МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»

Циклова комісія математичних дисциплін та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ КЕРУВАННЯ АСОРТИМЕНТОМ МАГАЗИНУ ОДЯГУ»

Виконала: здобувачка освіти 4 курсу, групи К121-22
Спеціальність 121 Інженерія програмного забезпечення
(шифр і спеціальність)
Катерина ЗАЯЦ
(ім'я та ПРІЗВИЩЕ)
Керівник Альона ХОДАКОВСЬКА
(ім'я та ПРІЗВИЩЕ)
Рецензент професор кафедри програмної інженерії ЗНУ, к. ф.-м. н, доцент Олексій КУДІН
(посада, вчене звання, науковий ступінь, ім'я та ПРІЗВИЩЕ)

Запоріжжя
2026

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ВСП «Економіко-правничий фаховий коледж ЗНУ»

Освітньо–кваліфікаційний ступінь фаховий молодший бакалавр

Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Голова циклової комісії математичних дисциплін та інформаційних технологій

_____ Т.М. Смолянкова
(підпис)

“ 05 ” червня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ЗАЯЦ Катерині Сергіївгні
(ПРІЗВИЩЕ, ім’я та по– батькові)

1. Тема роботи «Розробка вебзастосунку для керування асортиментом магазину одягу»

Керівник роботи PhD ХОДАКОВСЬКА Альона Валеріївна
(науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

Затверджені наказом ВСП ЕПФК ЗНУ від «01» грудня 2025 р. № 1728-с

2. Строк подання студентом роботи 05.06.2026

3. Вихідні дані до роботи 1. Постановка задачі
2. Перелік літератури

4. Зміст розрахунково– пояснювальної записки (перелік питань, які потрібно розробити)
1.Сучасні інформаційні системи
2. Розробка проєкту інформаційної системи
3. Програмна реалізація інформаційної системи

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація до захисту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.12.2025

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи:		
	постановка задачі	Грудень 2025	виконано
2.	Збір вихідних даних, обробка методичних		
	та теоретичних джерел	Січень 2026	виконано
3.	Розробка першого розділу:		
	огляд інформаційних систем та технологій для їхнього створення	Лютий 2026	виконано
4.	Розробка другого розділу:		
	розробка проєкту інформаційної системи та вибір технологій	Квітень 2026	виконано
5.	Розробка третього розділу:		
	розробка застосунку	Травень 2026	виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи та перевірка на плагіат	Червень 2026	виконано
7.	Захист кваліфікаційної роботи	16.06.2026	виконано

Здобувач освіти _____ Катерина ЗАЯЦ
(підпис) (ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Альона ХОДАКОВСЬКА
(підпис) (ім'я ПРІЗВИЩЕ)

Нормоконтроль пройдено

Нормоконтролер _____ Юлія БОРИСОВСЬКА
(підпис) (ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Кваліфікаційна робота: 51 сторінок, 25 рисунків, 2 таблиці, 14 джерел.

Об'єкт дослідження: процес автоматизації управління товарним асортиментом у роздрібному магазині одягу.

Предмет дослідження: вебзастосунок для керування каталогом товарів, обліку складських залишків, фіксації продажів і розмежування прав доступу персоналу.

Мета роботи: проєктування та розробка вебзастосунку для керування асортиментом магазину одягу, що забезпечує повний цикл роботи з товарами, облік продажів і розмежування прав доступу між різними категоріями користувачів.

Методи дослідження: аналітичний, порівняльний аналіз, метод системного проєктування, метод практичної реалізації та тестування.

У кваліфікаційній роботі проведено аналіз предметної області роздрібної торгівлі одягом та розглянуто сучасний інструментарій веброзробки. На основі порівняльного аналізу існуючих програмних аналогів визначено функціональні вимоги до системи та обґрунтовано вибір технологічного стеку. З використанням мови програмування PHP, системи керування базами даних MySQL та клієнтських технологій HTML, CSS і JavaScript розроблено повноцінний вебзастосунок для магазину одягу.

У межах роботи спроєктовано реляційну базу даних із семи взаємопов'язаних таблиць, побудовано UML-діаграми та реалізовано трирівневу рольову модель доступу для адміністратора, менеджера і касира. Застосунок включає публічну частину для покупців і адміністративну панель для персоналу з модулем аналітики продажів. Забезпечено захист від SQL-ін'єкцій і XSS-атак.

Розроблений застосунок може бути впроваджений у роботу магазину одягу малого або середнього масштабу як готовий інструмент автоматизації

операційної діяльності. Модульна архітектура продукту дозволяє розширювати його функціональність відповідно до потреб конкретного підприємства.

ВЕБЗАСТОСУНОК, PHP, MYSQL, HTML, CSS, JAVASCRIPT, CHART.JS, РОЛЬОВА МОДЕЛЬ ДОСТУПУ, АСОРТИМЕНТ, МАГАЗИН ОДЯГУ

SUMMARY

Thesis: 51 pages, 25 figures, 2 tables, 14 references.

