 'waist', userVal: parseFloat(user.waist_circ), min:
  parseFloat(grid.min_waist), max: parseFloat(grid.max_waist) },
    { key: 'hip', userVal: parseFloat(user.hip_circ), min: parseFloat(grid.min_hip),
  max: parseFloat(grid.max_hip) }
  ];

  for (const dim of dimensions) {
    const weight = weights[dim.key];

    // Якщо вага ознаки дорівнює 0, вона виключається з обчислення
    геометричної відстані
    if (weight === 0) continue;

    // Розрахунок оптимальної (медіанної) точки всередині заданої розмірної
    сітки бренді
    const targetOptimal = (dim.min + dim.max) / 2;
  }
}

```

```
// Визначення технологічного допуску похибки (15% від медіани як критична межа)
const maxAllowedDeviation = targetOptimal * 0.15;

// Обчислення абсолютної лінійної дельти відхилення заміру від ідеалу
const deviation = Math.abs(dim.userVal - targetOptimal);

// Розрахунок нелінійного штрафу за квадратичною евклідовою метрикою з урахуванням ваги ознаки
const penalty = (Math.pow(deviation, 2) / Math.pow(maxAllowedDeviation, 2))
* weight;

totalPenalty += penalty;
}

// Конвертація накопиченої зваженої відстані-штрафу у відсоткове значення комфорту
// Використовується квадратний корінь для нормалізації масштабу
const normalizedScore = (1 - Math.sqrt(totalPenalty)) * 100;

// Обмеження вихідного значення суворим математичним інтервалом [0; 100]
const finalFitScore = Math.max(0, Math.min(100,
Math.round(normalizedScore)));

return finalFitScore;
};
```

# Додаток 4. Дизайн сторінок

