

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Відокремлений структурний підрозділ
«Тернопільський фаховий коледж Тернопільського національного технічного
університету імені Івана Пулюя»

(повне найменування вищого навчального закладу)

Відділення телекомунікацій та електронних систем

(назва відділення)

Циклова комісія комп'ютерних наук

(повна назва циклової комісії)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи
фахового молодшого бакалавра

на тему: Розробка вебсайту магазину спортивного одягу «SportStore»

Виконав: студент (ка) 4 курсу, Групи КН-423

спеціальності 122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Савчук Артем Вікторович

Керівник Цимбалюк Людмила Володимирівна

(прізвище та ініціали)

Рецензент Гладьо Сергій Володимирович

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем

Циклова комісія комп'ютерних наук

Освітньо-професійний ступінь «фаховий молодший бакалавр»

Спеціальність 122 Комп'ютерні науки

Галузь знань 12 Інформаційні технології

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Савчуку Артему Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебсайту магазину спортивного одягу «SportStore»

керівник роботи Цимбалюк Людмила Володимирівна,
(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студенткою роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: Java, JavaScript; фреймворки: Spring Boot; бібліотека React, система керування збіркою Maven, база даних H2 Database, система контролю версій Git, стандарти IEEE 29148-2018, IEEE 29119, ДСТУ 8302:2015.

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

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу вебзастосунку

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

- 2 Розробка технічного та робочого проекту
 - 2.1 Розробка структури web-застосунку
 - 2.2 Створення та верстка web-сторінок
 - 2.3 Розробка структури бази даних
 - 2.4 Програмування web-застосунку
 - 2.4.1 Реалізація клієнтської частини
 - 2.4.2 Реалізація адміністративної частини
 - 2.5 Тестування web-застосунку
- 3 Спеціальний розділ
 - 3.1 Інструкція з розміщення сайту в Інтернеті
 - 3.2 Інструкція з користування web-застосунку
 - 3.3 Інструкція з підтримки та обслуговування системи
- 4 Економічний розділ
 - 4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт
 - 4.2 Розрахунок витрат на оплату праці та відрахувань на соціальні заходи
 - 4.3 Розрахунок витрат на електроенергію
 - 4.4 Розрахунок амортизаційних відрахувань web-застосунку «SportStore»
 - 4.5 Обчислення накладних витрат
 - 4.6 Складання кошторису витрат та визначення собівартості web-застосунку «SportStore»
 - 4.7 Розрахунок ціни web-застосунку «SportStore»
 - 4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень
- 5 Охорона праці, техніка безпеки та екологічні вимоги
 - 5.1 Професійні захворювання та отруєння
 - 5.2 Шум та основні джерела шкідливого шуму
- 6 Висновки

1. Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – вебсайту.

5. Перелік графічного матеріалу:

1. Схема структурна клієнтської частини
2. UML-діаграма діяльності користувача
3. ER-діаграма бази даних вебсайту
4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОРЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	2. Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проекту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення пояснювальної записки	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист кваліфікаційної роботи	12.06.2026	
12	Захист кваліфікаційної роботи	26 .06.2026	

7. Дата видачі завдання: 05 березня 2026 р.

Студент

(підпис)

Артем САВЧУК

Керівник роботи

(підпис)

Людмила ЦИМБАЛЮК

ЗМІСТ

АНОТАЦІЯ	7
ANNOTATION	8
ВСТУП	9
1 ЗАГАЛЬНИЙ РОЗДІЛ	11
1.1 Аналітичний огляд існуючих рішень	11
1.2 Технічне завдання	12
1.2.1 Найменування та область застосування	13
1.2.2 Призначення розробки	14
1.2.3 Вимоги до програмного забезпечення	15
1.2.4 Вимоги до програмної документації	18
1.2.5 Техніко-економічні показники	20
1.2.6 Стадії та етапи розробки	21
1.2.7 Порядок контролю та прийому	23
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ	25
2.1 Розробка структури web-застосунку	25
2.2 Створення та верстка web-сторінок	29
2.3 Розробка структури бази даних	33
2.4 Програмування web-застосунку	35
2.4.1 Реалізація клієнтської частини	38
2.4.2 Реалізація адміністративної частини	40
2.5 Тестування web-застосунку	42
3 СПЕЦІАЛЬНИЙ РОЗДІЛ	46
3.1 Інструкція з розміщення сайту в Інтернеті	46
3.2 Інструкція з користування web-застосунком	48
3.3 Інструкція з підтримки та обслуговування системи	50

					2026.KBP.122.423.19.00.00 ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Савчук А.В.			Розробка вебсайту «SportStore » Пояснювальна записка	Літ.	Арк.
Перевір.		Цимбалюк Л.В.					5
Реценз.						ВСП ТФК ТНТУ КН-423п м. Тернопіль	
Н. Контр.		Приймак В.А					
Затверд.							

4	ЕКОНОМІЧНИЙ РОЗДІЛ.....	53
4.1	Визначення стадій технологічного процесу та загальної тривалості проведення робіт	53
4.2	Розрахунок витрат на оплату праці та відрахувань на соціальні заходи 55	
4.3	Розрахунок витрат на електроенергію	57
4.4	Розрахунок амортизаційних відрахувань web-застосунку «SportStore» 58	
4.5	Обчислення накладних витрат	59
4.6	Складання кошторису витрат та визначення собівартості web-застосунку «SportStore»	59
4.7	Розрахунок ціни web-застосунку «SportStore».....	60
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень 61	
5	ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ63	
5.1	Професійні захворювання та отруєння	63
5.2	Шум та основні джерела шкідливого шуму	64
	ВИСНОВКИ.....	65
	ПЕРЕЛІК ПОСИЛАНЬ	67
	ДОДАТКИ.....	68
	ДОДАТОК А. Файл SecurityConfig.java	68
	ДОДАТОК Б. Файл ProductController.java	71
	ДОДАТОК В. Файл ProductService.java.....	76
	ДОДАТОК Г. Файл AdminPanel.jsx	81
	ДОДАТОК Д. Файл ProductDetails.jsx	87
	ДОДАТОК Е. Файл api.js	94

АНОТАЦІЯ

Тема кваліфікаційної роботи: Розробка вебсайту магазину спортивного одягу «SportStore».

Метою кваліфікаційної роботи є розробка сучасного web-застосунку для продажу спортивного одягу, який забезпечує перегляд каталогу товарів, фільтрацію продукції, реєстрацію та авторизацію користувачів, додавання товарів до кошика, оформлення замовлень, а також адміністрування товарів, категорій і замовлень.

Пояснювальна записка складається з п'яти розділів.

У загальній частині описано аналітичний огляд існуючих рішень у сфері інтернет-магазинів спортивного одягу та розглянуто технічне завдання на розробку web-застосунку SportStore.

У другому розділі представлено процес розробки технічного та робочого проєкту, описано структуру web-застосунку, створення та верстку web-сторінок, розробку структури бази даних, реалізацію клієнтської частини на React, серверної частини на Spring Boot, адміністративної панелі та тестування програмного продукту.

У спеціальній частині описано інструкцію з розміщення сайту в Інтернеті, інструкцію з користування web-застосунком та інструкцію з підтримки й обслуговування системи.

Розрахунок вартості розробки програмного продукту, витрат на оплату праці, електроенергію, амортизаційних і накладних витрат, а також визначення економічної ефективності наведено в економічному розділі.

Основні питання охорони праці, техніки безпеки та екологічні вимоги під час роботи з комп'ютерною технікою розглянуто в п'ятому розділі.

Обсяг пояснювальної записки — 99 сторінок.

До складу кваліфікаційної роботи входить графічна частина, що містить структурну схему web-застосунку, UML-діаграми, ER-діаграму бази даних та інші ілюстративні матеріали проєкту.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ANNOTATION

Qualification Work Topic: Development of the SportStore Sportswear Store Website.

The purpose of the qualification work is to develop a modern web application for selling sportswear that provides product catalog browsing, product filtering, user registration and authentication, adding products to the shopping cart, order placement, as well as administration of products, categories, and orders.

The explanatory note consists of five chapters.

The general section contains an analytical review of existing solutions in the field of online sportswear stores and considers the technical specification for the development of the SportStore web application.

The second chapter presents the process of developing the technical and working project, describes the structure of the web application, the creation and layout of web pages, the database structure design, the implementation of the client-side using React, the server-side using Spring Boot, the administrative panel, and software testing.

The special section describes the instructions for deploying the website on the Internet, the user guide for the web application, and the system maintenance and support instructions.

The economic section includes the calculation of software development costs, labor costs, electricity expenses, depreciation and overhead costs, as well as the determination of the economic efficiency of the project.

The main issues of occupational safety, workplace safety, and environmental requirements related to working with computer equipment are considered in the fifth chapter.

The explanatory note contains 99 pages.

The qualification work also includes a graphical part containing the structural diagram of the web application, UML diagrams, the database ER diagram, and other illustrative project materials.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасні інформаційні технології відіграють важливу роль у розвитку електронної комерції та автоматизації процесів онлайн-продажів. Використання web-застосунків дозволяє забезпечити швидкий доступ до інформації, автоматизувати процес оформлення замовлень та підвищити зручність взаємодії користувача із системою. У сучасних умовах більшість користувачів здійснюють пошук та придбання товарів саме через мережу Інтернет, тому розробка функціональних та адаптивних web-ресурсів є актуальним напрямом у сфері інформаційних технологій.

Особливо популярними є web-магазини спортивного одягу та взуття, оскільки вони дозволяють користувачам переглядати каталог продукції, порівнювати товари, оформляти замовлення та отримувати інформацію про новинки й акції. Використання сучасних web-технологій дає можливість створювати зручні та ефективні системи з адаптивним інтерфейсом, високою швидкістю роботи та підтримкою різних типів пристроїв.

Актуальність теми кваліфікаційної роботи «Розробка web-застосунку “SportStore”» зумовлена розвитком сучасних технологій web-розробки та постійним зростанням популярності онлайн-магазинів спортивного одягу. Розробка власного web-застосунку дозволяє отримати практичні навички проєктування інтерфейсів, створення структури web-сайту, реалізації функціональних модулів та організації взаємодії користувача із системою.

Метою кваліфікаційної роботи є розробка web-застосунку «SportStore», який забезпечує перегляд спортивного одягу, пошук товарів, оформлення замовлень та адміністрування системи.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати предметну область та існуючі web-ресурси;
- спроектувати структуру web-застосунку;
- розробити інтерфейс користувача;
- реалізувати функціонал для користувачів та адміністратора;

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

- створити основні модулі програмного забезпечення;
- провести тестування web-застосунку.

Об'єктом дослідження є процес організації онлайн-системи для продажу спортивного одягу.

Предметом дослідження є методи та засоби розробки web-застосунку для перегляду та замовлення спортивного одягу.

Практичне значення роботи полягає у створенні програмного продукту, який може бути використаний для автоматизації роботи онлайн-магазину спортивного одягу та покращення взаємодії користувачів із web-ресурсом.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

У процесі виконання кваліфікаційної роботи було проведено аналіз сучасних web-магазинів спортивного одягу та взуття. Основною метою аналізу було дослідження функціональних можливостей існуючих web-ресурсів, особливостей їх інтерфейсу користувача, структури сторінок та організації взаємодії користувача із системою.

Для аналізу були обрані популярні онлайн-магазини спортивного одягу, які мають сучасний дизайн, адаптивний інтерфейс та широкий функціонал для роботи з товарами й замовленнями. Під час дослідження було проаналізовано структуру головних сторінок сайтів, систему навігації, пошук товарів, організацію каталогу продукції та принципи оформлення замовлень.

Більшість сучасних web-магазинів використовують адаптивний дизайн, що дозволяє зручно користуватися web-ресурсом як на персональних комп'ютерах, так і на мобільних пристроях. Значна увага приділяється швидкості завантаження сторінок, логічній структурі меню та простоті взаємодії користувача із системою.

У процесі аналізу було встановлено, що важливими елементами сучасного web-магазину є каталог товарів, система категорій продукції, фільтрація та пошук товарів, кошик покупок, форма оформлення замовлення та особистий кабінет користувача. Для адміністратора системи зазвичай передбачено окрему панель керування, яка забезпечує можливість додавання, редагування та видалення товарів, а також керування замовленнями користувачів.

Особливу увагу було приділено аналізу інтерфейсу користувача сучасних web-магазинів. Було визначено, що зручне розташування елементів меню, простота навігації та зрозуміла структура сторінок значно покращують комфорт роботи користувача із web-застосунком. Також важливим фактором є адаптивність інтерфейсу та коректне відображення сторінок на різних типах пристроїв.

Під час аналізу існуючих рішень були виявлені також окремі недоліки web-

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

ресурсів. До них належать перевантаженість сторінок великою кількістю елементів інтерфейсу, складна система навігації, наявність зайвих рекламних блоків та недостатня оптимізація окремих сторінок сайту.

На основі проведеного аналізу було визначено основні вимоги до web-застосунку «SportStore». Система повинна мати сучасний та зручний інтерфейс користувача, адаптивний дизайн, логічну структуру сторінок, систему пошуку та фільтрації товарів, кошик покупок і функцію оформлення онлайн-замовлень. Також у web-застосунку повинна бути реалізована адміністративна панель для керування товарами та замовленнями користувачів.

Результати проведеного аналізу стали основою для подальшого проектування структури web-застосунку, визначення функціональних можливостей системи та розробки інтерфейсу користувача.

1.2 Технічне завдання

Технічне завдання є основним документом, який визначає вимоги до розробки web-застосунку «SportStore». У технічному завданні встановлюються призначення програмного продукту, функціональні можливості, вимоги до програмного забезпечення, структури даних, інтерфейсу користувача та умов експлуатації системи.

Метою розробки є створення сучасного інтернет-магазину спортивного одягу, який забезпечить зручний перегляд каталогу товарів, пошук і фільтрацію продукції, реєстрацію та авторизацію користувачів, оформлення замовлень і адміністрування товарного асортименту. Програмний продукт повинен працювати через web-браузер та забезпечувати швидкий доступ до інформації про товари.

У даному розділі наведено основні вимоги до web-застосунку «SportStore», визначено сферу його використання, функціональні можливості, етапи розробки та порядок контролю результатів виконання проєкту.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.1 Найменування та область застосування

Повне найменування програмного продукту – web-застосунок для продажу спортивного одягу «SportStore».

Скорочене найменування програмного продукту – «SportStore».

Розроблюваний програмний продукт належить до класу web-застосунків електронної комерції та призначений для автоматизації процесів продажу спортивного одягу через мережу Інтернет. Система забезпечує можливість перегляду каталогу товарів, пошуку та фільтрації продукції, перегляду детальної інформації про товари, додавання товарів до кошика та оформлення замовлень.

Областю застосування програмного продукту є діяльність підприємств, організацій або фізичних осіб-підприємців, які здійснюють реалізацію спортивного одягу та супутніх товарів через мережу Інтернет. Web-застосунок може використовуватися як основа для створення повноцінного інтернет-магазину спортивної продукції або як інформаційна система для представлення товарного асортименту підприємства.

Основними користувачами системи є покупці та адміністратори магазину. Покупці отримують можливість переглядати каталог продукції, здійснювати пошук товарів за різними параметрами, реєструватися в системі, додавати товари до кошика та оформляти замовлення. Адміністратор має можливість керувати каталогом товарів, створювати та редагувати категорії, додавати фотографії товарів, переглядати замовлення користувачів і змінювати їх статус.

Програмний продукт реалізовано за клієнт-серверною архітектурою. Клієнтська частина розроблена з використанням бібліотеки React та виконується у web-браузері користувача. Серверна частина створена на основі фреймворку Spring Boot та забезпечує обробку запитів, взаємодію з базою даних і реалізацію бізнес-логіки системи. Для зберігання інформації використовується система керування базами даних PostgreSQL.

Web-застосунок «SportStore» є багатоплатформним рішенням і може використовуватися на персональних комп'ютерах, ноутбуках, планшетах та

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

мобільних пристроях, що мають доступ до мережі Інтернет та сучасний web-браузер. Завдяки адаптивному інтерфейсу система забезпечує коректне відображення інформації на екранах різних розмірів.

Таким чином, web-застосунок «SportStore» призначений для організації та автоматизації процесу продажу спортивного одягу через мережу Інтернет, забезпечуючи зручну взаємодію між покупцями та адміністраторами магазину.

1.2.2 Призначення розробки

Метою розробки web-застосунку «SportStore» є створення сучасного інтернет-магазину спортивного одягу, який забезпечує зручний процес вибору, перегляду та придбання товарів через мережу Інтернет. Програмний продукт повинен автоматизувати основні процеси електронної торгівлі та надати користувачам простий і зрозумілий інтерфейс для роботи з каталогом продукції.

Функціональним призначенням системи є забезпечення можливості перегляду товарів, пошуку та фільтрації продукції за визначеними параметрами, перегляду детальної інформації про товар, вибору розміру та кольору, додавання товарів до кошика й оформлення замовлень. Також система повинна забезпечувати реєстрацію та авторизацію користувачів, збереження інформації про замовлення та захист персональних даних.

Для адміністратора магазину програмний продукт призначений для керування асортиментом та підтримки актуальності інформації на сайті. Адміністратор має можливість створювати нові категорії та товари, редагувати їх характеристики, завантажувати фотографії, додавати окремі зображення для різних кольорів товару, а також переглядати та обробляти замовлення користувачів.

Експлуатаційне призначення системи полягає у забезпеченні цілодобового доступу користувачів до каталогу спортивного одягу незалежно від їхнього місцезнаходження. Web-застосунок повинен працювати через сучасні браузери та підтримувати використання на персональних комп'ютерах, планшетах і мобільних пристроях.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

Навчальне призначення проєкту полягає у практичному застосуванні сучасних технологій web-розробки, зокрема React, Spring Boot та PostgreSQL, а також набутті навичок проєктування клієнт-серверних систем, роботи з базами даних, реалізації REST API, організації авторизації користувачів та розгортання програмного забезпечення в мережі Інтернет.

Отже, розробка web-застосунку «SportStore» спрямована на створення функціонального, надійного та зручного інтернет-магазину спортивного одягу, який забезпечує ефективну взаємодію між покупцями та адміністраторами системи.