Research object: the process of automating product assortment management in a retail clothing store.

Research subject: a web application for managing the product catalog, tracking inventory, recording sales, and assigning staff access rights.

Objective of the thesis: to design and develop a web application for managing a clothing store's product assortment, ensuring a full cycle of product management, sales tracking, and the allocation of access rights among different user categories.

Research methods: analytical, comparative analysis, systems design method, practical implementation and testing.

This thesis analyzes the subject area of clothing retail and examines modern web development tools. Based on a comparative analysis of existing software counterparts, the functional requirements for the system were defined and the choice of the technology stack was justified. Using the PHP programming language, the MySQL database management system, and client-side technologies such as HTML, CSS, and JavaScript, a full-fledged web application for a clothing store was developed.

As part of the work, a relational database consisting of seven interconnected tables was designed, UML diagrams were created, and a three-level role-based access model was implemented for the administrator, manager, and cashier. The application includes a public section for customers and an administrative panel for staff with a sales analytics module. Protection against SQL injections is provided.

As part of this project, a relational database consisting of seven interconnected tables was designed, UML diagrams were created, and a three-tier role-based access model was implemented for the administrator, manager, and cashier. The application includes a public section for customers and an administrative panel for staff,

featuring a sales analytics module. Protection against SQL injection and XSS attacks is ensured.

The developed application can be implemented in a small or medium-sized clothing store as a ready-to-use tool for automating operational activities. The product’s modular architecture allows its functionality to be expanded according to the needs of a specific business.

WEB APPLICATION, PHP, MYSQL, HTML, CSS, JAVASCRIPT, CHART.JS, ROLE-BASED ACCESS MODEL, PRODUCT RANGE, CLOTHING STORE

ЗМІСТ

Завдання на кваліфікаційну роботу студенту 2
Реферат 4
Summary 6
Вступ 9
1 Аналіз предметної області та сучасних вебтехнологій 12
1.1 Веброзробка та її складові 12
1.2 Огляд сучасного інструментарію для розробки вебзастосунків 15
1.3 Характеристика предметної області та постановка задачі 17
1.4 Порівняльний аналіз існуючих програмних рішень 18
2 Розробка проєкту вебсайту 22
2.1 Технічне завдання 22
2.2 Діаграма прецедентів 24
2.3 Діаграма класів 26
2.4 Діаграма послідовності 27
2.5 Діаграма розгортання 29
2.6 Проєктування бази даних 29
3 Програмна реалізація 32
3.1 Архітектура та технології реалізації 32
3.2 Реалізація функціональних вимог 34
3.3 Тестування вебсайту 45
Висновки 48
Перелік використаних джерел 50

ВСТУП

На сьогодні вебсайти є важливою частиною сучасного бізнесу, і вони надають компаніям широкий спектр можливостей, від автоматизації рутинних операцій і ведення обліку до аналізу продажів і керування персоналом. У результаті розробка спеціалізованих вебзастосунків стає все більш затребуваною сферою бізнесу. Створення зручних, ефективних і функціональних програмних рішень, які відповідають сучасним стандартам якості та задовольняють потреби конкретного бізнесу, є однією з головних завдань, з якими стикаються розробники.

Ця кваліфікаційна робота присвячена створенню вебдодатка для управління асортиментом магазину одягу «Aura». Мета цього застосунку полягає в тому, щоб надати зручний і ефективний інструмент, щоб автоматизувати операції, пов’язані з роботою в магазині, такі як ведення товарного каталогу, облік складських залишків, фіксація продажів і розмежування прав доступу між різними посадовими особами, такими як адміністратори, менеджери та касирі.

Тема є актуальною, оскільки більшість магазинів одягу малого та середнього розміру часто все ще використовують електронні таблиці для ведення обліку товарів. Це призводить до дублювання даних, помилок в обліку залишків і відсутності аналітики. Великі підприємства не можуть фінансово і технічно використовувати ERP-системи, розроблені спеціально для них. Існує значна потреба в спеціалізованому, доступному та функціональному вебрішенні.

Мета кваліфікаційної роботи – спроєктувати та створити вебзастосунок для керування асортиментом магазину одягу, який забезпечить облік продажів, розподіл прав доступу між різними користувачами та забезпечить повний цикл роботи з товарами.

Для досягнення мети були поставлені наступні завдання: аналіз предметної області роздрібної торгівлі одягом; вивчення сучасних інструментів веброзробки; розробка проєкту вебзастосунку, включаючи розробку технічного завдання, розробку UML-діаграми та проєктування структури реляційної бази даних; і реалізація клієнтської та серверної частини застосунку, а також проведення функціонального тестування.

Об'єктом кваліфікаційної роботи є процес автоматизації управління товарним асортиментом у роздрібному магазині одягу.

Предметом роботи є вебдодаток для каталогізації товарів, обліку складських запасів, фіксації продажів і контролю доступу персоналу.

Для кваліфікаційної роботи використовуються такі методи: аналітичний – для вивчення предметної області та наявних програмних рішень; порівняльний аналіз – для вибору технологічного стеку та оцінки конкурентних продуктів; системне проєктування – для розробки архітектури та структури бази даних; практична реалізація та тестування – для програмної розробки та перевірки функціональності застосунку.

Новизна роботи полягає в тому, щоб створити вебзастосунок, який поєднує публічну клієнтську частину та адміністративну панель у одному рішенні без використання зовнішніх фреймворків. Застосунок також має трирівневу рольову модель доступу, а також комплексне захисту від SQL-ін’єкцій і XSS-атак. Розроблений застосунок має практичне значення, оскільки він може бути використаний у реальних магазинах одягу як готовий інструмент для щоденної роботи персоналу. Крім того, його структура дозволяє легко розширювати його функції, не переписуючи основний код.

Вступна частина роботи охоплює основні етапи та елементи розробки вебзастосунку на основі технологічного стеку PHP та MySQL.

У першому розділі розглядається предметна область роздрібної торгівлі одягом, а також дається огляд сучасних вебтехнологій. Розглядаються елементи веброзробки, характеристики предметної області та постановка задачі, сучасний інструментарій для створення вебзастосунків і порівняльний

аналіз існуючих програмних аналогів. Визначення функціональних вимог до системи та обґрунтування вибору технологічного стеку є важливими компонентами.

Другий розділ розкриває різні етапи проектування вебзастосунку, включаючи розробку технічного завдання, створення UML-діаграм прецедентів, класів, послідовності та розгортання, а також проектування реляційної бази даних. Визначення сутностей, їх характеристик і зв’язків між ними є частиною проектування бази даних з урахуванням принципів нормалізації. Діаграми створюють основу для реалізації архітектури та пояснюють логіку взаємодії між компонентами системи.

У третьому розділі представлено програмну реалізацію вебзастосунку. Описується процес розробки програмного забезпечення, який включає в себе впровадження клієнтських і серверних компонентів, роботу з базою даних, механізми захисту від SQL-ін’єкцій і XSS-атак, а також реалізацію рольової моделі доступу. Окремо розглядається тестування застосунку та потенційні майбутні вдосконалення.

У висновку підбиваються підсумки виконаної роботи. Розробка вебзастосунку для керування асортиментом магазину одягу є результатом цієї кваліфікаційної роботи. Цей вебсайт дозволяє співробітникам ефективно вести облік товарів і продажів, надаючи їм зручний і функціональний інструмент для щоденної роботи. Застосунок базується на сучасних вебтехнологіях, що гарантує його безпеку, стабільність і зручність використання. В роботі показано весь процес створення вебдодатку, починаючи від аналізу предметної області та проєктування та закінчуючи програмною реалізацією та тестуванням.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ ВЕБТЕХНОЛОГІЙ

1.1 Веброзробка та її складові

Розглянемо веброзробку та її основні компоненти з точки зору трирівневої архітектури: клієнтська частина, серверна логіка та рівень зберігання даних. Створення вебрішень включає весь обсяг завдань, пов'язаних із формуванням, підтримкою функціонування та покращення інтернет-сторінок та вебдодатків, від простих статичних сторінок до складних корпоративних систем із непростою логікою та сховищами даних. Ця галузь розвивається шаленими темпами: інструменти, фреймворки та методології, що були актуальними кілька років тому, поступово витісняються новими, більш продуктивними підходами.

Зараз веброзробку класифікують на три основні складові: клієнтську частину, що відповідає за відображення інтерфейсу користувача; серверну частину, яка виконує бізнес-логіки; рівень зберігання даних, відомий як база даних. Кожен із цих компонентів виконує особливі функції, коли вони взаємодіють, то створюють цілісну архітектуру, що в літературі називається «повний стек» [1].

Всі елементи, з якими користувач взаємодіє через веббраузер , входять до клієнтської частину вебдодатку. Зазвичай реалізація клієнтської частини ґрунтується на трьох основних технологіях: HTML, CSS та JavaScript. Вони відповідають за структуру, оформлення та інтерактивність вебсторінки.

Мовою розмітки HTML визначається структурний каркас сторінки. Вона забезпечує розташування заголовків, текстових блоків, форм введення, таблиць та інших елементів інтерфейсу.

Візуальне оформлення регулює CSS, включаючи шрифти, відступи, палітру кольорів, анімаційні ефекти та здатність адаптуватися під різні розміри екранів.

JavaScript надає динамічності сторінці, дозволяючи реагувати на дії користувача без перезавантаження сторінки та оновлювати дані у реальному часі та керувати логікою безпосередньо на машині користувача [2].