1.2.3 Вимоги до програмного забезпечення

Програмне забезпечення «SportStore» повинно забезпечувати надійну та стабільну роботу інтернет-магазину спортивного одягу, підтримувати виконання основних бізнес-процесів і надавати користувачам зручний інтерфейс для взаємодії із системою. Web-застосунок повинен працювати в режимі клієнт-серверної архітектури та забезпечувати коректну взаємодію між клієнтською частиною, сервером і базою даних.

До функціональних вимог програмного забезпечення належить забезпечення можливості перегляду каталогу товарів. Користувач повинен мати доступ до списку продукції із зазначенням основної інформації про кожен товар, включаючи назву, бренд, категорію, ціну та зображення. Система повинна підтримувати відображення товарів у вигляді карток із можливістю переходу до сторінки детального перегляду.

Програмне забезпечення повинно реалізовувати механізми пошуку та фільтрації товарів. Користувач повинен мати можливість здійснювати пошук продукції за назвою, а також використовувати фільтри за категорією, брендом і ціновим діапазоном. Результати пошуку повинні відображатися без суттєвих затримок та забезпечувати швидке знаходження необхідного товару.

Система повинна забезпечувати відображення детальної інформації про

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

товар. На сторінці товару необхідно відображати назву, опис, ціну, категорію, бренд, доступні розміри та кольори. Користувач повинен мати можливість переглядати основне зображення товару, додаткові фотографії та фотографії, прив'язані до конкретного кольору продукції.

Програмне забезпечення повинно підтримувати механізм вибору кольору товару з автоматичним оновленням галереї зображень. При виборі певного кольору система повинна відображати відповідні фотографії товару. Перемикання між фотографіями повинно здійснюватися за допомогою стрілок навігації та шляхом вибору мініатюр зображень. Це забезпечує більш наочне представлення продукції та покращує взаємодію користувача із системою.

Однією з основних функцій програмного забезпечення є робота з кошиком покупця. Користувач повинен мати можливість додавати товари до кошика, змінювати їх кількість, видаляти непотрібні позиції та переглядати загальну вартість замовлення. Система повинна зберігати обрані параметри товару, включаючи розмір і колір.

Програмний продукт повинен забезпечувати можливість реєстрації та авторизації користувачів. Під час реєстрації користувач вводить необхідні дані для створення облікового запису. Авторизація здійснюється за допомогою логіна та пароля. Для забезпечення безпеки доступу повинна використовуватися JWT-автентифікація, а паролі користувачів мають зберігатися у зашифрованому вигляді.

Система повинна забезпечувати оформлення замовлень користувачами. Після підтвердження покупки інформація про замовлення повинна зберігатися в базі даних із зазначенням переліку товарів, кількості, загальної вартості та поточного статусу замовлення. Користувач повинен мати можливість переглядати історію власних замовлень.

До функціональних вимог адміністративної частини належить можливість створення, редагування та видалення товарів. Адміністратор повинен мати доступ до панелі керування, через яку можна додавати нові товари, змінювати їх характеристики та видаляти застарілі записи.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

Програмне забезпечення повинно підтримувати завантаження графічних матеріалів. Адміністратор повинен мати можливість додавати головне зображення товару, додаткові фотографії та окремі набори фотографій для кожного доступного кольору товару. Завантажені файли повинні зберігатися на сервері та бути доступними для відображення на сторінках сайту.

До вимог надійності належить забезпечення коректної роботи системи при виникненні помилок введення даних або нештатних ситуацій. У разі виникнення помилки система повинна відображати зрозуміле повідомлення для користувача та запобігати втраті даних. Програмне забезпечення повинно забезпечувати перевірку коректності введених даних як на стороні клієнта, так і на стороні сервера.

До вимог безпеки належить захист персональних даних користувачів, розмежування прав доступу між звичайними користувачами та адміністраторами, використання механізмів авторизації та автентифікації, а також перевірка прав доступу до захищених ресурсів системи. Адміністративні функції повинні бути доступні лише користувачам із відповідними правами.

До технічних вимог належить підтримка сучасних web-браузерів, зокрема Google Chrome, Mozilla Firefox, Microsoft Edge та Opera. Користувач повинен мати можливість працювати із системою без встановлення додаткового програмного забезпечення. Для роботи достатньо наявності браузера та доступу до мережі Інтернет.

Клієнтська частина програмного продукту повинна бути реалізована з використанням бібліотеки React. Серверна частина повинна бути розроблена на основі фреймворку Spring Boot із використанням мови програмування Java. Для зберігання даних необхідно використовувати систему керування базами даних PostgreSQL. Обмін даними між клієнтською та серверною частинами повинен здійснюватися через REST API із використанням формату JSON.

Програмне забезпечення повинно підтримувати адаптивний дизайн інтерфейсу. Усі сторінки системи мають коректно відображатися на персональних комп'ютерах, планшетах і мобільних пристроях. Елементи інтерфейсу повинні

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

автоматично підлаштовуватися під розміри екрана, забезпечуючи комфортну роботу користувача незалежно від типу пристрою.

Таким чином, програмне забезпечення «SportStore» повинно забезпечувати повний цикл роботи інтернет-магазину спортивного одягу, включаючи перегляд товарів, оформлення замовлень, адміністрування каталогу та безпечну взаємодію користувачів із системою.

1.2.4 Вимоги до програмної документації

Програмна документація для web-застосунку «SportStore» повинна забезпечувати можливість ефективного використання, супроводу та подальшого розвитку програмного продукту. Документація має містити відомості про структуру системи, принципи її роботи, порядок встановлення, налаштування та експлуатації.

Основним документом проєкту є пояснювальна записка, яка містить опис предметної області, аналіз існуючих рішень, технічне завдання, особливості проєктування та реалізації програмного продукту, результати тестування, економічне обґрунтування та питання охорони праці. Пояснювальна записка повинна бути оформлена відповідно до вимог чинних стандартів та методичних рекомендацій щодо виконання кваліфікаційних робіт.

Документація повинна містити опис архітектури програмного продукту, бази даних, взаємодії клієнтської та серверної частин, а також призначення основних модулів системи. Особлива увага повинна приділятися опису функціональних можливостей web-застосунку та особливостям його використання.

Для користувачів системи повинна бути підготовлена інструкція з використання web-застосунку. В інструкції необхідно описати порядок реєстрації та авторизації, роботу з каталогом товарів, використання фільтрів, перегляд інформації про товар, додавання товарів до кошика та оформлення замовлень. Інструкція повинна містити послідовність виконання основних операцій та рекомендації щодо роботи із системою.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

Для адміністратора повинна бути створена окрема інструкція, яка містить опис роботи з адміністративною панеллю. У документації необхідно навести порядок створення та редагування товарів, керування категоріями, завантаження фотографій продукції, перегляду замовлень та зміни їх статусів. Також повинні бути описані основні можливості адміністрування та підтримки системи.

Документація повинна містити інструкцію зі встановлення та розгортання програмного продукту. У ній необхідно описати вимоги до програмного забезпечення, порядок встановлення клієнтської та серверної частин, налаштування бази даних PostgreSQL, запуск проєкту в локальному середовищі та процес розміщення системи в мережі Інтернет. Окремо необхідно навести відомості щодо використання сервісів Render і Vercel для публікації web-застосунку.

До складу документації повинні входити тестові сценарії, які використовуються для перевірки працездатності системи. Вони мають містити перелік тестових випадків, очікувані результати та результати фактичного виконання перевірок.

У додатках до пояснювальної записки доцільно розмістити основні фрагменти програмного коду, які демонструють реалізацію ключових компонентів системи. До них належать файли клієнтської частини React, серверної частини Spring Boot, SQL-скрипти створення структури бази даних, конфігураційні файли та інші програмні модулі, необхідні для розуміння принципів роботи web-застосунку.

Програмна документація повинна бути достатньою для забезпечення подальшого супроводу, модернізації та розвитку програмного продукту без необхідності детального аналізу всього програмного коду. Вона повинна надавати повне уявлення про призначення, структуру та принципи функціонування web-застосунку «SportStore».

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.5 Техніко-економічні показники

Розробка web-застосунку «SportStore» є економічно доцільною, оскільки дозволяє автоматизувати процес продажу спортивного одягу через мережу Інтернет та зменшити витрати на організацію традиційної торгівлі. Використання сучасних web-технологій забезпечує доступність програмного продукту для широкого кола користувачів та створює можливість подальшого розширення функціональних можливостей системи.

Під час розробки програмного продукту використовуються переважно безкоштовні програмні засоби та середовища розробки. Для створення клієнтської частини застосовується бібліотека React, для реалізації серверної частини використовується Spring Boot, а для зберігання даних – система керування базами даних PostgreSQL. Використання зазначених технологій дозволяє зменшити витрати на придбання ліцензійного програмного забезпечення та скоротити загальну собівартість розробки.

До основних витрат під час створення програмного продукту належать витрати на оплату праці розробника, використання комп'ютерної техніки та споживання електроенергії. Значна частина вартості проєкту формується саме за рахунок трудових витрат, пов'язаних з аналізом предметної області, проєктуванням системи, програмуванням, тестуванням та підготовкою документації.

Для розробки програмного продукту використовується персональний комп'ютер із доступом до мережі Інтернет. Додаткових витрат на спеціалізоване обладнання або програмне забезпечення не передбачається. Зберігання вихідного коду здійснюється за допомогою системи контролю версій Git та сервісу GitHub.

Розгортання програмного продукту здійснюється з використанням хмарних сервісів Render та Vercel. На етапі розробки та тестування можуть використовуватися безкоштовні тарифні плани зазначених платформ, що дозволяє суттєво знизити експлуатаційні витрати. У разі подальшого впровадження системи в комерційну діяльність можливе використання платних тарифів із розширеними

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

можливостями та підвищеною продуктивністю.

Очікуваним результатом впровадження web-застосунку є підвищення ефективності процесу продажу спортивного одягу, покращення доступності товарів для покупців та зменшення часу на обробку замовлень. Автоматизація основних бізнес-процесів дозволяє знизити навантаження на персонал та забезпечити оперативне оновлення інформації про продукцію.

Таким чином, розробка web-застосунку «SportStore» є економічно обгрунтованою, оскільки використовує сучасні технології, не потребує значних витрат на програмне забезпечення та обладнання, а також створює передумови для підвищення ефективності діяльності інтернет-магазину спортивного одягу.

1.2.6 Стадії та етапи розробки

Розробка web-застосунку «SportStore» виконується поетапно відповідно до вимог життєвого циклу програмного забезпечення. Кожен етап передбачає виконання певного комплексу робіт, спрямованих на створення якісного та функціонального програмного продукту.

Першим етапом розробки є аналіз предметної області та вивчення існуючих рішень. На цьому етапі здійснюється дослідження сучасних інтернет-магазинів спортивного одягу, визначаються їх переваги та недоліки, аналізуються потреби майбутніх користувачів і формуються основні вимоги до програмного продукту.

Наступним етапом є розробка технічного завдання. Під час виконання цього етапу визначаються мета та призначення системи, формулюються функціональні та нефункціональні вимоги, встановлюються вимоги до програмного забезпечення, документації та умов експлуатації майбутнього web-застосунку.

Після затвердження технічного завдання виконується етап проєктування системи. На даному етапі розробляється структура web-застосунку, визначаються основні модулі системи, створюються UML-діаграми, проєктується структура бази даних та встановлюються взаємозв'язки між окремими компонентами програмного продукту.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

Наступним етапом є реалізація клієнтської частини програмного забезпечення. Під час виконання цього етапу створюються сторінки користувацького інтерфейсу, реалізується адаптивний дизайн, налаштовується навігація між сторінками та забезпечується взаємодія з серверною частиною системи.

Паралельно виконується розробка серверної частини програмного продукту. На цьому етапі створюються програмні модулі для роботи з базою даних, реалізується бізнес-логіка системи, механізми авторизації та автентифікації користувачів, а також REST API для обміну даними між клієнтською та серверною частинами.

Після завершення програмування здійснюється інтеграція окремих модулів системи та їх спільне тестування. Перевіряється правильність роботи основних функцій, коректність обміну даними між компонентами, відповідність реалізованого функціоналу вимогам технічного завдання та стабільність роботи програмного забезпечення.

Наступним етапом є налагодження та усунення виявлених недоліків. На основі результатів тестування виправляються помилки, оптимізується робота системи та покращується взаємодія користувача з web-застосунком.

Після завершення тестування виконується розгортання програмного продукту в мережі Інтернет. Для цього клієнтська частина розміщується на платформі Vercel, серверна частина — на платформі Render, а база даних PostgreSQL налаштовується для роботи в хмарному середовищі. На цьому етапі також проводиться перевірка працездатності системи після публікації.

Заключним етапом є підготовка програмної документації та оформлення результатів кваліфікаційної роботи. До складу документації входять пояснювальна записка, інструкції користувача та адміністратора, тестові сценарії, а також матеріали, необхідні для подальшого супроводу та розвитку програмного продукту.

Таким чином, розробка web-застосунку «SportStore» здійснюється

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

послідовно та включає етапи аналізу, проєктування, програмування, тестування, розгортання та документування, що забезпечує створення якісного та функціонального програмного продукту.

1.2.7 Порядок контролю та прийому

Контроль якості розробки web-застосунку «SportStore» здійснюється на всіх етапах створення програмного продукту. Основною метою контролю є перевірка відповідності реалізованого функціоналу вимогам технічного завдання, забезпечення стабільної роботи системи та своєчасне виявлення можливих помилок.

Контроль виконання робіт проводиться під час аналізу предметної області, проєктування структури системи, розробки бази даних, програмування клієнтської та серверної частин, а також у процесі тестування та розгортання програмного продукту. На кожному етапі перевіряється правильність виконання поставлених завдань і відповідність отриманих результатів вимогам проєкту.

Перевірка працездатності програмного забезпечення здійснюється шляхом функціонального тестування основних можливостей системи. Особлива увага приділяється роботі каталогу товарів, системі пошуку та фільтрації, сторінкам перегляду товарів, роботі кошика, оформленню замовлень, реєстрації та авторизації користувачів, а також функціонуванню адміністративної панелі.

Під час тестування перевіряється правильність взаємодії клієнтської та серверної частин програмного продукту, коректність обробки запитів до бази даних та стабільність роботи REST API. Також виконується перевірка механізмів авторизації та розмежування прав доступу між звичайними користувачами та адміністраторами системи.

Додатково проводиться перевірка роботи системи в умовах, наближених до реальної експлуатації. Оцінюється швидкість завантаження сторінок, коректність відображення графічних матеріалів, робота механізмів навігації та стабільність функціонування web-застосунку при одночасному виконанні декількох

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

користувацьких операцій. Така перевірка дозволяє оцінити готовність програмного продукту до використання в мережі Інтернет.

Окремо здійснюється контроль правильності збереження та обробки інформації в базі даних. Перевіряється створення нових записів, редагування існуючих даних, видалення інформації та цілісність зв'язків між таблицями. Особлива увага приділяється роботі із замовленнями, користувачами, категоріями та товарами, оскільки саме ці дані є основою функціонування системи.

Окремо проводиться перевірка роботи web-застосунку на різних типах пристроїв та у сучасних браузерях. Контролюється правильність відображення сторінок на персональних комп'ютерах, планшетах і мобільних телефонах, а також коректність роботи адаптивного інтерфейсу.

Після завершення тестування виконується усунення виявлених недоліків та повторна перевірка виправлених функцій. Програмний продукт вважається готовим до експлуатації за умови успішного проходження всіх тестових сценаріїв і відсутності критичних помилок, які можуть впливати на роботу системи.

Прийом розробленого програмного продукту здійснюється на підставі результатів перевірки його працездатності та відповідності технічному завданню. Критеріями прийому є коректна робота основних функцій системи, стабільне функціонування клієнтської та серверної частин, правильна робота бази даних, успішне проходження тестування та наявність необхідної програмної документації.

Підтвердженням завершення розробки є успішне розгортання web-застосунку в мережі Інтернет та можливість його використання кінцевими користувачами. Після проведення всіх перевірок програмний продукт може бути рекомендований до експлуатації та подальшого вдосконалення.

Результатом прийому є підтвердження готовності web-застосунку «SportStore» до використання, демонстрації та подальшого розвитку відповідно до поставлених вимог і завдань кваліфікаційної роботи.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури web-застосунку

Розробка структури web-застосунку є одним із найважливіших етапів створення програмного продукту, оскільки саме на цьому етапі визначається загальна організація системи, її основні модулі, функціональні частини та взаємозв'язки між ними. Правильно спроектована структура дозволяє забезпечити зручність використання web-застосунку, спростити його подальше розширення та полегшити супровід програмного коду.

Web-застосунок «SportStore» розроблено як клієнт-серверну систему, що складається з клієнтської частини, серверної частини та бази даних. Клієнтська частина відповідає за відображення інтерфейсу користувача та взаємодію з ним. Серверна частина забезпечує обробку запитів, виконання бізнес-логіки, перевірку прав доступу та взаємодію з базою даних. База даних використовується для збереження інформації про користувачів, товари, категорії, замовлення, фотографії товарів та інші дані, необхідні для роботи системи.

Основною метою проектування структури web-застосунку є створення логічної та зрозумілої системи, яка дозволяє користувачу швидко знаходити потрібні товари, переглядати їх характеристики, додавати продукцію до кошика та оформлювати замовлення. Для адміністратора структура системи повинна забезпечувати можливість керування товарами, категоріями, зображеннями та замовленнями через окрему адміністративну панель.

Клієнтська частина web-застосунку реалізована з використанням бібліотеки React. Вона складається з окремих сторінок і компонентів, які відповідають за певні функціональні частини системи. До основних сторінок належать головна сторінка, каталог товарів, сторінка детального перегляду товару, кошик, сторінка авторизації, сторінка реєстрації, профіль користувача та адміністративна панель. Такий підхід дозволяє розділити інтерфейс на окремі логічні блоки та спростити підтримку коду.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

Серверна частина web-застосунку створена на основі Spring Boot. Вона реалізує REST API, через який клієнтська частина отримує та передає дані. Серверна частина містить контролери, сервіси, репозиторії, сутності бази даних, DTO-класи, мапери та конфігураційні класи. Контролери приймають HTTP-запити від клієнтської частини, сервіси виконують основну бізнес-логіку, репозиторії відповідають за доступ до бази даних, а сутності описують структуру даних, що зберігаються в PostgreSQL.