Серверна частина відповідає за обробку запитів, реалізацію бізнес-правил та взаємодію зі сховищем даних. На сервері виконуються операції, які клієнт не може побачити, такі як автентифікація, авторизація, валідація вхідних даних і формування відповіді. Одними з популярних серверних рішень є Node.js, який використовує JavaScript для бекенду, Python з його фреймворками Django та Flask, PHP, Java, а також Go та Rust, які стають все більш популярними завдяки їхній високій швидкодії. Node.js дозволяє використовувати одну мову програмування як для клієнта, так і для сервера, що значно зменшує труднощі для входу та прискорює процес розробки . Це робить його особливо примітним [3].

Передача даних між клієнтським та серверним середовищем відбувається за допомогою програмного інтерфейсу API (Application Programming Interface). REST (Representational State Transfer) є найбільш поширеним архітектурним підходом. У цьому підході кожен інформаційний ресурс визначається унікальною адресою (URL), а операції над ним виконуються за допомогою звичайних HTTP-методів: GET, POST, PUT, DELETE.

Системи керування базами даних (СУБД) використовуються для реалізації рівнів даних у звичайних вебпродуктах. Реляційні СУБД включають такі продукти, як MySQL, PostgreSQL, Microsoft SQL Server. Вони використовують мову SQL для запитів та зберігають дані у таблиці з чіткими зв'язками. Переліки товарів, замовлення, фінансові операції та облікові записи користувачів – це все приклади баз даних, які добре підходять для роботи зі структурованими даними. З іншого боку, рішення NoSQL, тобто MongoDB,

Redis, Cassandra, спрямовані на неструктуровані або слабко структуровані дані та забезпечують просте горизонтальне масштабування, яке важливо для систем з високим піковим навантаженням [4].

Важливим поняттям для галузі сучасної веброзробки є адаптивний дизайн. Цей підхід передбачає коректне відображення вебінтерфейсу на різних пристроях, від смартфонів до широкоформатних моніторів. Цього можна досягти завдяки використанню медіавикликів CSS, гнучких макетних систем (grid та flex) і готових CSS-фреймворків. Мобільні пристрої використовували більше 58% вебтрафіку у 2024 році, що свідчить про те, що адаптація давно перетворилася з «бажаного доповнення» на абсолютну необхідність. Ці дані були отримані Statista [5].

Веброзробку також класифікують відповідно за типом кінцевого продукту. Вебсайти здебільшого виконують інформаційну функцію. Їх основним завданням є демонстрація контенту без додаткової інтерактивності. Натомість вебзастосунки працюють за принципом десктопних програм, але реалізуються у веббраузері. Вони включають розвинені інтерфейсні рішення, здійснення транзакцій, повну роботу з даними та інструменти авторизації.

Прогресивні вебзастосунки, також відомі як PWA, поєднують переваги класичних вебтехнологій та нативних мобільних продуктів. Вони мають досвід роботи з мобільними додатками, підтримують push-сповіщення та можуть працювати в режимі офлайн. Повнофункціональний вебзастосунок є оптимальним варіантом для впровадження системи управління каталогом товарів магазину одягу. Такий метод забезпечує централізоване зберігання даних, що дозволяє інформації бути доступною з будь-якого пристрою, підключеного до мережі, а також можливість гнучко обмежувати права доступу відповідно до ролей користувачів [6].

Безпека вебдодатків є окремим і критично важливим аспект розробки. Організація OWASP (Open Web Application Security Project) щороку складає перелік найбільш поширених вразливостей: серед них SQL-ін'єкції, XSS (атаки через міжсайтовий скриптинг), CSRF (підміна запитів) та інші. Для

будь-якого комерційного застосунку, який опрацьовує дані про продукцію, фінансові операції та облікові записи клієнтів, захист від подібних загроз є не додатковою можливістю, а базовою вимогою на етапі архітектурного проєктування [7].

1.2 Огляд сучасного інструментарію для розробки вебзастосунків

Одне з ключових завдань на старті розробки є вибір технологічного стеку для будь-якого вебпроєкту. Сучасний ринок веброзробки пропонує широкий спектр інструментів, як від програмних засобів для написання коду і до хмарних сервісів, що забезпечують фінальне розгортання застосунків. Вибір певного технологічного стеку безпосередньо впливає на швидкість створення продукту, зручність його подальшої підтримки та ефективність функціонування в умовах реальної експлуатації.

У кваліфікаційній роботі для реалізації вебзастосунку магазину одягу обрано технологічний стек PHP + MySQL на серверній стороні та HTML/CSS/JavaScript на клієнтській.

Для розробки серверної частини динамічного вебресурсу було обрано мову програмування PHP. Її архітектурна специфіка полягає у виконанні коду на стороні сервера, де здійснюється обробка клієнтських запитів, безпосередня взаємодія з базою даних та динамічна генерація HTML-сторінок для браузера користувача. Доцільність використання PHP обґрунтована такими чинниками: наявністю детальної офіційної документації, розвиненою спільнотою розробників, що спрощує пошук типових технічних рішень, відносно низьким порогом входження, що дозволяє оптимізувати терміни розробки. Локальне розгортання та тестування програмного продукту реалізовано за допомогою кросплатформного середовища розробки XAMPP. Даний інструмент дозволив інтегрувати в єдиний локальний стек вебсервер Apache, СУБД MySQL та інтерпретатор мови PHP. Такий підхід забезпечив повний цикл проєктування

та автономного налагодження вебресурсу на локальній машині без залучення хмарної інфраструктури [8].

Обрано систему управління базами даних MySQL для збору та обробки даних. Вона використовує реляційну модель зберігання, де дані складаються з таблиць, пов’язаних між собою. MySQL добре працює у вебпроєктах середнього масштабу, оскільки вона стабільна, підтримує транзакції, індексує та підтримує складні SQL-запити. У розробленій базі даних є сім таблиць: Category, Product, Product_images, Customer, Sale, Sale_Details, System_User. Їхня взаємодія забезпечує весь процес роботи магазину, починаючи з каталогу товарів і закінчуючи зберіганням замовлень [9].

На клієнтській стороні використано стандартне тріо веброзробки: HTML відповідає за структуру сторінок, CSS за їхній зовнішній вигляд, JavaScript відповідає за інтерактивність. Зокрема, JavaScript дозволяє працювати з кошиком, відкривати модальні вікна, які містять деталі товару, і постійно оновляти вміст без перезавантаження сторінки. Для побудови графіків у розділі аналітики адмін-панелі підключено бібліотеку Chart.js, легку та зручну у використанні бібліотеку для побудови інтерактивних діаграм у браузері [10].

Особливу увагу заслуговує безпека застосунку. Prepared statements, або механізм підготовлених виразів, використовується для захисту від SQL-ін’єкцій у всіх місцях введення даних. Цей механізм передає SQL-запит і дані окремо на сервер. Функція htmlspecialchars() використовується під час видалення будь-яких користувацьких даних, щоб запобігти XSS-атакам, які спричиняють впровадження шкідливого JavaScript-коду через поля форм. Bcrypt-хеші, алгоритм, спеціально розроблений для захисту паролів, значно надійніші за застарілі MD5 чи SHA1. Ці хеші використовуються лише для зберігання паролів адміністраторів [11].

Перелічені технології в сукупності утворюють цілісний стек, що дозволяє повноцінно запустити вебзастосунок з адмін-панеллю та публічною частиною Їх вибір зумовлений не лише функціональною відповідністю завданню, а й доступністю середовища розробки та можливістю швидкого локального розгортання.

1.3 Характеристика предметної області та постановка задачі

Постійне оновлення асортименту, сезонні коливання попиту та необхідність швидко реагувати на зміни на ринку є важливими для торгівельної точки, що спеціалізується на реалізації одягу. У цьому випадку основними елементами, що потребують контролю, є складські площі, товарні запаси, обсяги реалізації та клієнтська база. Без використання автоматизованих інструментів управління обліком цих складових швидко перетворюється на хаотичний процес і схильний до значних помилок.

Особливість торгівлі текстильними виробами полягає в тому, що кожен товар може мати безліч варіацій, пов’язаних із розмірним рядом та кольоровою гамою, які повинні бути опубліковані окремо. Наприклад, модель сукні певного стилю може бути представлена в п’яти різних розмірах і трьох різних кольорах. Тому, щоб правильно показати залишки, кожна така комбінація вважається окремою одиницею для обліку запасів (SKU). Ця ситуація значно ускладнює ручне ведення документації, що робить автоматизацію особливо виправданою.

Магазини одягу виконують такі бізнес-процеси: створення та підтримка актуального товарного каталогу з детальним описом категорії, торгової марки, розміру, кольору, складу тканини та вартості; моніторинг запасів на складі; фіксація продажів; забезпечення різного доступу для різних посадових осіб, таких як адміністратори, менеджери та касира. Кожен із процесів, наведених вище, потребує надійного зберігання даних і зручного інтерфейсу для щоденної роботи працівників.

На практиці невеликі та середні магазини одягу часто покладаються на електронні таблиці. Це неминуче призводить до усталених проблем, таких як надмірне дублювання даних, затримки в оновленні запасів і відсутність якісної аналітики. Крім того, у такій організації роботи немає розмежування повноважень, що означає, що працівники можуть навмисно або ненавмисно змінювати важливі дані. Більшість цих проблем усувається за допомогою

переходу до спеціалізованого вебпродукту. Дані зберігаються в центрі, до них можна отримати з будь-якого пристрою в мережевій інфраструктурі, а логіка верифікації та обмежень інтегрована безпосередньо в програмний комплекс.