База даних PostgreSQL використовується для збереження всієї основної інформації web-застосунку. У базі даних зберігаються записи про користувачів, ролі, товари, категорії, замовлення, елементи замовлень, розміри, кольори та зображення товарів. Використання реляційної бази даних дозволяє забезпечити структуроване збереження інформації та підтримувати зв'язки між окремими об'єктами системи.

Структура web-застосунку передбачає розподіл користувачів за ролями. У системі виділяються два основні типи користувачів: звичайний користувач і адміністратор. Звичайний користувач має можливість переглядати товари, користуватися фільтрами, додавати товари до кошика, оформлювати замовлення та переглядати власні замовлення. Адміністратор має розширені права доступу та може керувати товарами, категоріями, фотографіями товарів і замовленнями користувачів.(див. рис 2.1)

Захист доступу до функцій системи реалізовано за допомогою Spring Security та JWT-авторизації. Після успішного входу користувач отримує токен, який використовується для виконання захищених запитів до серверної частини. Це дозволяє розмежувати доступ до функцій системи та запобігти використанню адміністративних можливостей звичайними користувачами.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

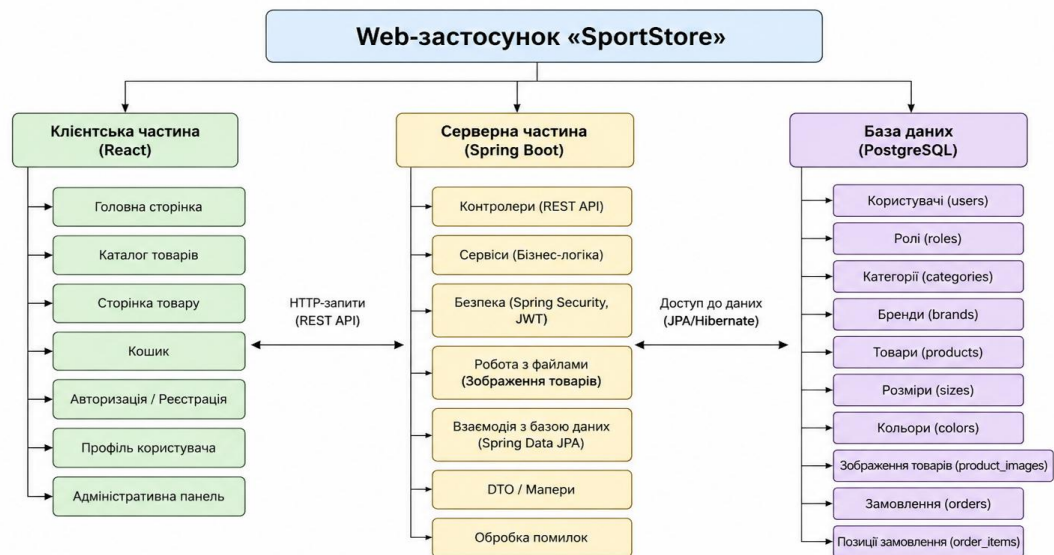


Рисунок 2.1 — Структурна схема web-застосунку «SportStore»

На структурній схемі web-застосунку доцільно відобразити три основні рівні системи: клієнтську частину, серверну частину та базу даних. У клієнтській частині необхідно показати головні сторінки web-застосунку: головну сторінку, каталог, сторінку товару, кошик, авторизацію, реєстрацію, профіль користувача та адміністративну панель. У серверній частині слід відобразити основні модулі: модуль авторизації, модуль товарів, модуль категорій, модуль замовлень, модуль завантаження зображень та модуль статистики. У базі даних потрібно показати основні сутності: користувачі, товари, категорії, замовлення, позиції замовлення та зображення товарів.

Окреме місце у структурі web-застосунку займає каталог товарів. Він є центральною частиною системи, оскільки саме через каталог користувач взаємодіє з основним асортиментом магазину. Каталог отримує дані з серверної частини, відображає товари у вигляді карток і дозволяє застосовувати фільтри за категорією, брендом та ціною. Кожен товар має окрему сторінку детального перегляду.

Сторінка товару містить розширену інформацію про продукцію. На ній відображаються назва, бренд, ціна, категорія, доступні розміри, кольори, основне зображення та галерея фотографій. Якщо для товару завантажені окремі фотографії для певного кольору, то при виборі відповідного кольору галерея змінюється. Це

підвищує зручність користування та дозволяє покупцю краще оцінити зовнішній вигляд товару.

Кошик є проміжним модулем між переглядом товарів і оформленням замовлення. У кошику користувач бачить перелік вибраних товарів, їх кількість, розмір, колір і загальну вартість. Після підтвердження покупки дані передаються на серверну частину, де створюється нове замовлення та зберігається в базі даних.

Адміністративна панель є окремою функціональною частиною web-застосунку. Вона призначена для керування вмістом інтернет-магазину. Через адміністративну панель адміністратор може створювати нові товари, редагувати характеристики наявних товарів, додавати основні та додаткові фотографії, завантажувати зображення для різних кольорів, створювати категорії, переглядати замовлення та змінювати їх статуси.

На UML-діаграмі варіантів використання необхідно відобразити двох акторів: користувача та адміністратора. Користувач взаємодіє з такими варіантами використання, як перегляд головної сторінки, перегляд каталогу, фільтрація товарів, перегляд товару, реєстрація, авторизація, додавання товару до кошика, оформлення замовлення та перегляд власних замовлень. Адміністратор, окрім частини загальних дій, має доступ до керування товарами, категоріями, зображеннями, замовленнями та перегляду статистики.

Важливою особливістю структури web-застосунку є розділення відповідальності між його частинами. Клієнтська частина не працює безпосередньо з базою даних, а лише надсилає запити до серверної частини. Серверна частина перевіряє запити, виконує необхідну логіку, звертається до бази даних і повертає результат клієнту. Такий підхід підвищує безпеку системи та спрощує її підтримку.

У сценарії оформлення замовлення користувач спочатку додає товар до кошика, після чого переходить до оформлення замовлення. Клієнтська частина формує запит до серверної, передаючи інформацію товарів, їх кількість, розмір, колір та загальну суму. Серверна частина перевіряє дані, створює замовлення, зберігає в базі і повертає користувачу повідомлення про успішне оформлення.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

Таким чином, структура web-застосунку «SportStore» побудована з урахуванням принципів клієнт-серверної архітектури, розподілу прав доступу, модульності та можливості подальшого розширення. Такий підхід забезпечує зручність використання системи, надійність її роботи та можливість подальшого розвитку програмного продукту.

2.2 Створення та верстка web-сторінок

Створення та верстка web-сторінок є одним із ключових етапів розробки web-застосунку «SportStore», оскільки саме інтерфейс забезпечує взаємодію користувача із системою. Від якості реалізації інтерфейсу залежить зручність використання програмного продукту, швидкість виконання основних операцій та загальне враження користувача від роботи із сайтом. Під час розробки web-сторінок особлива увага приділялася простоті навігації, зрозумілому розташуванню елементів, адаптивності та відповідності сучасним вимогам до дизайну інтернет-магазинів.

Клієнтська частина web-застосунку реалізована з використанням бібліотеки React. Для створення проєкту використано інструмент Vite, який забезпечує високу швидкість збірки та зручність розробки. Використання компонентного підходу дозволило розділити інтерфейс на незалежні функціональні блоки, що значно спростило підтримку та подальший розвиток програмного продукту.

У процесі розробки було створено набір сторінок, які забезпечують повний цикл взаємодії користувача із системою. До складу web-застосунку входять головна сторінка, каталог товарів, сторінка детального перегляду товару, кошик, сторінки реєстрації та авторизації, а також адміністративна панель для керування вмістом сайту.

Особливу увагу приділено проєктуванню головної сторінки, оскільки саме вона формує перше враження користувача про систему. Структура головної сторінки наведена на плакаті 1.

На структурній схемі відображено основні елементи інтерфейсу, серед яких

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

верхнє меню навігації, рекламний банер, блок популярних категорій товарів, секція рекомендованої продукції та інформаційний нижній колонтитул. Така структура забезпечує швидкий доступ до ключових функцій системи та дозволяє користувачу легко орієнтуватися на сайті.

Зовнішній вигляд головної сторінки наведено на рисунку 2.2.

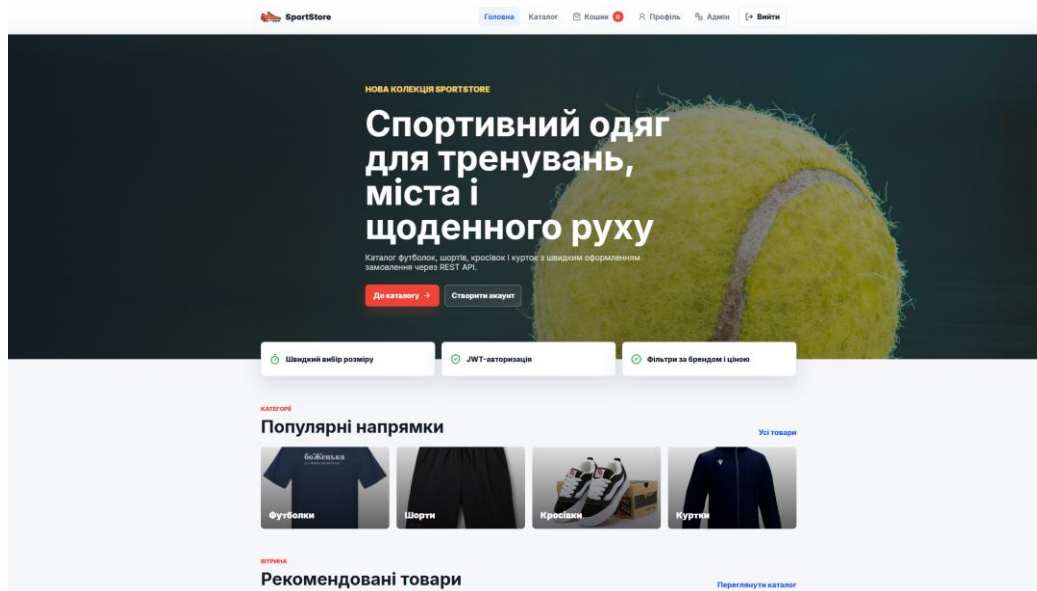


Рисунок 2.2 – Головна сторінка web-застосунку «SportStore»

На головній сторінці користувач може ознайомитися з основною інформацією про інтернет-магазин, переглянути популярні категорії товарів та швидко перейти до каталогу продукції. Верхня частина сторінки містить меню навігації, логотип та елементи керування обліковим записом користувача. Центральна частина використовується для демонстрації основних пропозицій магазину та привернення уваги відвідувачів до актуальних товарів.

Однією з основних функціональних можливостей системи є перегляд каталогу товарів. Каталог забезпечує відображення продукції у вигляді карток із фотографією, назвою, брендом та ціною. Для зручності реалізовано систему фільтрації, яка дозволяє відбирати товари за категорією, брендом і ціною. Кожен товар має окрему сторінку детального перегляду, на якій відображається повна інформація про продукцію, доступні розміри, кольори та фотографії.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

Для роботи із персональними даними користувача реалізовано сторінку реєстрації. Її зовнішній вигляд наведено на рисунку 2.3.

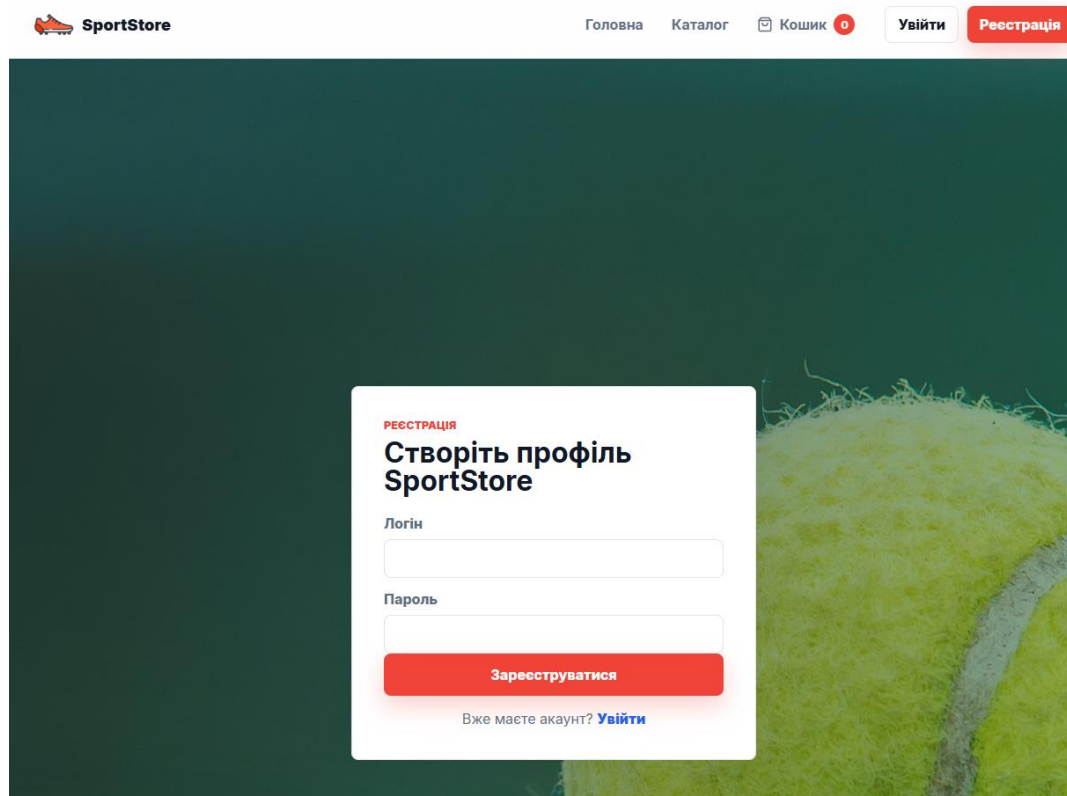


Рисунок 2.3 – Сторінка реєстрації користувача

Сторінка реєстрації містить форму введення персональних даних, необхідних для створення нового облікового запису. Після успішної реєстрації користувач отримує можливість виконувати авторизацію в системі, оформлювати замовлення та переглядати історію своїх покупок. Для підвищення зручності використання всі поля форми мають зрозумілі підписи та перевірку правильності введених даних.

Для оформлення замовлень реалізовано кошик покупця, який дозволяє переглядати обрані товари, змінювати їх кількість та контролювати загальну вартість покупки. Після підтвердження замовлення інформація передається на серверну частину системи для подальшої обробки та збереження у базі даних.

Окремою складовою web-застосунку є адміністративна панель, яка

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

використовується для керування вмістом інтернет-магазину. Її зовнішній вигляд наведено на рисунку 2.4.

The screenshot displays the administrative interface of the SportStore web application. It features a top navigation bar with status counts for orders (1 New, 0 Shipped, 0 Delivered). The main content area is divided into three primary sections:

- Новий товар (New Product):** Includes input fields for Name, Brand, Price, and Size Type. It offers checkboxes for selecting sizes (XS, S, M, L, XL, XXL) and colors (Black, White, Gray, Blue, Red, Green). There are also dropdowns for Category and image uploads for the main product and additional photos.
- Нова категорія (New Category):** A simple form with a Name field and a 'Додати категорію' (Add Category) button.
- Категорії (Categories):** A list of existing categories (Футболки, Шорти, Кросівки, Куртки) with options to edit or delete each.
- Замовлення (Orders):** A table showing order details such as ID, Date, Customer Name, Sum, City, and Status.

A red button at the bottom of the 'Новий товар' section is labeled '+ Додати товар' (Add Product).

Рисунок 2.4 – Адміністративна панель web-застосунку «SportStore»

Адміністративна панель надає доступ до функцій створення, редагування та видалення товарів, керування категоріями продукції, завантаження фотографій та перегляду замовлень користувачів. За допомогою даного інтерфейсу адміністратор може підтримувати актуальність інформації на сайті та оперативно реагувати на зміни асортименту товарів.

Для оформлення сторінок використано каскадні таблиці стилів CSS. У процесі верстки застосовувалися сучасні методи побудови інтерфейсів, зокрема Flexbox та Grid Layout. Це дозволило забезпечити правильне розташування елементів на сторінці та створити адаптивний дизайн для різних типів пристроїв.

Особливу увагу приділено мобільній версії web-застосунку. Під час адаптації інтерфейсу було змінено структуру навігаційного меню, оптимізовано відображення карток товарів та покращено зручність перегляду сторінок на смартфонах і планшетах. Завдяки використанню медіа-запитів інтерфейс

автоматично підлаштовується під розміри екрана користувача.

Таким чином, у результаті створення та верстки web-сторінок було реалізовано сучасний, адаптивний та функціональний інтерфейс користувача, який забезпечує зручну взаємодію із системою та відповідає вимогам, що висуваються до сучасних інтернет-магазинів спортивного одягу.

2.3 Розробка структури бази даних

Розробка структури бази даних є одним із найважливіших етапів створення web-застосунку «SportStore», оскільки саме база даних забезпечує зберігання, обробку та отримання інформації, необхідної для функціонування системи. Від правильності проєктування структури бази даних залежить швидкість роботи програмного продукту, цілісність інформації та можливість подальшого розширення функціональних можливостей системи.

Під час розробки web-застосунку було обрано реляційну систему керування базами даних PostgreSQL. Вибір даної СКБД обумовлений її високою продуктивністю, надійністю, безкоштовністю та широкими можливостями роботи з великими обсягами даних. PostgreSQL підтримує механізми забезпечення цілісності даних, транзакції та складні зв'язки між таблицями, що є важливим для реалізації інтернет-магазину.