Об'єктом проєкту є автоматизація процесу управління асортиментом магазину вбрання за допомогою сучасних вебтехнологій. Предметом розробки є вебзастосунок, який може виконувати такі завдання, як каталогізація товарів, облік складських запасів, фіксація продажів і розподіл прав доступу між різними категоріями користувачів.

Основне завдання: розробити концепцію та створити вебзастосунок для управління товарним асортиментом магазину одягу. Цей вебзастосунок повинен забезпечувати повний цикл роботи з продукцією, починаючи від додавання нових позицій та закінчуючи генерацією звіту про реалізацію протягом визначеного періоду часу.

Для реалізації поставленої цілі необхідно виконати наступні кроки:

- провести аналіз предметної сфери та ідентифікувати ключові системні сутності;
- вивчити та порівняти програмні аналоги, що існують на ринку;
- визначити набір технологій (стек) та надати обґрунтування для кожного обраного елемента;
- спроєктувати модель бази даних, враховуючи принципи нормалізації;
- розробити клієнтську та серверні складові частини програмного рішення;
- забезпечити розмежування повноважень між ролями адміністратора, менеджера та касира.

1.4 Порівняльний аналіз існуючих програмних рішень

У межах аналізу було розглянуто відомі інтернет-магазини одягу, які відповідають сучасним вимогам до електронної торгівлі та створюють високі

очікування клієнтів щодо зручності, оперативності та привабливого дизайну. При порівнянні їхніх підходів можна вибрати корисні ідеї, які варто врахувати при створенні власного вебсервісу.

Один із найбільш впізнаваних інтернет-магазинів одягу є Zara. Мінімалістичний дизайн інтерфейсу цього вебсайту зосереджується на об’ємних зображеннях і чіткому групуванні товарів у каталозі. Картки товарів містять детальні описи та високоякісні фотографії товару, а системи фільтрації працюють дуже швидко. Це дозволяє відвідувачам легко орієнтуватися в пропозиціях і миттєво знайти потрібну позицію. Тим не менш, велика кількість візуальних компонентів іноді зменшує загальну швидкість завантаження сторінок [12].

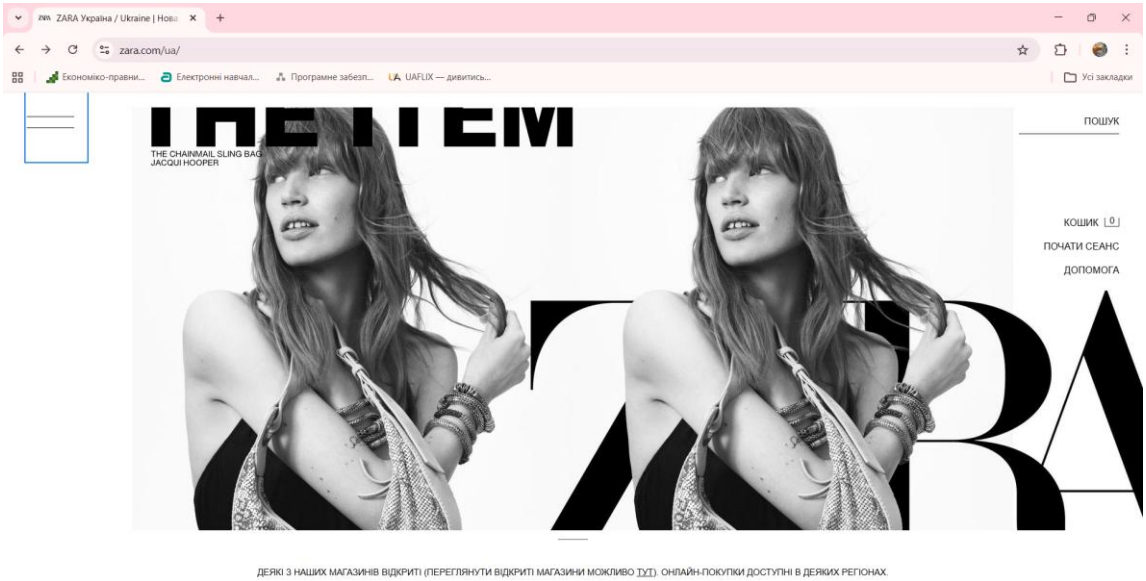


Рисунок 1.1 – Головна сторінка інтернет-магазину Zara

Серед платформ, що представлені на ринку України, показовим прикладом є INTERTOP. Цей сайт має продуману структуру каталогу та широкий вибір одягу та взуття. Стриманий дизайн, а функції сортування дозволяють швидко відфільтрувати позиції за виробником, розмірною сіткою, кольором чи матеріалом. Значною перевагою є наявність детальних технічних характеристик і великої кількості фотографій, що спрощує процес прийняття

вебпродукту. Додаткові переваги платформи включають бонусну програму, що сприяє зміцненню прихильності клієнтів. Ще одною перевагою є доступ до перегляду історії попередніх замовлень і зручний особистий простір [13].