Структура бази даних розроблена відповідно до функціональних вимог системи та враховує особливості роботи інтернет-магазину спортивного одягу. У базі даних зберігається інформація про користувачів, товари, категорії, замовлення, фотографії товарів та інші об'єкти, необхідні для роботи web-застосунку.

Основною таблицею системи є таблиця користувачів. Вона містить інформацію про зареєстрованих користувачів, їхні облікові записи, електронні адреси, паролі та ролі доступу. Наявність ролей дозволяє розмежувати права звичайних користувачів та адміністраторів системи. Для забезпечення безпеки паролі користувачів зберігаються у зашифрованому вигляді.

Для зберігання інформації про товари використовується окрема таблиця. У

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

ній містяться назва товару, опис, ціна, бренд, категорія, доступні розміри, кольори та інші характеристики. Кожен товар має унікальний ідентифікатор, який використовується для його пошуку та зв'язку з іншими таблицями системи.

Важливе місце у структурі бази даних займає таблиця категорій товарів. Вона дозволяє групувати продукцію за певними ознаками та спрощує реалізацію механізмів пошуку й фільтрації. Кожен товар належить до певної категорії, що забезпечується за допомогою зовнішніх ключів та зв'язків між таблицями.

Для реалізації процесу оформлення покупок використовується таблиця замовлень. У ній зберігається інформація про користувача, який створив замовлення, дату оформлення, поточний статус та загальну суму покупки. Завдяки цьому система дозволяє відстежувати стан виконання замовлень і зберігати історію покупок кожного користувача.

Оскільки одне замовлення може містити декілька товарів, додатково використовується таблиця позицій замовлення. Вона забезпечує зв'язок між товарами та конкретним замовленням, зберігаючи кількість товарів, їх вартість та інші параметри, необхідні для формування замовлення.

Окремі таблиці використовуються для роботи із зображеннями товарів. У них зберігаються посилання на основні фотографії, додаткові зображення та фотографії, прив'язані до певних кольорів продукції. Це дозволяє реалізувати розширену галерею товару та забезпечити більш зручне ознайомлення покупця з продукцією.

Під час проєктування бази даних особлива увага приділялася забезпеченню цілісності інформації. Для цього використовуються первинні та зовнішні ключі, які дозволяють підтримувати правильні зв'язки між таблицями. Такий підхід запобігає появі дубльованих або некоректних записів та забезпечує узгодженість усіх даних системи.

Для роботи з базою даних у серверній частині застосунку використовується технологія Spring Data JPA. Вона дозволяє виконувати операції створення, читання, оновлення та видалення даних за допомогою об'єктно-орієнтованого підходу.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

Використання даної технології значно спрощує взаємодію програмного коду з базою даних та скорочує обсяг службового коду.

Під час розробки структури бази даних також враховувалася можливість подальшого розвитку програмного продукту. Створена структура дозволяє без значних змін додавати нові функції, розширювати характеристики товарів, впроваджувати додаткові сервіси та інтегрувати систему з іншими програмними продуктами.

Таким чином, розроблена структура бази даних забезпечує надійне зберігання інформації, підтримує всі основні функції web-застосунку «SportStore» та створює основу для стабільної роботи системи й її подальшого розвитку.

2.4 Програмування web-застосунку

Програмування web-застосунку «SportStore» є основним етапом реалізації програмного продукту, під час якого було створено клієнтську та серверну частини системи, налаштовано взаємодію з базою даних, реалізовано механізми авторизації користувачів та забезпечено обмін інформацією між усіма компонентами програмного забезпечення.

Для реалізації клієнтської частини було використано бібліотеку React, яка дозволяє створювати сучасні односторінкові web-застосунки. Використання React забезпечило компонентний підхід до розробки інтерфейсу, що значно спростило створення та підтримку програмного коду. Кожен функціональний елемент системи реалізований у вигляді окремого компонента, який може використовуватися повторно в різних частинах застосунку.

Під час розробки клієнтської частини було створено набір сторінок і компонентів, які відповідають за відображення товарів, навігацію між розділами сайту, роботу кошика, авторизацію користувачів та взаємодію з адміністративною панеллю. Для організації маршрутизації між сторінками використовувався React Router, що дозволило реалізувати зручну навігацію без перезавантаження сторінок.

Важливою складовою клієнтської частини стала реалізація взаємодії з

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

сервером. Для отримання інформації про товари, категорії та замовлення використовуються HTTP-запити до серверної частини застосунку. Отримані дані обробляються та відображаються у відповідних компонентах інтерфейсу. Такий підхід дозволяє підтримувати актуальність інформації та забезпечує швидке оновлення даних без необхідності повторного завантаження сторінок.

Серверна частина web-застосунку реалізована на основі фреймворку Spring Boot та мови програмування Java. Використання Spring Boot дозволило значно прискорити процес розробки та забезпечити створення надійної серверної архітектури. Серверна частина відповідає за виконання бізнес-логіки, обробку запитів клієнтів, взаємодію з базою даних та контроль доступу до ресурсів системи.

Архітектура серверної частини побудована відповідно до принципів багаторівневої організації програмного забезпечення. Для обробки HTTP-запитів використовуються контролери, які приймають запити від клієнтської частини та передають їх до сервісного рівня. Сервіси реалізують основну бізнес-логіку системи та взаємодіють із репозиторіями, які забезпечують доступ до бази даних.

Для роботи з базою даних використовується технологія Spring Data JPA. Вона дозволяє виконувати операції створення, читання, оновлення та видалення даних без написання великої кількості SQL-запитів. Взаємодія між програмним кодом і таблицями бази даних здійснюється за допомогою сутностей, які відображають структуру даних у вигляді Java-класів.

Особлива увага під час програмування була приділена реалізації системи авторизації та автентифікації користувачів. Для забезпечення безпеки використовується Spring Security та JWT-токени. Після успішного входу до системи користувач отримує спеціальний токен доступу, який використовується під час виконання захищених запитів до серверної частини. Такий підхід забезпечує захист даних та обмежує доступ до адміністративних функцій стороннім користувачам.

У процесі розробки було реалізовано функціонал керування товарами. Адміністратор має можливість створювати нові товари, редагувати існуючі,

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

видаляти застарілі позиції та завантажувати фотографії продукції. Для кожного товару можуть зберігатися додаткові характеристики, що дозволяє детально описувати одяг та забезпечувати користувачів інформацією про продукцію.

Окремим напрямком програмування стала реалізація роботи із замовленнями. Після оформлення покупки інформація передається до серверної частини, де створюється новий запис у базі даних. Адміністратор може переглядати отримані замовлення та змінювати їх статус залежно від етапу обробки. Це дозволяє контролювати процес виконання замовлень та забезпечувати ефективну взаємодію між магазином і покупцем.

Для забезпечення зручності користування реалізовано механізми пошуку та фільтрації товарів. Користувач може швидко знаходити необхідну продукцію за допомогою різних параметрів, що значно покращує процес роботи з каталогом. Усі операції виконуються динамічно без повного оновлення сторінки.

Під час програмування також було реалізовано адаптивний інтерфейс користувача. Усі сторінки коректно відображаються на персональних комп'ютерах, ноутбуках, планшетах і смартфонах. Це дозволяє використовувати систему незалежно від типу пристрою та забезпечує комфортну роботу для всіх категорій користувачів.

Після завершення розробки клієнтська та серверна частини були інтегровані в єдину систему та протестовані на коректність взаємодії. У результаті було створено повноцінний web-застосунок для продажу спортивного одягу, який забезпечує роботу з товарами, користувачами та замовленнями в режимі реального часу.

Таким чином, у процесі програмування web-застосунку «SportStore» було реалізовано сучасну клієнт-серверну архітектуру, створено функціональний користувацький інтерфейс, налаштовано роботу з базою даних та забезпечено безпечну взаємодію користувачів із системою.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4.1 Реалізація клієнтської частини

Клієнтська частина web-застосунку «SportStore» реалізована з використанням бібліотеки React, яка дозволяє створювати сучасні інтерактивні web-інтерфейси та забезпечує швидке оновлення вмісту сторінок без їх повного перезавантаження. Використання React дозволило побудувати застосунок за компонентним принципом, що значно спростило процес розробки, тестування та подальшого супроводу програмного забезпечення.

Для створення проєкту було використано середовище Vite, яке забезпечує швидкий запуск застосунку в режимі розробки та оптимізовану збірку готового програмного продукту. Використання даного інструмента дозволило скоротити час розробки та підвищити продуктивність роботи над проєктом.

Архітектура клієнтської частини побудована на основі окремих сторінок і компонентів. До основних сторінок належать головна сторінка, каталог товарів, сторінка детального перегляду товару, сторінки авторизації та реєстрації, кошик покупця, сторінка замовлень користувача та адміністративна панель. Кожна сторінка реалізована як окремий React-компонент, що відповідає за виконання конкретного набору функцій.

Для організації навігації між сторінками використовується React Router. Даний інструмент забезпечує можливість переходу між розділами сайту без перезавантаження сторінки, що позитивно впливає на швидкість роботи застосунку та комфорт користувачів. Завдяки маршрутизації кожна сторінка має власну адресу, що спрощує навігацію та роботу з web-застосунком.

Одним із ключових компонентів клієнтської частини є каталог товарів. Він забезпечує відображення продукції у вигляді карток із фотографіями, назвами, брендами та цінами. Для покращення зручності користування реалізовано механізми пошуку та фільтрації, які дозволяють швидко знаходити потрібні товари серед великого асортименту продукції.

Важливим елементом інтерфейсу є сторінка детального перегляду товару. На цій сторінці користувач отримує повну інформацію про продукцію, включаючи

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

опис, ціну, доступні розміри, кольори та фотографії. Реалізовано механізм перегляду галереї зображень та автоматичної зміни фотографій залежно від обраного кольору товару.

Для збереження вибраних товарів реалізовано кошик покупця. Кошик дозволяє переглядати список доданих товарів, змінювати їх кількість, видаляти окремі позиції та контролювати загальну суму замовлення. Вибрані товари тимчасово зберігаються на стороні клієнта до моменту оформлення покупки.

Для роботи з користувачами створено сторінки реєстрації та авторизації. Форми містять перевірку правильності введених даних та забезпечують взаємодію із серверною частиною системи. Після успішної авторизації користувач отримує доступ до персональних функцій застосунку та можливість оформлення замовлень.

Окрему увагу приділено реалізації адміністративної частини інтерфейсу. Адміністратор має можливість створювати нові товари, редагувати існуючі записи, керувати категоріями, завантажувати фотографії та переглядати інформацію про замовлення користувачів. Усі адміністративні функції доступні лише після проходження авторизації з відповідними правами доступу.

Для взаємодії із серверною частиною використовується REST API. Передача даних між клієнтом і сервером здійснюється за допомогою HTTP-запитів та формату JSON. Усі операції отримання, створення, редагування та видалення інформації виконуються через спеціально розроблені API-запити.

Особливістю клієнтської частини є використання механізму збереження токена авторизації. Після успішного входу JWT-токен зберігається у локальному сховищі браузера та використовується для підтвердження прав доступу під час виконання захищених операцій. Це дозволяє підтримувати сеанс користувача без необхідності повторної авторизації при кожному переході між сторінками.

Для оформлення інтерфейсу застосовано сучасні засоби CSS-верстки. Під час створення дизайну використовувалися гнучкі контейнери Flexbox, CSS Grid, адаптивні блоки та медіа-запити. Завдяки цьому web-застосунок коректно відображається на пристроях із різними розмірами екранів.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

Таким чином, клієнтська частина web-застосунку забезпечує зручний та зрозумілий інтерфейс, підтримує необхідні функції інтернет-магазину та створює комфортні умови для роботи як звичайних користувачів, так і адміністраторів.

2.4.2 Реалізація адміністративної частини

Адміністративна частина web-застосунку «SportStore» призначена для керування основними даними системи та забезпечення підтримки актуальності інформації, яка відображається користувачам інтернет-магазину. Вона надає адміністратору доступ до спеціального набору функцій, недоступних звичайним користувачам, та дозволяє виконувати операції зі створення, редагування та видалення даних без безпосереднього втручання в базу даних або програмний код.

Реалізація адміністративної частини здійснена на основі клієнт-серверної архітектури. Користувацький інтерфейс адміністратора створено за допомогою бібліотеки React, а обробка даних та виконання бізнес-логіки реалізовані на серверній частині із використанням Spring Boot. Такий підхід забезпечує розділення функцій між клієнтом і сервером, підвищує безпеку системи та спрощує її подальший розвиток.

Доступ до адміністративної панелі надається лише авторизованим користувачам із роллю адміністратора. Перевірка прав доступу здійснюється за допомогою механізму JWT-автентифікації та засобів Spring Security. Після успішного входу до системи адміністратор отримує доступ до всіх функцій керування даними, тоді як звичайні користувачі можуть працювати лише з функціями, пов'язаними з переглядом товарів та оформленням замовлень.

Основною функцією адміністративної частини є керування товарами. Адміністратор може створювати нові товари, вказуючи їх назву, опис, вартість, бренд, категорію та інші характеристики. Також передбачена можливість редагування інформації про існуючі товари та видалення застарілих записів із системи. Завдяки цьому забезпечується постійне оновлення асортименту продукції та підтримка актуальності каталогу.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

Важливою складовою адміністративної панелі є модуль роботи із категоріями товарів. За його допомогою адміністратор може створювати нові категорії, змінювати їх назви та видаляти непотрібні записи. Наявність категорій значно спрощує структурування продукції та забезпечує зручну роботу механізмів пошуку і фільтрації товарів.

Для підвищення інформативності каталогу реалізовано функціонал завантаження фотографій товарів. Адміністратор може додавати основне зображення продукції, завантажувати додаткові фотографії та створювати окремі набори зображень для різних кольорів товару. Усі файли передаються на серверну частину та зберігаються у відповідному сховищі, після чого стають доступними для відображення на сторінках сайту.

Окремий модуль адміністративної частини відповідає за керування замовленнями користувачів. Адміністратор має можливість переглядати перелік отриманих замовлень, аналізувати їхній склад та змінювати статус виконання. Використання даного механізму дозволяє контролювати процес обробки замовлень від моменту їх створення до завершення доставки покупцю.

Для отримання інформації про стан системи реалізовано модуль статистики. Він забезпечує відображення загальної кількості товарів, категорій та замовлень, що дозволяє швидко оцінювати поточний стан інтернет-магазину та контролювати його наповнення.

У процесі розробки адміністративної частини значна увага приділялася питанням безпеки. Усі операції, пов'язані зі зміною даних, проходять перевірку прав доступу на серверній стороні. Це дозволяє запобігти несанкціонованому доступу до функцій керування системою та забезпечити захист важливої інформації.

Для взаємодії адміністративної панелі із серверною частиною використовуються REST API-запити. Кожна операція створення, редагування, видалення або отримання даних виконується через відповідний програмний інтерфейс. Передача інформації здійснюється у форматі JSON, що забезпечує

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

простоту обробки даних та високу швидкість взаємодії між компонентами системи.

Під час реалізації адміністративної частини було використано компонентний підхід, який дозволяє легко масштабувати функціональність системи. За необхідності до адміністративної панелі можуть бути додані нові модулі та інструменти керування без значної зміни існуючої архітектури програмного забезпечення.

Таким чином, реалізована адміністративна частина web-застосунку «SportStore» забезпечує ефективне керування товарами, категоріями, фотографіями та замовленнями користувачів, а також створює зручні умови для адміністрування та подальшого розвитку інтернет-магазину.

2.5 Тестування web-застосунку

Тестування є завершальним етапом розробки web-застосунку «SportStore» та виконується з метою перевірки правильності роботи всіх функціональних компонентів системи. Основним завданням тестування є виявлення можливих помилок, перевірка відповідності реалізованого функціоналу вимогам технічного завдання та забезпечення стабільної роботи програмного продукту в реальних умовах експлуатації.

Під час тестування перевірялася робота клієнтської та серверної частин застосунку, взаємодія з базою даних PostgreSQL, коректність виконання REST API-запитів, а також функціонування механізмів авторизації та розмежування прав доступу користувачів. Особлива увага приділялася перевірці основних бізнес-процесів інтернет-магазину, зокрема роботі з товарами, категоріями та замовленнями.

Першим тестуванням стала перевірка механізму реєстрації та авторизації. Було перевірено створення нових облікових записів, правильність збереження даних у базі даних, роботу механізму входу до системи та коректність формування JWT-токенів. У результаті тестування встановлено, що користувачі можуть успішно реєструватися та виконувати авторизацію без виникнення помилок.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

Наступним етапом стала перевірка роботи каталогу товарів. Було протестовано отримання списку товарів із сервера, коректність відображення інформації про продукцію, функціонування пошуку та фільтрації. Результати тестування підтвердили правильність роботи механізмів відображення товарів та відповідність отриманих даних інформації, що зберігається у базі даних.

Особливу увагу приділено тестуванню сторінки детального перегляду товару. Перевірялася коректність відображення фотографій, описів, вартості, доступних розмірів та кольорів продукції. Також було протестовано механізм перемикання фотографій залежно від вибраного кольору товару. Усі перевірки показали коректну роботу даного функціоналу.

Важливим етапом тестування стала перевірка роботи кошика покупця. Було протестовано додавання товарів до кошика, зміну кількості продукції, видалення окремих позицій та обчислення загальної вартості замовлення. Під час тестування не було виявлено помилок, які могли б вплинути на процес оформлення покупки.

Перевірка функціоналу оформлення замовлень включала створення нових замовлень, передачу даних на серверну частину та збереження інформації у базі даних. Також було протестовано можливість перегляду історії замовлень користувача та зміну статусів замовлень адміністратором. Результати тестування підтвердили правильність реалізації даного механізму.

Окремо проводилося тестування адміністративної частини системи. Було перевірено створення нових товарів, редагування існуючих записів, видалення продукції, роботу з категоріями та завантаження фотографій. Також тестувалася система перегляду та обробки замовлень користувачів. Усі адміністративні функції працювали відповідно до встановлених вимог.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

Результати тестування створення нового товару наведено на рисунку 2.16.