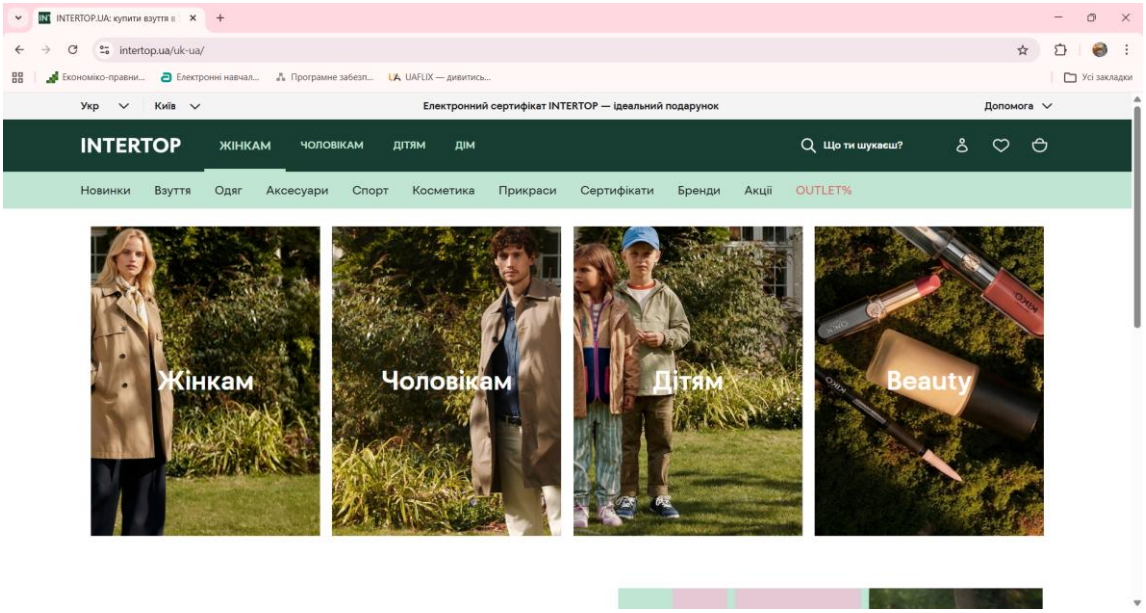


Рисунок 1.2 – Головна сторінка інтернет-магазину INTERTOP

Також чудовим прикладом є VOVK. Вебплатформа магазину пропонує користувачам широкий асортимент авторських виробів, поєднуючи стильний візуальний стиль із високою функціональністю. Інтерфейс сайту розроблений таким чином, щоб користувач міг легко орієнтуватися. Спеціальні інструменти сортування дозволяють миттєво сортувати продукти за категоріями, розмірами, кольоровою палітрою чи типом тканини. Наповнення карток товарів високоякісними фотографіями та детальними описами характеристик є важливою перевагою, оскільки це допомагає покупцям скласти чітке враження про товар ще до моменту покупки. Додатково, платформа пропонує зручне керування замовленнями через особистий кабінет. Також вебсайт має бонусну програму, яка разом із можливістю перевірити наявність товарів у мережевих студіях значно підвищує лояльність клієнтів [14].