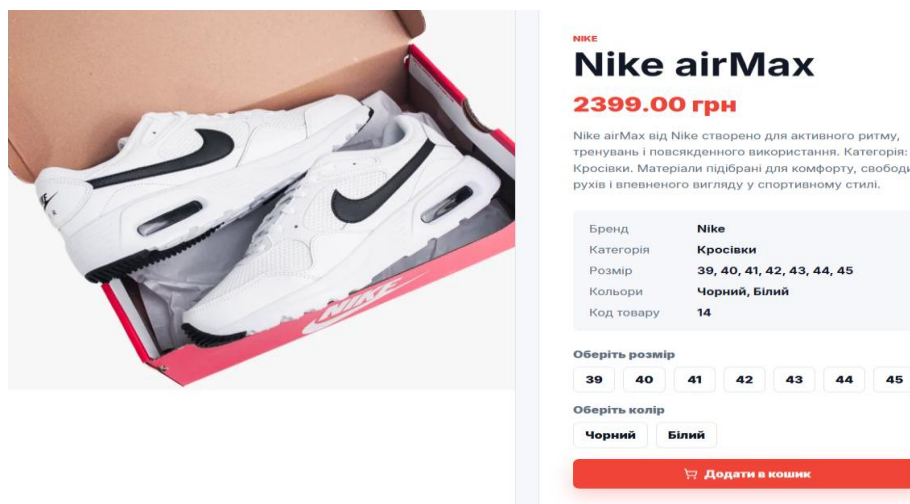


Рисунок 2.6 – Тестування створення нового товару

Під час виконання даного тесту перевірялося коректне збереження інформації про товар у базі даних та його подальше відображення у каталозі продукції. Після завершення операції новий товар успішно з'явився у системі та став доступним для перегляду користувачами.

Результати тестування редагування категорій наведено на рисунку 2.7.

Товар оновлено. Фото збережено на backend.

Рисунок 2.7 – Тестування редагування категорій

У процесі тестування було перевірено можливість зміни назв категорій, збереження внесених змін та коректне оновлення інформації на сторінках каталогу. Усі зміни успішно відображалися після оновлення даних.

Для оцінки стабільності роботи системи проводилося тестування взаємодії клієнтської та серверної частин. Перевірялися сценарії одночасного виконання декількох операцій, передача великих обсягів даних та обробка помилкових запитів. Система продемонструвала стабільну роботу та коректну обробку

виняткових ситуацій.

Крім функціонального тестування було виконано перевірку адаптивності інтерфейсу на різних пристроях. Web-застосунок успішно працював на персональних комп'ютерах, ноутбуках, планшетах і смартфонах. Усі сторінки коректно відображалися незалежно від роздільної здатності екрана.

За результатами проведених перевірок критичних помилок у роботі web-застосунку не виявлено. Усі основні функції працюють відповідно до вимог технічного завдання та забезпечують коректну взаємодію користувачів із системою.

Таким чином, проведене тестування підтвердило працездатність, надійність та готовність web-застосунку «SportStore» до експлуатації. Реалізований функціонал повністю відповідає поставленим вимогам та забезпечує стабільну роботу інтернет-магазину спортивного одягу.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розміщення сайту в Інтернеті

Розміщення web-застосунку «SportStore» у мережі Інтернет є завершальним етапом розробки програмного продукту. Основною метою даного процесу є забезпечення постійного доступу користувачів до системи через мережу Інтернет без необхідності встановлення додаткового програмного забезпечення на локальні комп'ютери.

Для розгортання проєкту було використано сучасні хмарні сервіси Vercel та Render. Клієнтська частина, реалізована за допомогою React, розміщується на платформі Vercel, яка забезпечує автоматичне збирання проєкту, швидке завантаження сторінок та підтримку сучасних web-технологій. Серверна частина, створена на основі Spring Boot, розміщується на платформі Render, яка забезпечує виконання серверного коду та взаємодію з базою даних.

Перед початком розгортання необхідно підготувати вихідний код проєкту та розмістити його у репозиторії GitHub. Використання системи контролю версій Git дозволяє централізовано зберігати всі файли проєкту та забезпечує автоматичне оновлення застосунку після внесення змін до програмного коду.

Першим етапом розміщення є публікація серверної частини застосунку. Для цього на платформі Render створюється новий web-сервіс, який підключається до репозиторію GitHub. Після вибору необхідного репозиторію виконується налаштування параметрів збірки, запуску застосунку та змінних середовища. Під час розгортання серверна частина автоматично компілюється та запускається на віддаленому сервері.

Наступним кроком є створення та налаштування бази даних PostgreSQL. Для зберігання інформації про користувачів, товари, категорії та замовлення використовується окремий екземпляр бази даних, який також розміщується на платформі Render. Після створення бази даних необхідно налаштувати параметри підключення та передати їх до серверної частини через змінні середовища.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

Після успішного розгортання серверної частини виконується розміщення клієнтського застосунку на платформі Vercel. Для цього репозиторій із React-проектом підключається до облікового запису Vercel. Під час створення нового проекту система автоматично визначає використовуваний фреймворк та виконує збірку програмного продукту.

Особливу увагу необхідно приділити налаштуванню взаємодії між клієнтською та серверною частинами системи. Для цього у файлі конфігурації клієнтського застосунку вказується адреса серверної частини, розміщеної на платформі Render. Це забезпечує коректний обмін даними між усіма компонентами web-застосунку.

Після завершення процесу розгортання виконується перевірка працездатності системи. На даному етапі тестується відкриття сторінок, отримання даних із сервера, робота авторизації користувачів, створення товарів, оформлення замовлень та інші функції програмного продукту. У разі виявлення помилок виконується їх усунення та повторне розгортання оновленої версії застосунку.

Важливою перевагою використання платформ Vercel та Render є автоматичне оновлення системи після внесення змін до репозиторію GitHub. Після завантаження нової версії програмного коду сервіси автоматично виконують збірку та повторне розгортання застосунку без необхідності ручного втручання адміністратора.

Після завершення всіх налаштувань web-застосунк стає доступним для користувачів через мережу Інтернет за постійною веб-адресою. Це дозволяє використовувати систему з будь-якого пристрою, що має доступ до мережі Інтернет, незалежно від місця розташування користувача.

Таким чином, процес розміщення web-застосунку «SportStore» у мережі Інтернет забезпечує доступність програмного продукту для кінцевих користувачів, підтримує автоматичне оновлення системи та створює умови для її стабільної експлуатації в реальному середовищі.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2 Інструкція з користування web-застосунком

Інструкція з користування web-застосунком «SportStore» призначена для ознайомлення користувачів із основними можливостями системи та порядком виконання основних операцій. Web-застосунок забезпечує можливість перегляду товарів, реєстрації користувачів, оформлення замовлень та керування даними через адміністративну панель.

Для початку роботи користувачу необхідно відкрити web-застосунок у будь-якому сучасному браузері, який підтримує технології HTML5, CSS3 та JavaScript. Після завантаження сайту відкривається головна сторінка, на якій розміщено основну інформацію про інтернет-магазин, меню навігації та посилання на основні розділи системи.

Для перегляду асортименту продукції необхідно перейти до каталогу товарів. У каталозі відображаються всі доступні товари із зазначенням їх назв, вартості та фотографій. Для швидкого пошуку необхідної продукції користувач може використовувати систему фільтрації та пошуку, яка дозволяє відбирати товари за категоріями, брендами та іншими параметрами.

Для отримання детальної інформації про товар необхідно натиснути на відповідну картку продукції. Після цього відкривається сторінка детального перегляду товару, де відображається його опис, вартість, доступні розміри, кольори та фотографії. За необхідності користувач може переглядати додаткові зображення та обирати потрібні характеристики продукції перед додаванням товару до кошика.

Для оформлення замовлення необхідно додати обраний товар до кошика. Після натискання кнопки додавання система автоматично зберігає вибраний товар та відображає його у списку покупок. У кошику користувач може переглядати всі додані товари, змінювати їх кількість або видаляти непотрібні позиції.

Для створення облікового запису необхідно перейти на сторінку реєстрації та заповнити відповідну форму. Під час реєстрації користувач вказує необхідні персональні дані та створює пароль для подальшого входу до системи. Після успішного завершення реєстрації користувач може виконати авторизацію та

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

отримати доступ до додаткових можливостей застосунку.

Для входу до системи необхідно перейти на сторінку авторизації та ввести зареєстровані облікові дані. Після успішного входу користувач отримує доступ до власного профілю, історії замовлень та інших персоналізованих функцій системи.

Після авторизації користувач може оформлювати замовлення. Для цього необхідно перейти до кошика, перевірити перелік обраних товарів та підтвердити створення замовлення. Після завершення операції інформація автоматично передається до серверної частини та зберігається у базі даних. Користувач отримує можливість переглядати інформацію про свої замовлення та контролювати їх поточний статус.

Для адміністрування системи використовується спеціальна адміністративна панель. Доступ до неї мають лише користувачі з відповідними правами доступу. Після входу адміністратор отримує можливість керувати товарами, категоріями, фотографіями та замовленнями користувачів.

Для створення нового товару адміністратор повинен перейти до відповідного розділу адміністративної панелі та заповнити форму додавання продукції. Після введення всіх необхідних даних товар автоматично зберігається у базі даних та стає доступним для відображення у каталозі.

Для редагування існуючих товарів необхідно обрати відповідний запис у списку продукції та внести необхідні зміни. Після збереження оновлена інформація автоматично відображається на сторінках сайту. Аналогічним чином адміністратор може створювати нові категорії, редагувати їх та видаляти непотрібні записи.

У розділі керування замовленнями адміністратор може переглядати список усіх отриманих замовлень, аналізувати їх склад та змінювати статус виконання залежно від етапу обробки. Це дозволяє контролювати процес виконання покупок та забезпечувати якісне обслуговування клієнтів.

Під час роботи із системою рекомендується використовувати актуальні версії браузерів та стабільне підключення до Інтернету. У разі виникнення помилок необхідно оновити сторінку або повторно виконати авторизацію в системі.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

Таким чином, web-застосунок «SportStore» забезпечує простий та зручний інтерфейс для роботи користувачів і адміністраторів, дозволяючи ефективно здійснювати пошук товарів, оформлення замовлень та керування інформацією інтернет-магазину.

3.3 Інструкція з підтримки та обслуговування системи

Підтримка та обслуговування web-застосунку «SportStore» є важливою складовою забезпечення його стабільної та безперервної роботи. Основною метою обслуговування системи є підтримка працездатності всіх функціональних модулів, своєчасне усунення можливих помилок, оновлення програмного забезпечення та забезпечення безпеки даних користувачів.

У процесі експлуатації необхідно регулярно контролювати роботу клієнтської та серверної частин застосунку. Особливу увагу слід приділяти доступності сервера, коректності виконання запитів до бази даних та працездатності основних функцій системи. Для цього рекомендується періодично перевіряти роботу каталогу товарів, механізмів авторизації, оформлення замовлень та адміністративної панелі.

Одним із важливих завдань підтримки є контроль стану бази даних PostgreSQL. Необхідно регулярно перевіряти цілісність даних, наявність актуальної інформації про товари, категорії, користувачів та замовлення. У разі виявлення помилок або пошкодження інформації необхідно виконати відновлення даних із резервної копії.

Для забезпечення надійності роботи системи рекомендується регулярно створювати резервні копії бази даних. Резервне копіювання дозволяє уникнути втрати важливої інформації у випадку технічних збоїв, помилок програмного забезпечення або несанкціонованого втручання. Створені резервні копії повинні зберігатися у безпечному місці та бути доступними для оперативного відновлення системи.

Важливим елементом супроводу є оновлення програмного забезпечення. У

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

разі внесення змін до програмного коду необхідно виконувати повторне тестування функціоналу перед розгортанням нової версії застосунку. Це дозволяє виявити можливі помилки та запобігти виникненню проблем під час експлуатації системи.

Для розгортання оновлених версій використовується репозиторій GitHub та сервіси Vercel і Render. Після завантаження змін до репозиторію запускається автоматичний процес збірки та публікації нової версії програмного продукту. Завдяки цьому підтримується актуальний стан системи та забезпечується швидке впровадження нових функцій.

Під час обслуговування необхідно контролювати використання дискового простору сервером, особливо для зберігання фотографій товарів. За потреби слід видаляти застарілі або невикористовувані файли, щоб уникнути перевищення доступних ресурсів сховища.

Для забезпечення інформаційної безпеки необхідно регулярно оновлювати паролі адміністративних облікових записів та контролювати права доступу користувачів. Також рекомендується використовувати актуальні версії програмних бібліотек та фреймворків, що дозволяє усунути відомі вразливості та підвищити рівень захисту системи.

У випадку виникнення помилок під час роботи web-застосунку необхідно проаналізувати журнали роботи сервера та повідомлення про помилки. Аналіз логів дозволяє швидко визначити причину несправності та виконати необхідні дії для її усунення. Після внесення виправлень рекомендується провести повторне тестування відповідного функціоналу.

Для підтримання високої продуктивності системи необхідно періодично перевіряти швидкість роботи клієнтської та серверної частин, оптимізувати запити до бази даних та контролювати навантаження на серверні ресурси. Такі заходи дозволяють забезпечити комфортну роботу користувачів навіть при збільшенні кількості відвідувачів сайту.

Таким чином, дотримання наведених рекомендацій щодо підтримки та обслуговування web-застосунку «SportStore» забезпечує стабільну роботу системи,

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

збереження даних користувачів, своєчасне впровадження оновлень та підтримання високого рівня надійності програмного продукту.

Окрему увагу під час супроводу системи необхідно приділяти контролю коректності роботи програмних інтерфейсів взаємодії між клієнтською та серверною частинами. Після кожного оновлення програмного забезпечення рекомендується перевіряти роботу REST API, правильність передачі даних та коректність відображення отриманої інформації на сторінках web-застосунку. Це дозволяє своєчасно виявляти помилки інтеграції між окремими модулями системи.

У процесі експлуатації також необхідно контролювати актуальність інформації, що відображається на сайті. Адміністратор повинен періодично перевіряти наявність застарілих товарів, правильність цін, коректність фотографій та відповідність описів фактичним характеристикам продукції. Своєчасне оновлення інформації позитивно впливає на якість обслуговування користувачів та підвищує довіру до інтернет-магазину.

Для забезпечення подальшого розвитку системи рекомендується здійснювати аналіз відгуків користувачів та збирати інформацію про можливі напрями вдосконалення програмного продукту. На основі отриманих даних можуть впроваджуватися нові функції, покращуватися інтерфейс користувача та оптимізуватися існуючі бізнес-процеси. Такий підхід дозволяє підтримувати конкурентоспроможність web-застосунку та забезпечувати його відповідність сучасним вимогам електронної комерції.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини даної кваліфікаційної роботи є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки web-застосунку «SportStore», прийняття рішення про доцільність його впровадження та подальшого використання.

Об'єктом розробки є web-застосунок інтернет-магазину спортивного одягу «SportStore», який призначений для автоматизації процесів продажу товарів через мережу Інтернет, забезпечення взаємодії між користувачами та адміністраторами системи, а також спрощення керування каталогом товарів і замовленнями.

Розрахунок вартості розробки виконується в декілька етапів:

- визначення стадій технологічного процесу та трудомісткості виконання робіт;
- розрахунок витрат на оплату праці виконавців та відрахувань на соціальні заходи;
- визначення витрат на електроенергію;
- розрахунок амортизаційних відрахувань;
- обчислення накладних витрат;
- складання кошторису витрат та визначення собівартості програмного продукту;
- розрахунок ціни програмного продукту;
- визначення економічної ефективності та терміну окупності розробки.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт

У даному підрозділі розглянемо основні етапи технологічного процесу розробки web-застосунку «SportStore». Для визначення загальної тривалості виконання робіт доцільно дані витрат часу за окремими операціями технологічного процесу звести в таблицю 4.1.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 – Середній час виконання робіт та стадії технологічного процесу розробки web-застосунку «SportStore»

№ 4.1	Назва операції (стадії)	Виконавець	Середній час виконання, год.
1	Планування та аналіз предметної області	Керівник проєкту (Pm)	8
		Інженер (l1)	8
2	Розробка технічного завдання	Керівник проєкту (Pm)	7
		Інженер (l1)	4
3	Проектування структури бази даних та інтерфейсу	Інженер (l1)	16
		Інженер (l2)	24
4	Розробка функціоналу клієнтської та серверної частини	Інженер (l1)	42
5	Тестування та налагодження програмного продукту	Тестувальник	8
6	Підготовка програмної документації	Інженер (l1)	2
7	Розгортання та підтримка системи	Інженер (l2)	16
8	Управління проєктом та контроль виконання робіт	Керівник проєкту (Pm)	24
Разом			159

Сумарний час виконання операцій технологічного процесу розробки web-застосунку «SportStore» становить 159 годин.

Найбільшу частину часу займає розробка функціоналу клієнтської та серверної частин системи, оскільки саме на цих етапах реалізуються основні можливості web-застосунку, забезпечується взаємодія користувача із системою та обробка даних на сервері. Значний обсяг робіт також припадає на проєктування структури бази даних, створення користувацького інтерфейсу та подальше тестування програмного продукту.

У процесі виконання робіт брали участь керівник проєкту, інженери та тестувальник. Такий розподіл обов'язків дозволяє ефективно організувати процес розробки, забезпечити контроль якості програмного продукту та скоротити терміни виконання робіт.

4.2 Розрахунок витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки web-застосунку «SportStore».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c * K_r \quad (4.1)$$

де: T_c – тарифна ставка, грн. (приймаємо для керівника проєкту (Pm) – 450 грн./год, інженера (I2) – 272 грн./год., інженера (I1) – 113 грн./год., тестувальник – 100 грн./год.; K_r – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Керівника проєкту (Pm)	$Z_{\text{осн.}} = 39 * 450 = 17\,550$ грн.
Інженера (I2)	$Z_{\text{осн.}} = 40 * 272 = 10\,880$ грн.
Інженера (I1)	$Z_{\text{осн.}} = 72 * 113 = 8136$ грн.
Тестувальник	$Z_{\text{осн.}} = 8 * 100 = 800$ грн.

Сумарна основна заробітна плата становить

$$З_{\text{осн}} = 17\,550 + 10\,880 + 8\,136 + 800 = 37\,366 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$З_{\text{дод.}} = З_{\text{осн.}} \cdot K_{\text{допл.}} \quad (4.2)$$

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

$$\text{Керівника проекту} \quad З_{\text{дод}2} = 17\,550 \cdot 0,1 = 1\,755 \text{ грн.}$$

$$\text{Інженера (І2)} \quad З_{\text{дод}3} = 10\,880 \cdot 0,1 = 1\,088 \text{ грн.}$$

$$\text{Інженера (І1)} \quad З_{\text{дод}4} = 8\,136 \cdot 0,1 = 814 \text{ грн.}$$

$$\text{Тестувальник} \quad З_{\text{дод}4} = 800 \cdot 0,1 = 80 \text{ грн.}$$

Загальна додаткова заробітна плата становить:

$$З_{\text{дод}} = 1\,755 + 1\,088 + 813,6 + 80 = 3\,737 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{\text{о.п.}}$) визначаються за формулою:

$$B_{\text{о.п.}} = З_{\text{осн.}} + З_{\text{дод.}} \quad (4.3)$$

$$B_{\text{о.п.}} = 37\,366 + 3\,737 = 41\,103 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$B_{\text{есв}} = B_{\text{оп}} \cdot 0,22 \quad (4.4)$$

$$B_{\text{есв}} = 41\,103 \cdot 0,22 = 9\,043 \text{ грн.}$$

Отримані результати свідчать про те, що витрати на оплату праці є однією з основних складових загальної вартості розробки web-застосунку. Це пояснюється значною трудомісткістю процесів аналізу вимог, проєктування, програмування, тестування та підготовки документації. Крім того, при визначенні загальних витрат було враховано додаткову заробітну плату працівників та обов'язкові відрахування на соціальні заходи відповідно до чинного законодавства України.

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ 4.2	Категорія працівник ів	Основна заробітна плата, грн.			Додатко ва заробітн а плата, грн.	ЄСВ , грн.	Всього витрати на оплату праці, грн. $6 = 3+4+5$
		Тариф на ставка, грн.	К-сть годин	Фактичн о нарах. зарплати , грн.			
		1	2	3	4	5	6
1	Кер. проєкту (Рм)	450	39	17 550	1755	4 248	23 553
2	Інженера (І2)	272	40	10 880	1088	2 633	14 601
3	Інженера (І1)	113	72	8 136	814	1 969	10 919
4	Тестуваль ник	100	8	800	80	194	1 074
Разом				37 366	3 736	9 043	50 146

Отже, загальні витрати на оплату праці та відрахування на соціальні заходи при розробці web-застосунку «SportStore» становлять 50146 грн.

4.3 Розрахунок витрат на електроенергію

Розрахуємо витрати на електроенергію, необхідну для розробки web-застосунку «SportStore». Затрати на електроенергію визначаються на основі потужності обладнання, тривалості його використання та вартості однієї кіловат-години електроенергії.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

Витрати на електроенергію визначаються за формулою:

$$З_{\text{в}} = W * T * S \quad (4.5)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання;
 S – вартість кіловат-години електроенергії (приймаємо 15, 94 грн).

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: комп'ютер – 0,82 кВт/год.

$$З_{\text{ек}} = 0,82 * 159 * 15,94 = 2078 \text{ грн.}$$

Витрати на електроенергію становлять 2078 грн.

4.4 Розрахунок амортизаційних відрахувань web-застосунку «SportStore»

Характерною особливістю використання основних фондів у процесі розробки програмного забезпечення є їх поступове зношення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом нарахування амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп'ютерна техніка належить до четвертої групи основних фондів. Для визначення амортизаційних відрахувань застосовується формула:

$$A = \frac{Б_{\text{в}} * Н_{\text{а}}}{100\%} * T, \quad (4.6)$$

де: A – амортизаційні відрахування за звітний період, грн.;

$Б_{\text{в}}$ – балансова вартість групи основних фондів на початок звітного періоду, грн.;

$Н_{\text{а}}$ – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 50000,00 грн., то сума амортизаційних відрахувань становитиме:

$$A = \frac{50\,000,00 * 0,04}{150} * 159 = 2\,120 \text{ грн.}$$

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства та створенням необхідних умов праці.

Залежно від організаційно-правової форми діяльності господарюючого суб'єкта накладні витрати можуть становити від 20 % до 60 % від суми основної та додаткової заробітної плати працівників. Для даного розрахунку приймаємо накладні витрати на рівні 40 % від загальних витрат на оплату праці.

Накладні витрати визначаються за формулою:

$$Нв = Во.п \times 0,4, (4.7)$$

де:

Нв – накладні витрати, грн;

Во.п – витрати на оплату праці, грн.

Тоді сума накладних витрат становитиме:

$$Нв = 41103 \times 0,4 = 16441 \text{ грн.}$$

Отже, накладні витрати, пов'язані з розробкою web-застосунку «SportStore», становлять 16441 грн.

4.6 Складання кошторису витрат та визначення собівартості web-застосунку «SportStore»

Для визначення собівартості web-застосунку «SportStore» необхідно врахувати всі витрати, пов'язані з процесом його розробки. До складу витрат входять витрати на оплату праці виконавців, відрахування на соціальні заходи, витрати на електроенергію, амортизаційні відрахування та накладні витрати. Узагальнення отриманих результатів дає можливість оцінити структуру витрат та визначити загальну собівартість програмного продукту. результати проведених вище розрахунків зведемо у таблицю 4.3.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.3 – Кошторис витрат на розробку web-застосунку «SportStore»

№ 4.3	Зміст витрат	Сума, грн	В % до загальної суми
1	Витрати на оплату праці	50146	71
2	Витрати на електроенергію	2078	3
3	Амортизаційні відрахування	2120	3
4	Накладні витрати	16441	23
5	Собівартість	70785	100

Собівартість розробки програмного продукту визначається за формулою:

$$C_{\text{в}} = B_{\text{о.п}} + B_{\text{с.з}} + 3e + A + H_{\text{в}} \quad (4.8)$$

$$C_{\text{в}} = 50146 + 2078 + 2120 + 16441 = 70785 \text{ грн.}$$

Отже, собівартість розробки web-застосунку «SportStore» становить 70785 грн.

4.7 Розрахунок ціни web-застосунку «SportStore»

Розрахунок ціни програмного продукту включає врахування собівартості розробки, рівня рентабельності та податку на додану вартість. Визначення ціни дозволяє оцінити ринкову вартість розробленого web-застосунку та встановити економічну доцільність його впровадження.

Ціну програмного продукту визначаємо за формулою:

$$Ц = C_{\text{в}} * (1 + P_{\text{рен}}) * (1 + \text{ПДВ}), \quad (4.9)$$

$$Ц = 70785 \times (1 + 0,3) \times (1 + 0,2) = 110\,425 \text{ грн.}$$

Отже, розрахункова ціна web-застосунку «SportStore» становить 110425 грн.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$\text{ЧТВ} = -C_B + \sum_{i=1}^t \frac{\Gamma_{\Pi}}{(1+i)^t}, \quad (4.10)$$

де: C_B – собівартість розробки; Γ_{Π} – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$\text{ЧТВ} = -70\,785 + \frac{39\,640}{(1+0,1)^1} + \frac{39\,640}{(1+0,1)^2} + \frac{39\,640}{(1+0,1)^3} = 27\,815 \text{ грн}$$

Якщо $\text{ЧТВ} \geq 0$, то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{\text{ок}} = T_{\text{пв}} + \frac{H_B}{\Gamma_{\text{пр}}} \quad (4.11)$$

де: $T_{\text{пв}}$ – період до повного відшкодування витрат, років; H_B – невідшкодовані витрати на початок року, грн.; $\Gamma_{\text{пр}}$ – грошовий потік на початок року, грн.

$$T_{\text{ок}} = 2 + \frac{1\,988}{39\,640} = 2,05 \text{ р.}$$

Отримані значення чистої теперішньої вартості та терміну окупності дозволяють оцінити економічну ефективність розробки web-застосунку «SportStore». Позитивне значення ЧТВ свідчить про доцільність реалізації проекту та можливість отримання економічного ефекту від його впровадження. Крім того, розрахований термін окупності є прийнятним для програмних продуктів даного типу та підтверджує перспективність розробки.

Всі дані внесемо в зведену таблицю 4.5.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.4 – Техніко-економічні показники web-застосунку «SportStore»

№ п/п	Показник	Значення
1	Собівартість, грн	70785
2	Плановий прибуток, грн	39640
3	Ціна, грн	110425
4	Чиста теперішня вартість, грн	27 815
5	Термін окупності, рік	2,05

Прибутковість проєкту та термін його окупності свідчать про ефективність вкладення коштів у розробку програмного продукту. Отримані показники підтверджують можливість повернення витрат на створення системи протягом 2,05 роки.

Отже, на основі проведених розрахунків можна зробити висновок, що розробка web-застосунку «SportStore» є економічно ефективною та доцільною для подальшого впровадження і використання.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

Під час розробки та експлуатації web-застосунку «SportStore» працівники виконують роботу переважно за персональним комп'ютером або ноутбуком. Такий вид діяльності належить до робіт із підвищеним навантаженням на органи зору, нервову систему та опорно-руховий апарат, тому особливу увагу необхідно приділяти профілактиці професійних захворювань і дії шкідливих факторів виробничого середовища.

5.1 Професійні захворювання та отруєння

Для працівників, які тривалий час працюють за комп'ютером, найбільш характерними є професійні захворювання, пов'язані з перенапруженням зору, статичним навантаженням на хребет і м'язи, а також психоемоційним напруженням. До можливих негативних наслідків належать:

- втома та сухість очей;
- зниження гостроти зору;
- головний біль;
- біль у шиї, плечах та попереку;
- порушення постави;
- підвищена втомлюваність та стрес.

Для профілактики професійних захворювань необхідно правильно організувати робоче місце, використовувати ергономічне крісло, розташовувати монітор на відстані 50–70 см від очей та робити регулярні перерви через кожні 45–60 хвилин роботи. Також рекомендується виконувати вправи для очей і змінювати положення тіла протягом робочого дня.

Професійні отруєння у процесі розробки програмного забезпечення зустрічаються рідко, оскільки робота не пов'язана з використанням токсичних хімічних речовин. Проте можливий негативний вплив випарів від нагрітого електронного обладнання, пилу та недостатньої вентиляції приміщення. Для

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

запобігання таким ризикам необхідно забезпечити регулярне провітрювання робочих приміщень та підтримувати чистоту робочого місця.

5.2 Шум та основні джерела шкідливого шуму

Шум є одним із факторів виробничого середовища, який може негативно впливати на працездатність людини. Навіть відносно невисокий рівень шуму здатний викликати втому, зниження концентрації уваги, дратівливість та погіршення самопочуття.

Основними джерелами шкідливого шуму в приміщеннях, де виконується розробка програмного забезпечення, є:

- системи охолодження комп'ютерів та ноутбуків (вентилятори, кулери);
- принтери та інша оргтехніка;
- кондиціонери та вентиляційні установки;
- розмови працівників і сторонній шум у приміщенні;
- шум від обладнання серверних приміщень.

Тривалий вплив шуму може призводити до зниження працездатності, підвищеної втоми та нервового напруження. Для зменшення шумового навантаження рекомендується використовувати сучасне мал шумне обладнання, своєчасно очищати та обслуговувати системи охолодження, ізолювати шумні пристрої та організовувати робочі місця таким чином, щоб джерела шуму знаходилися на максимально можливій відстані від працівників.

Таким чином, основними небезпечними та шкідливими факторами під час розробки web-застосунку «SportStore» є професійні захворювання, пов'язані з тривалою роботою за комп'ютером, та шум від електронного обладнання. Дотримання вимог ергономіки, організації праці та зниження шумового навантаження дозволяє забезпечити безпечні умови роботи та зберегти працездатність працівників.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розроблено web-застосунок «SportStore», призначений для автоматизації діяльності інтернет-магазину спортивного одягу. У результаті виконання поставлених завдань було проведено аналіз предметної області, визначено функціональні вимоги до програмного продукту та обрано сучасні технології для його реалізації.

Під час роботи було спроектовано структуру web-застосунку, розроблено клієнтську та серверну частини системи, створено структуру бази даних та реалізовано механізми взаємодії між усіма компонентами програмного забезпечення. Для реалізації клієнтської частини використано бібліотеку React, а серверну частину побудовано на основі фреймворку Spring Boot. Для зберігання даних використовується база даних PostgreSQL.

У розробленому web-застосунку реалізовано основні функції інтернет-магазину, зокрема перегляд каталогу товарів, пошук та фільтрацію продукції, перегляд детальної інформації про товари, реєстрацію та авторизацію користувачів, оформлення замовлень і перегляд історії покупок. Також створено адміністративну панель, яка забезпечує можливість керування товарами, категоріями, фотографіями та замовленнями користувачів.

У ході виконання роботи було проведено тестування програмного продукту, яке підтвердило коректність функціонування всіх основних модулів системи. Результати тестування показали стабільну роботу клієнтської та серверної частин, правильність взаємодії з базою даних та відповідність реалізованого функціоналу вимогам технічного завдання.

Також було виконано економічне обґрунтування розробки програмного продукту. Проведені розрахунки показали, що створення web-застосунку є економічно доцільним, а отримані показники собівартості, прибутку та терміну окупності підтверджують ефективність проєкту.

У спеціальній частині роботи розроблено інструкції з розміщення web-застосунку в мережі Інтернет, користування системою та її подальшого супроводу.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

Крім того, розглянуто питання охорони праці, техніки безпеки та екологічних вимог під час розробки й експлуатації програмного забезпечення.

Під час виконання кваліфікаційної роботи було отримано практичні навички проєктування, розробки та розгортання сучасних web-застосунків із використанням клієнт-серверної архітектури. Застосування сучасних технологій розробки дозволило створити програмний продукт, який відповідає актуальним вимогам до швидкодії, зручності використання та безпеки даних.

Розроблений web-застосунок має модульну структуру, що забезпечує можливість його подальшого вдосконалення та масштабування. У майбутньому система може бути доповнена новими функціональними можливостями, зокрема інтеграцією платіжних систем, розширеним особистим кабінетом користувача, системою відгуків та рейтингів товарів, а також механізмами аналітики продажів.

Використання хмарних сервісів для розміщення клієнтської та серверної частин дозволило забезпечити доступність програмного продукту через мережу Інтернет та спростити процес його подальшого супроводу. Завдяки автоматизованому розгортанню та оновленню система може швидко адаптуватися до змін і підтримувати актуальність програмного забезпечення.

Отже, поставлена мета кваліфікаційної роботи досягнута повністю. У результаті виконання роботи створено сучасний web-застосунок «SportStore», який може використовуватися для організації продажу спортивного одягу через мережу Інтернет та має можливість подальшого розширення й удосконалення функціональних можливостей.

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Фленаган Д. JavaScript. Повне керівництво. Київ : КМ-Букс, 2021. 704 с.
- 2) Фрімен Е., Робсон Е. Head First HTML та CSS. Київ : Фабула, 2020. 720 с.
- 3) Duckett J. HTML and CSS: Design and Build Websites. Indianapolis : Wiley, 2011. 490 p.
- 4) Banks A., Porcello E. Learning React. Sebastopol : O'Reilly Media, 2023. 310 p.
- 5) Walls C. Spring Boot in Action. New York : Manning Publications, 2016. 264 p.
- 6) Spring Framework Documentation. URL: <https://spring.io/projects/spring-framework> (дата звернення: 10.06.2026).
- 7) Spring Boot Documentation. URL: <https://spring.io/projects/spring-boot> (дата звернення: 10.06.2026).
- 8) React Documentation. URL: <https://react.dev> (дата звернення: 10.06.2026).
- 9) Oracle. Java Documentation. URL: <https://docs.oracle.com/en/java> (дата звернення: 10.06.2026).
- 10) PostgreSQL Documentation. URL: <https://www.postgresql.org/docs> (дата звернення: 10.06.2026).
- 11) Vercel Documentation. URL: <https://vercel.com/docs> (дата звернення: 10.06.2026).
- 12) Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. URL: <https://zakon.rada.gov.ua> (дата звернення: 10.06.2026).
- 13) Про електронну комерцію : Закон України від 03.09.2015 № 675-VIII. URL: <https://zakon.rada.gov.ua> (дата звернення: 10.06.2026).
- 14) ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2016. 16 с.
- 15) Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p.

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

ДОДАТКИ

Додаток А. Файл SecurityConfig.java

```
package com.sportshop.security;

import lombok.RequiredArgsConstructor;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.http.HttpMethod;
import org.springframework.security.authentication.AuthenticationManager;
import
org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfigurat
ion;
import
org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;
import
org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;

import static
org.springframework.security.web.util.matcher.AntPathRequestMatcher.antMatcher;

@Configuration
@EnableMethodSecurity
@RequiredArgsConstructor
public class SecurityConfig {

    private final JwtAuthFilter jwtAuthFilter;
    private final CustomUserDetailsService userDetailsService;
```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

.cors(cors -> {})

.csrf(csrf -> csrf.disable())

.authorizeHttpRequests(auth -> auth

.requestMatchers(HttpMethod.OPTIONS, "**").permitAll()

// ===== PUBLIC =====

.requestMatchers("/auth/**").permitAll()

.requestMatchers("/swagger-ui/**", "/v3/api-docs/**", "/h2-console/**").permitAll()

// ===== ADMIN API =====

.requestMatchers("/admin/**").hasAuthority("ROLE_ADMIN")

// ===== GET — ДЛЯ BCIX =====

.requestMatchers(HttpMethod.GET, "/products/**").permitAll()

.requestMatchers(HttpMethod.GET, "/categories/**").permitAll()

.requestMatchers(HttpMethod.GET, "/uploads/**").permitAll()

// ===== ADMIN: POST/PUT/DELETE =====

.requestMatchers(antMatcher(HttpMethod.POST, "/products/*/images")).hasAuthority("ROLE_ADMIN")

.requestMatchers(antMatcher(HttpMethod.POST, "/products/**/images")).hasAuthority("ROLE_ADMIN")

.requestMatchers(antMatcher(HttpMethod.POST, "/products/*/color-images")).hasAuthority("ROLE_ADMIN")

.requestMatchers(antMatcher(HttpMethod.POST, "/products/**/color-images")).hasAuthority("ROLE_ADMIN")

.requestMatchers(HttpMethod.POST, "/products",

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

```

"/products/**").hasAuthority("ROLE_ADMIN")
                                .requestMatchers(HttpMethod.PUT, "/products",
"/products/**").hasAuthority("ROLE_ADMIN")
                                .requestMatchers(HttpMethod.DELETE, "/products",
"/products/**").hasAuthority("ROLE_ADMIN")

                                .requestMatchers(HttpMethod.POST, "/categories",
"/categories/**").hasAuthority("ROLE_ADMIN")
                                .requestMatchers(HttpMethod.PUT, "/categories",
"/categories/**").hasAuthority("ROLE_ADMIN")
                                .requestMatchers(HttpMethod.DELETE, "/categories",
"/categories/**").hasAuthority("ROLE_ADMIN")

// ===== ORDERS =====
                                .requestMatchers(HttpMethod.POST,
"/orders").hasAnyAuthority("ROLE_USER", "ROLE_ADMIN")
                                .requestMatchers(HttpMethod.GET,
"/orders/my").hasAnyAuthority("ROLE_USER", "ROLE_ADMIN")
                                .requestMatchers(HttpMethod.GET, "/orders",
"/orders/*").hasAuthority("ROLE_ADMIN")
                                .requestMatchers(HttpMethod.PUT,
"/orders/*/status").hasAuthority("ROLE_ADMIN")

// інше — обов'язково з JWT
.anyRequest().authenticated()
)

.sessionManagement(sess ->
sess.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
.addFilterBefore(jwtAuthFilter, UsernamePasswordAuthenticationFilter.class)
.userDetailsService(userDetailsService);

http.headers(headers -> headers.frameOptions(frame -> frame.disable()));

return http.build();

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    }

    @Bean
    public AuthenticationManager authenticationManager(AuthenticationConfiguration
config) throws Exception {
        return config.getAuthenticationManager();
    }

    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
}

```

Додаток Б. Файл ProductController.java

```

package com.sportshop.controller;

import com.sportshop.dto.ProductDto;
import com.sportshop.entity.Category;
import com.sportshop.entity.Product;
import com.sportshop.mapper.ProductMapper;
import com.sportshop.service.CategoryService;
import com.sportshop.service.FileStorageService;
import com.sportshop.service.ProductService;
import lombok.RequiredArgsConstructor;
import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.multipart.MultipartFile;
import org.springframework.web.server.ResponseStatusException;

import java.util.ArrayList;
import java.util.List;

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

```

@RestController
@RequestMapping("/products")
@RequiredArgsConstructor
public class ProductController {

    private final ProductService productService;
    private final CategoryService categoryService;
    private final ProductMapper productMapper;
    private final FileStorageService fileStorageService;

    @PostMapping
    public ProductDto createProduct(@RequestBody ProductDto dto) {

        Category category = categoryService.findById(dto.getCategoryId());
        if (category == null) {
            throw new RuntimeException("Category not found: " + dto.getCategoryId());
        }

        Product product = productMapper.toEntity(dto, category);
        Product saved = productService.save(product);

        return productMapper.toDTO(saved);
    }

    @GetMapping
    public List<ProductDto> getAll(
        @RequestParam(required = false) Long categoryId,
        @RequestParam(required = false) String brand,
        @RequestParam(required = false) Double minPrice,
        @RequestParam(required = false) Double maxPrice
    ) {

        List<Product> products = productService.filterProducts(categoryId, brand, minPrice,
maxPrice);

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

```

return products.stream()
    .map(productMapper::toDTO)
    .toList();
}

@GetMapping("/{id}")
public ProductDto getById(@PathVariable Long id) {
    Product product = productService.findById(id);
    return productMapper.toDTO(product);
}

@PutMapping("/{id}")
public ProductDto update(@PathVariable Long id, @RequestBody ProductDto dto) {
    Category category = categoryService.findById(dto.getCategoryId());
    if (category == null) {
        throw new RuntimeException("Category not found: " + dto.getCategoryId());
    }

    Product product = productMapper.toEntity(dto, category);
    Product updated = productService.update(id, product);
    if (updated == null) {
        throw new RuntimeException("Product not found: " + id);
    }
    return productMapper.toDTO(updated);
}

@DeleteMapping("/{id}")
public void delete(@PathVariable Long id) {
    productService.deleteById(id);
}

@PostMapping(value = "{id}/images", consumes =
MediaType.MULTIPART_FORM_DATA_VALUE)

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						73
Зм.	Арк.	№ докум.	Підпис	Дата		

```

public ProductDto uploadImages(
    @PathVariable Long id,
    @RequestParam(value = "mainImage", required = false) MultipartFile mainImage,
    @RequestParam(value = "galleryImages", required = false) List<MultipartFile>
galleryImages
) {
    Product product = productService.findById(id);
    if (product == null) {
        throw new RuntimeException("Product not found: " + id);
    }

    List<MultipartFile> nonEmptyGalleryImages = galleryImages == null
        ? List.of()
        : galleryImages.stream()
            .filter(file -> file != null && !file.isEmpty())
            .toList();

    if ((mainImage == null || mainImage.isEmpty()) &&
nonEmptyGalleryImages.isEmpty()) {
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "No product
images were provided");
    }

    String mainImageUrl = fileStorageService.saveProductImage(id, mainImage);
    List<String> galleryImageUrls = new ArrayList<>();
    if (!nonEmptyGalleryImages.isEmpty()) {
        galleryImageUrls = nonEmptyGalleryImages.stream()
            .map(file -> fileStorageService.saveProductImage(id, file))
            .filter(url -> url != null)
            .toList();
    }

    Product updated = productService.updateImages(id, mainImageUrl, galleryImageUrls);
    return productMapper.toDTO(updated);
}

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						74
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        @PostMapping(value = "{id}/color-images", consumes =
MediaType.MULTIPART_FORM_DATA_VALUE)
        public ProductDto uploadColorImages(
            @PathVariable Long id,
            @RequestParam("color") String color,
            @RequestParam(value = "images", required = false) List<MultipartFile> images
        ) {
            Product product = productService.findById(id);
            if (product == null) {
                throw new RuntimeException("Product not found: " + id);
            }

            List<MultipartFile> nonEmptyImages = images == null
                ? List.of()
                : images.stream()
                    .filter(file -> file != null && !file.isEmpty())
                    .limit(4)
                    .toList();

            if (color == null || color.isBlank()) {
                throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "Color is
required");
            }
            if (nonEmptyImages.isEmpty()) {
                throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "No color
images were provided");
            }

            List<String> imageUrls = nonEmptyImages.stream()
                .map(file -> fileStorageService.saveProductImage(id, file))
                .filter(url -> url != null)
                .toList();

            Product updated = productService.updateColorImages(id, color, imageUrls);

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						75
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        return productMapper.toDTO(updated);
    }
}

```

Додаток В. Файл ProductService.java

```
package com.sportshop.service;
```

```

import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.sportshop.entity.Category;
import com.sportshop.entity.Product;
import com.sportshop.repository.CategoryRepository;
import com.sportshop.repository.ProductRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

```

```

import java.util.ArrayList;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;

```

```
@Service
```

```
@RequiredArgsConstructor
```

```
public class ProductService {
```

```
    private final ProductRepository productRepository;
```

```
    private final CategoryRepository categoryRepository;
```

```
    private final ObjectMapper objectMapper = new ObjectMapper();
```

```
    private final TypeReference<Map<String, List<String>>> colorImagesType = new
TypeReference<>() {};
```

```
    public List<Product> findAll() {
```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						76
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        return productRepository.findAll();
    }

    public Product findById(Long id) {
        return productRepository.findById(id).orElse(null);
    }

    public Product save(Product product) {
        return productRepository.save(product);
    }

    public Product update(Long id, Product updatedProduct) {
        Product existingProduct = findById(id);
        if (existingProduct == null) {
            return null;
        }

        existingProduct.setName(updatedProduct.getName());
        existingProduct.setPrice(updatedProduct.getPrice());
        existingProduct.setSize(updatedProduct.getSize());
        existingProduct.setSizes(copyList(updatedProduct.getSizes()));
        existingProduct.setColors(copyList(updatedProduct.getColors()));
        existingProduct.setBrand(updatedProduct.getBrand());
        existingProduct.setCategory(updatedProduct.getCategory());
        if (updatedProduct.getColorImages() != null &&
            !"{}".equals(updatedProduct.getColorImages())) {
            existingProduct.setColorImages(updatedProduct.getColorImages());
        }

        return productRepository.save(existingProduct);
    }

    @Transactional
    public Product updateImages(Long id, String mainImage, List<String> galleryImages) {

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						77
Зм.	Арк.	№ докум.	Підпис	Дата		

```

Product existingProduct = findById(id);
if (existingProduct == null) {
    return null;
}

if (mainImage != null) {
    existingProduct.setMainImage(mainImage);
}

if (galleryImages != null) {
    if (existingProduct.getGalleryImages() == null) {
        existingProduct.setGalleryImages(new ArrayList<>());
    }
    existingProduct.getGalleryImages().clear();
    existingProduct.getGalleryImages().addAll(galleryImages);
}

return productRepository.saveAndFlush(existingProduct);
}

@Transactional
public Product updateColorImages(Long id, String color, List<String> imageUrls) {
    Product existingProduct = findById(id);
    if (existingProduct == null) {
        return null;
    }
    if (color == null || color.isBlank()) {
        throw new IllegalArgumentException("Color is required");
    }

    Map<String, List<String>> colorImages =
readColorImages(existingProduct.getColorImages());
    colorImages.put(color.trim(), copyList(imageUrls));
    existingProduct.setColorImages(writeColorImages(colorImages));
}

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

```

        return productRepository.saveAndFlush(existingProduct);
    }

    public void deleteById(Long id) {
        productRepository.deleteById(id);
    }

    public Category findCategory(Long id) {
        return categoryRepository.findById(id).orElse(null);
    }

    public List<Product> filterProducts(Long categoryId, String brand, Double minPrice,
Double maxPrice) {

        List<Product> products = productRepository.findAll();

        if (categoryId != null) {
            products = products.stream()
                                .filter(p -> p.getCategory() != null &&
p.getCategory().getId().equals(categoryId))
                                .toList();
        }

        if (brand != null) {
            products = products.stream()
                                .filter(p -> p.getBrand().equalsIgnoreCase(brand))
                                .toList();
        }

        if (minPrice != null) {
            products = products.stream()
                                .filter(p -> p.getPrice() >= minPrice)
                                .toList();
        }
    }

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

```

        if (maxPrice != null) {
            products = products.stream()
                .filter(p -> p.getPrice() <= maxPrice)
                .toList();
        }

        return products;
    }

    private List<String> copyList(List<String> values) {
        return values == null ? new ArrayList<>() : new ArrayList<>(values);
    }

    private Map<String, List<String>> readColorImages(String value) {
        if (value == null || value.isBlank()) {
            return new LinkedHashMap<>();
        }

        try {
            return objectMapper.readValue(value, colorImagesType);
        } catch (Exception ex) {
            return new LinkedHashMap<>();
        }
    }

    private String writeColorImages(Map<String, List<String>> value) {
        if (value == null || value.isEmpty()) {
            return "{}";
        }

        try {
            return objectMapper.writeValueAsString(value);
        } catch (Exception ex) {

```

```

        throw new RuntimeException("Could not serialize product color images", ex);
    }
}

```

Додаток Г. Файл AdminPanel.jsx

```

export default function AdminPanel() {
    const stats = useAsyncData(() => api.getAdminStats(), []);
    const products = useAsyncData(() => api.getProducts(), []);
    const categories = useAsyncData(() => api.getCategories(), []);
    const orders = useAsyncData(() => api.getOrders(), []);

    const [productForm, setProductForm] = useState(initialProduct);
    const [message, setMessage] = useState("");
    const [busy, setBusy] = useState(false);
    const [editingProductId, setEditingProductId] = useState(null);

    function updateProduct(key, value) {
        setProductForm((current) => ({
            ...current,
            [key]: value
        }));
    }

    async function createProduct(event) {
        event.preventDefault();

        setBusy(true);
        setMessage("");

        try {
            const payload = {
                name: productForm.name,
                brand: productForm.brand,

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

```

        price: Number(productForm.price),
        categoryId: Number(productForm.categoryId),
        size: productForm.selectedSizes[0],
        sizes: productForm.selectedSizes,
        colors: productForm.selectedColors
    };

    if (editingProductId) {
        await api.updateProduct(editingProductId, payload);
        setMessage('Товар успішно оновлено.');
```

```

    } else {
        await api.createProduct(payload);
        setMessage('Товар успішно створено.');
```

```

    }

    await products.reload();

} catch (err) {
    setMessage('Помилка збереження товару.');
```

```

} finally {
    setBusy(false);
}
}

function startEdit(product) {
    setEditingProductId(product.id);

    setProductForm({
        name: product.name,
        brand: product.brand,
        price: product.price,
        categoryId: product.categoryId,
        selectedSizes: product.sizes,
        selectedColors: product.colors
    });
}

```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

```

    });
}

async function deleteProduct(id) {
    setBusy(true);

    try {
        await api.deleteProduct(id);
        await products.reload();
        setMessage('Товар видалено.');
```

```

    } catch (err) {
        setMessage('Не вдалося видалити товар.');
```

```

    } finally {
        setBusy(false);
    }
}

async function createCategory(event) {
    event.preventDefault();

    try {
        await api.createCategory({
            name: categoryName
        });

        setMessage('Категорію створено.');
```

```

        categories.reload();
    } catch (err) {
        setMessage('Помилка створення категорії.');
```

```

    }
}

async function updateCategory() {
    try {
```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

```

    await api.updateCategory(editingCategoryId, {
      name: categoryEditName
    });

    setMessage('Категорію оновлено.');
    categories.reload();
  } catch (err) {
    setMessage('Не вдалося оновити категорію.');
  }
}

async function deleteCategory(id) {
  try {
    await api.deleteCategory(id);

    setMessage('Категорію видалено.');
    categories.reload();
  } catch (err) {
    setMessage('Помилка видалення категорії.');
  }
}

async function changeOrderStatus(orderId, status) {
  try {
    await api.updateOrderStatus(orderId, status);

    setMessage(
      `Статус замовлення №${orderId} змінено`
    );

    orders.reload();
  } catch (err) {
    setMessage(
      'Не вдалося змінити статус замовлення.'
    );
  }
}

```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

```

    );
  }
}

return (
  <section className="admin-page">

    <h1>Керування SportStore</h1>

    <form onSubmit={createProduct}>
      <input
        value={productForm.name}
        onChange={(e) =>
          updateProduct('name', e.target.value)
        }
      />

      <input
        value={productForm.brand}
        onChange={(e) =>
          updateProduct('brand', e.target.value)
        }
      />

      <input
        type="number"
        value={productForm.price}
        onChange={(e) =>
          updateProduct('price', e.target.value)
        }
      />

      <button type="submit">
        {editingProductId}

```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						85
Зм.	Арк.	№ докум.	Підпис	Дата		

```

      ? 'Зберегти зміни'
      : 'Додати товар'}
    </button>
  </form>

  <section>
    {products.data.map((product) => (
      <div key={product.id}>
        <strong>{product.name}</strong>

        <button
          onClick={() => startEdit(product)}
        >
          Редагувати
        </button>

        <button
          onClick={() =>
            deleteProduct(product.id)
          }
        >
          Видалити
        </button>
      </div>
    ))}
  </section>

  <section>
    {orders.data.map((order) => (
      <div key={order.id}>

        <strong>
          Замовлення #{order.id}
        </strong>

```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

```

        <select
          value={order.status}
          onChange={(event) =>
            changeOrderStatus(
              order.id,
              event.target.value
            )
          }
        >
          {ORDER_STATUSES.map((status) => (
            <option
              key={status}
              value={status}
            >
              {status}
            </option>
          ))}
        </select>

      </div>
    )}
  </section>

</section>
);
}

```

Додаток Д. Файл ProductDetails.jsx

```

export default function ProductDetails() {
  const { id } = useParams();
  const location = useLocation();
  const { addToCart } = useCart();

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						87
Зм.	Арк.	№ докум.	Підпис	Дата		

```

const [added, setAdded] = useState(false);
const [selectedSize, setSelectedSize] = useState("");
const [selectedColor, setSelectedColor] = useState("");
const [selectionError, setSelectionError] = useState("");
const [selectedImage, setSelectedImage] = useState("");

const { data: product, loading, error } =
  useAsyncData(() => api.getProduct(id), [id]);

const categories =
  useAsyncData(() => api.getCategories(), []);

if (loading) {
  return <StateBlock title="Завантаження товару..." />;
}

if (error) {
  return <StateBlock title="Товар недоступний" text={error} />;
}

if (!product?.id) {
  return <StateBlock title="Товар не знайдено" />;
}

const image = getProductImage(product);
const gallery = getProductGallery(product, selectedColor);
const activeImage =
  gallery.includes(selectedImage)
    ? selectedImage
    : gallery[0] || image;

const sizes = normalizeOptions(
  product.sizes,

```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						88
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    [product.size || 'універсальний']
  );

  const colors = normalizeOptions(
    product.colors,
    ['Чорний', 'Білий', 'Сірий']
  );

  const category = categories.data.find(
    (item) => item.id === product.categoryId
  );

  const categoryName =
    category?.name || `Категорія #${product.categoryId}`;

  const backTo =
    location.state?.from ||
    `/catalog${product.categoryId}
    ? `?categoryId=${product.categoryId}`
    : "}`;

  function handleAdd() {
    if (!selectedSize) {
      setSelectionError('Оберіть розмір');
      return;
    }

    if (!selectedColor) {
      setSelectionError('Оберіть колір');
      return;
    }

    addToCart({
      id: product.id,

```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						89
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        productId: product.id,
        name: product.name,
        brand: product.brand || 'SportStore',
        price: product.price,
        image: activeImage,
        selectedSize,
        selectedColor,
        quantity: 1
    });

    setAdded(true);
    setSelectionError("");
}

return (
    <section className="container product-details-page">
        <Link className="back-link" to={backTo}>
            Назад до каталогу
        </Link>

        <div className="product-details-layout">
            <div className="product-gallery">
                <div className="product-main-image">
                    <img src={activeImage} alt={product.name} />
                </div>

                {gallery.length > 1 && (
                    <div className="product-thumbnails">
                        {gallery.map((galleryImage) => (
                            <button
                                className={
                                    activeImage === galleryImage
                                    ? 'selected'
                                    : "

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						90
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    }
    key={galleryImage}
    type="button"
    onClick={() =>
      setSelectedImage(galleryImage)
    }
  >

  <img
    src={galleryImage}
    alt={` ${product.name} фото`}
  />

</button>

  )})
</div>

})
</div>

```

```

<div className="product-info">
  <p className="eyebrow">
    {product.brand || 'SportStore'}
  </p>

```

```

  <h1>{product.name}</h1>

```

```

  <p className="product-price">
    {Number(product.price).toFixed(2)} грн
  </p>

```

```

  <p className="product-description">
    {product.name} від {product.brand || 'SportStore'}
    створено для тренувань і повсякденного
    використання. Категорія: {categoryName}.
  </p>

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						91
Зм.	Арк.	№ докум.	Підпис	Дата		

```

<div className="spec-list">
  <span>Бренд</span>
  <strong>{product.brand || 'SportStore'}</strong>

  <span>Категорія</span>
  <strong>{categoryName}</strong>

  <span>Розмір</span>
  <strong>{sizes.join(', ')}</strong>

  <span>Кольори</span>
  <strong>{colors.join(', ')}</strong>

  <span>Код товару</span>
  <strong>{product.id}</strong>
</div>

```

```

<div className="product-choice-group">

```

```

  <span>Оберіть розмір</span>

```

```

<div className="choice-chip-grid">

```

```

  {sizes.map((size) => (
    <button
      className={`choice-chip ${
        selectedSize === size ? 'selected' : ''
      }`}
      key={size}
      type="button"
      onClick={() => {
        setSelectedSize(size);
        setAdded(false);
        setSelectionError("");
      }}
    >

```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						92
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        {size}
      </button>
    )})
  </div>
</div>

<div className="product-choice-group">
  <span>Оберіть колір</span>

  <div className="choice-chip-grid">
    {colors.map((color) => (
      <button
        className={`choice-chip ${
          selectedColor === color ? 'selected' : ''
        }`}
        key={color}
        type="button"
        onClick={() => {
          setSelectedColor(color);
          setSelectedImage("");
          setAdded(false);
          setSelectionError("");
        }}
      >
        {color}
      </button>
    )})
  </div>
</div>

{selectionError && (
  <p className="form-error">{selectionError}</p>
)}

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

```

    <button
      className="primary-button full"
      type="button"
      onClick={handleAdd}
    >
      {added
        ? 'Додано до кошика'
        : 'Додати в кошик'}
    </button>
  </div>
</div>
</section>
);
}

```

Додаток Е. Файл api.js

```

const API_URL =
  import.meta.env.VITE_API_URL || 'http://localhost:8080';

function getToken() {
  return localStorage.getItem('sportstore_token');
}

async function request(path, options = {}) {
  const headers = {
    'Content-Type': 'application/json',
    ...(options.headers || {})
  };

  const token = getToken();

```

					2026.КВР.122.423.19.00.00 ПЗ	Арк.
						94
Зм.	Арк.	№ докум.	Підпис	Дата		

```

if (token) {
    headers.Authorization = `Bearer ${token}`;
}

const response = await fetch(`${API_URL}${path}`, {
    ...options,
    headers
});

if (!response.ok) {
    const message = await response.text();
    throw new Error(message || `HTTP ${response.status}`);
}

if (response.status === 204) {
    return null;
}

const contentType = response.headers.get('content-type');

return contentType?.includes('application/json')
    ? response.json()
    : response.text();
}

export const api = {
    getProducts(params = {}) {
        const search = new URLSearchParams();
    }
}

```

```

Object.entries(params).forEach(([key, value]) => {
  if (value !== undefined && value !== null && value !== "") {
    search.set(key, value);
  }
});

const query = search.toString();

return request(`/products${query ? `?` + query : ""}`);
},

getProduct(id) {
  return request(`/products/${id}`);
},

getCategories() {
  return request('/categories');
},

login(payload) {
  return request('/auth/login', {
    method: 'POST',
    body: JSON.stringify(payload)
  });
},

register(payload) {
  return request('/auth/register', {
    method: 'POST',

```

```

        body: JSON.stringify(payload)
    });
},

createOrder(payload) {
    return request('/orders', {
        method: 'POST',
        body: JSON.stringify(payload)
    });
},

getOrders() {
    return request('/orders');
},

getMyOrders() {
    return request('/orders/my');
},

updateOrderStatus(id, status) {
    return request(`/orders/${id}/status`, {
        method: 'PUT',
        body: JSON.stringify({ status })
    });
},

createProduct(payload) {
    return request('/products', {
        method: 'POST',

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
						97
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        body: JSON.stringify(payload)
    });
},

```

```

updateProduct(id, payload) {
    return request(`/products/${id}`, {
        method: 'PUT',
        body: JSON.stringify(payload)
    });
},

```

```

deleteProduct(id) {
    return request(`/products/${id}`, {
        method: 'DELETE'
    });
},

```

```

createCategory(payload) {
    return request('/categories', {
        method: 'POST',
        body: JSON.stringify(payload)
    });
},

```

```

updateCategory(id, payload) {
    return request(`/categories/${id}`, {
        method: 'PUT',
        body: JSON.stringify(payload)
    });
},

```

```

    },

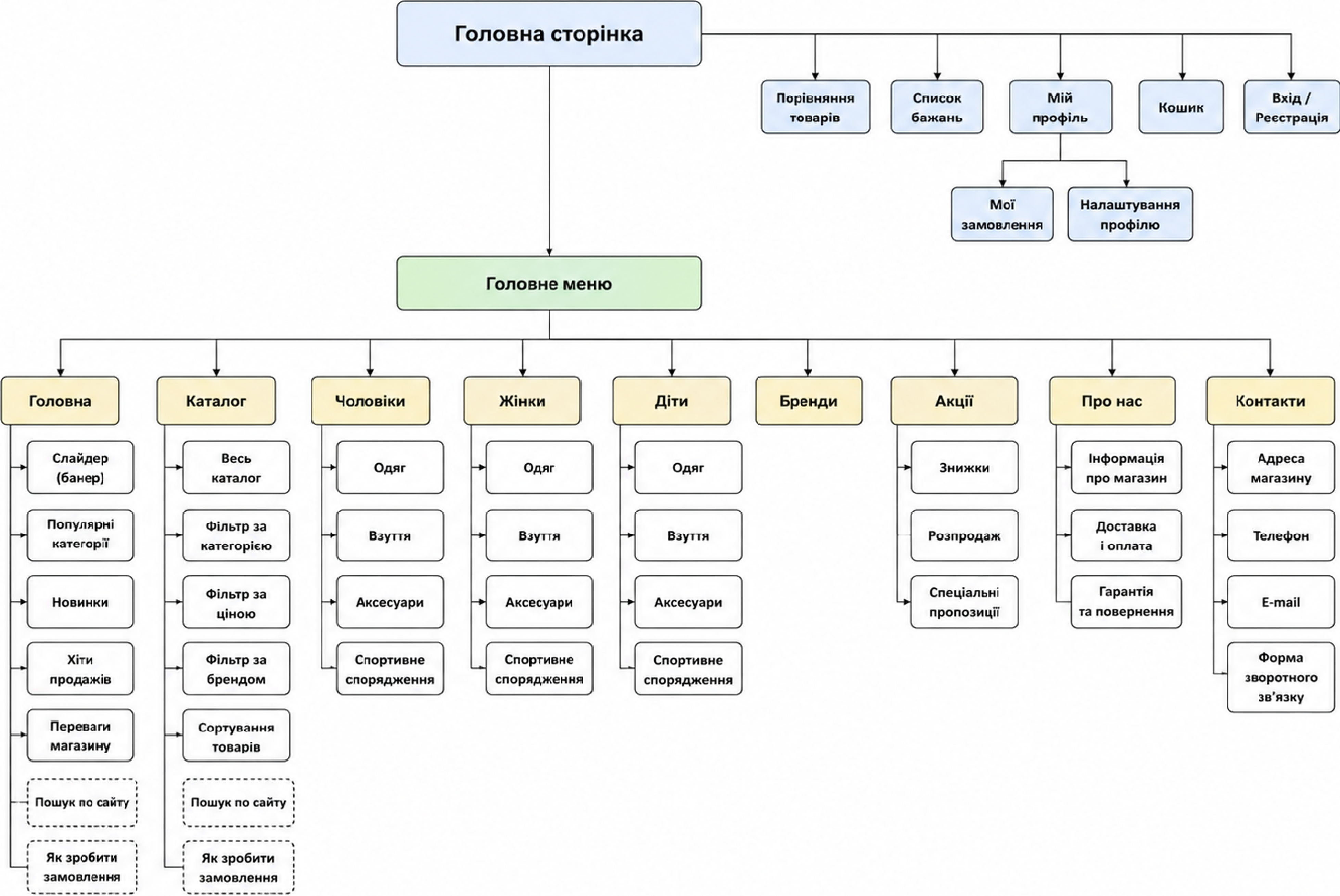
    deleteCategory(id) {
      return request(`/categories/${id}`, {
        method: 'DELETE'
      });
    }
  };

export { API_URL };

```

					2026.KBP.122.423.19.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		99

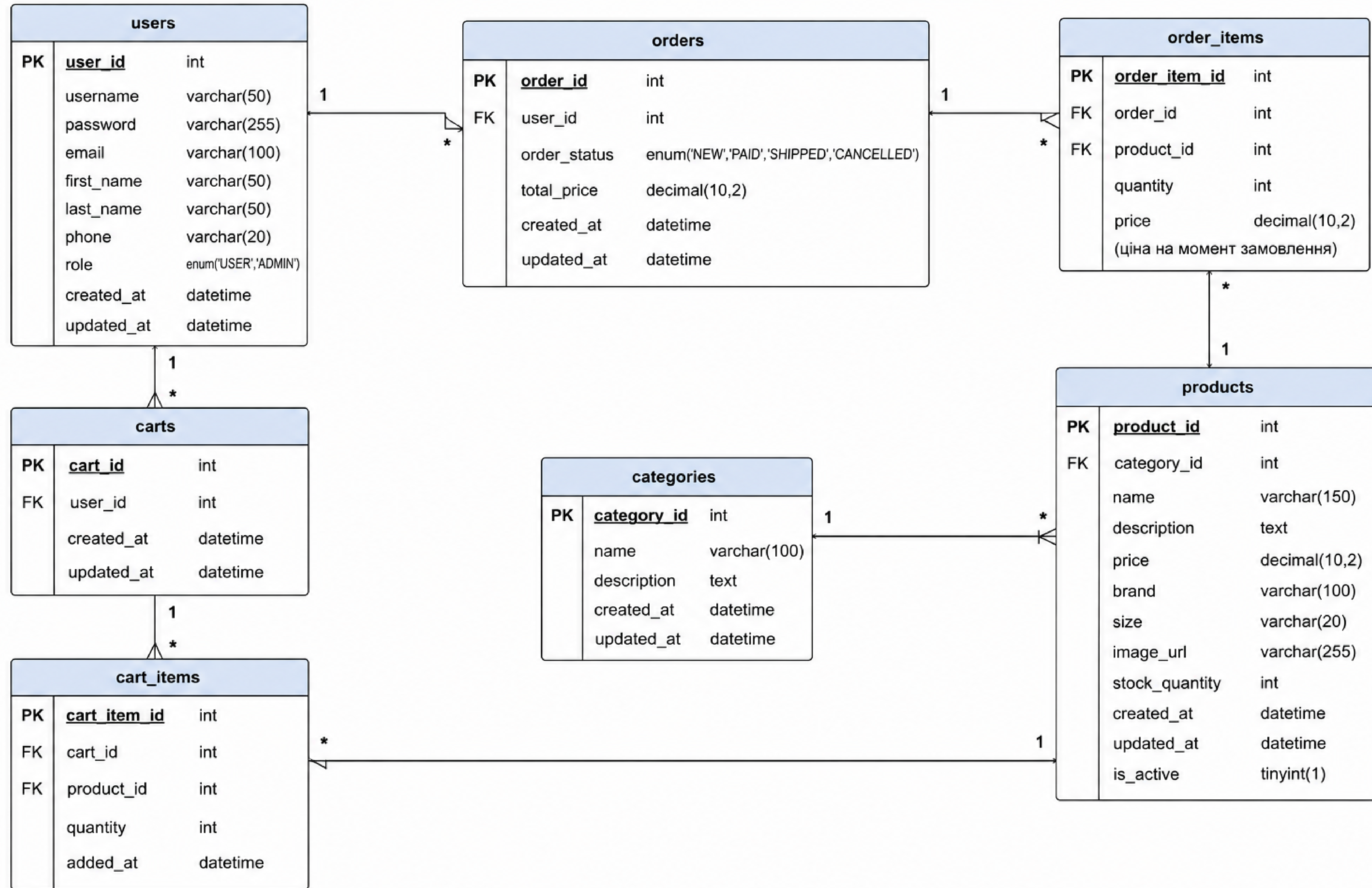
Структурна Схема



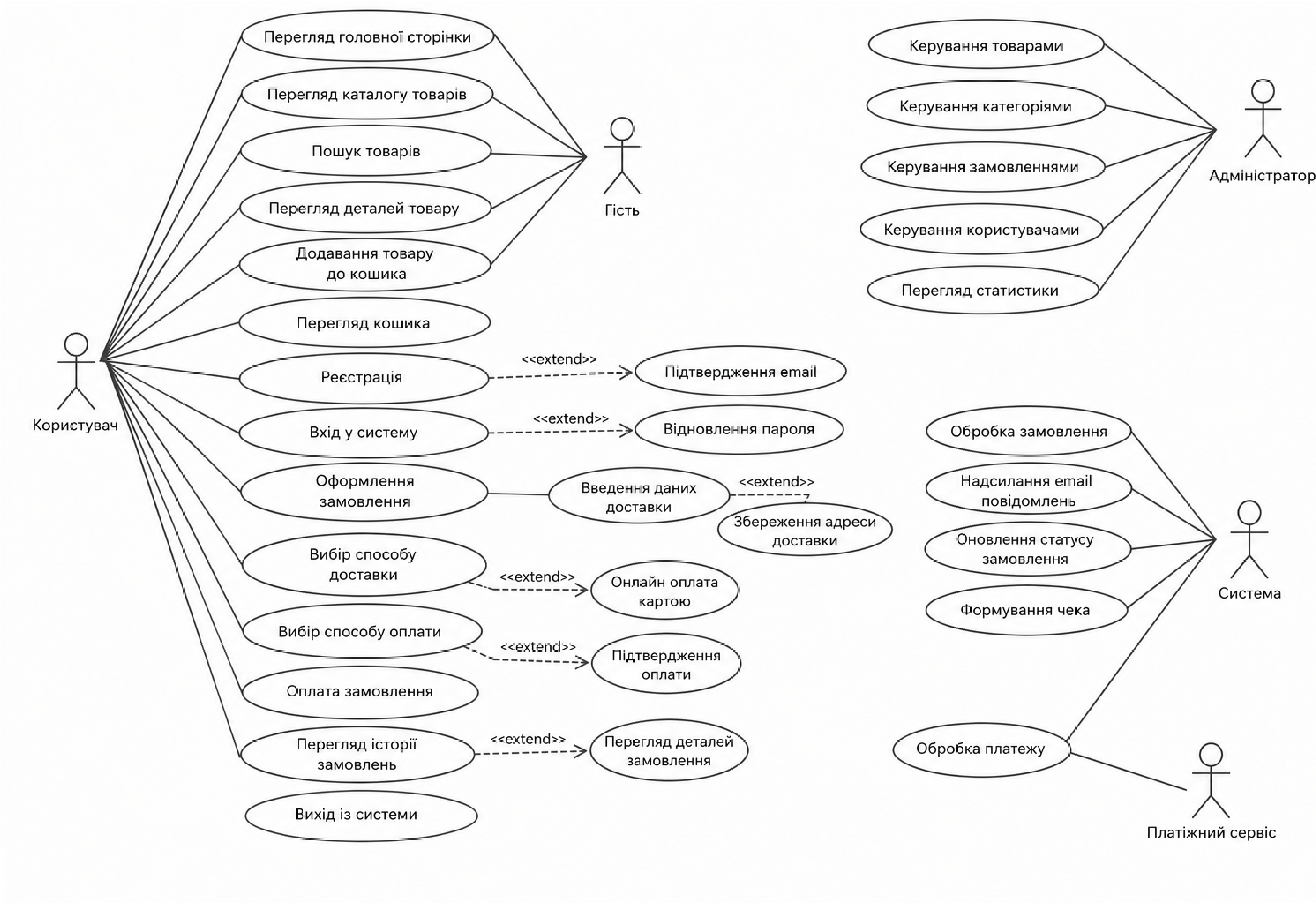
Умовні позначення:

- Верхня панель (службові розділи)
- Головне меню
- Основні розділи
- Підрозділи
- Додаткові елементи

ER-діаграма бази даних



UML-діаграма діяльності користувача



					2026.KBP.122.423.19.00.00 ТБ				
Изм./Ист.	№ докум.	Подп.	Дата	Разработка вебсайту магазину спортивного одягу «SportStore» Таблиця техніко-економічних показників			Лит.	Масса	Масштаб
Разработ.	Савчук АВ								
Проб.	Цимбалюк ЛВ								
Т.контр.									
Рецензент									
Н.контр.					Лист	Листов	1		
Утв.	Пиливчук В.А.			ВСР ТКХ ТНТУ КН-423 м. Тернопіль					