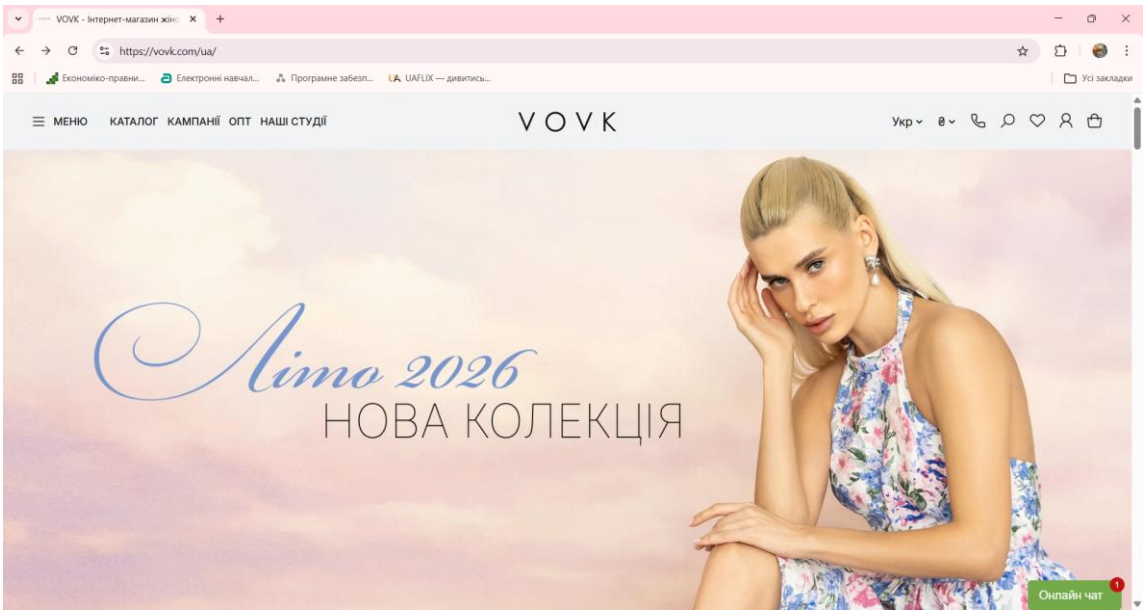


Рисунок 1.3 – Головна сторінка інтернет-магазину VOVK

Аналіз розглянутих ресурсів засвідчує, що при роботі з каталогом успішні онлайн-магазини поєднують зручну навігацію, гнучкі інструменти фільтрації, високоякісний візуальний контент і високу швидкість відгуку системи. Це створює позитивний досвід клієнта, що безпосередньо впливає на показники конверсії та кількість повторних купівель через вебсайт. Вивчення цих вебпродуктів дозволяє окреслити основні цілі для розробки власного вебзастосунку: логічна структура каталогу, адаптивність інтерфейсу, швидкість роботи та зрозумілі механізми взаємодії з товарами.

2 РОЗРОБКА ПРОЄКТУ ВЕБСАЙТУ

2.1 Технічне завдання

Технічне завдання (ТЗ) – це документ, який визначає вимоги до програмного продукту, встановлює функціональні межі системи та служить основою для проєктування та розробки подальших програм. Щоб уникнути розбіжностей у розумінні кінцевого результату, ТЗ погоджується між замовником і виконавцем ще до написання першого рядка коду [15].

У межах кваліфікаційної роботи розроблено вебсайт інтернет-магазину одягу та аксесуарів «Aura». Нижче викладено ключові технічні та функціональні вимоги, що стали основою для проєктування системи.

Основними цілями створення вебзастосунку є надання зручного каталогу товарів із розвиненою фільтрацією, підтримка повного циклу замовлення, починаючи від вибору товару до підтвердження покупки, надання адміністративних інструментів для управління асортиментом і аналіз продажів.

Стилістикою вебсайту обрано мінімалістичний преміальний дизайн. Колірна палітра: фон білий (#FFFFFF), основний текст глибокий антрацит (#1A1A1A), акцентний колір кнопок і деталей м'яке золото (#C5A059). Шрифт Montserrat для заголовків і тексту.

Програмний комплекс вебресурсу реалізовано на основі дворівневої архітектури «клієнт-сервер» із чітким розподілом функціональних обов'язків між технологічними рівнями. Серверна частина побудована на базі мови програмування PHP 8 із використанням реляційної системи керування базами даних MySQL, при цьому розробка здійснювалася на PHP без залучення сторонніх фреймворків задля демонстрації фундаментальних принципів вебпрограмування. Клієнтська частина базується на класичному стеку вебтехнологій, що включає мову семантичної розмітки HTML5, каскадні

таблиці стилів CSS3 для створення адаптивного інтерфейсу та мову програмування JavaScript для забезпечення динамічної взаємодії з користувачем без використання важковагових клієнтських каркасів. Інтеграція, локальне розгортання та автономне тестування всіх зазначених компонентів програмного продукту виконувалися за допомогою кросплатформного середовища розробки XAMPP, яке об’єднує в єдиний локальний вузол вебсервер Apache, СУБД MySQL та інтерпретатор мови PHP, що дозволило повністю імітувати роботу реального сервера на етапі проєктування.

Обраний стек взаємодоповнювальний і узгоджений. PHP створює HTML на сервері, HTML/CSS відображає інтерфейс у браузері, JavaScript дозволяє інтерактивно працювати без перезавантажень, MySQL зберігає структуровані дані, а XAMPP об’єднує всі ці функції в одне середовище без залежностей від зовнішніх джерел. Кожен компонент стеку був обраний відповідно до конкретних вимог проєкту, а не за популярністю. Це включало мінімальну кількість налаштувань, нативну підтримку SQL, повний контроль над кодом і відсутність «чорних скриньок» у вигляді фреймворків і ORM.

У системі є дві окремі частини: публічна (для покупців) і адміністративна (для персоналу). Доступ до адміністративної панелі дозволяється лише після перевірки особи, яка займає відповідну посаду, наприклад адміністратора або касира.

Функціональні вимоги до публічної частини:

- головна сторінка (index.php): hero-секція, три блоки категорій, перелік популярних товарів (Is_Featured = 1) у випадковому порядку;
- каталог (catalog.php): фільтрація за розділом, категорією, брендом, ціновим діапазоном; сортування; breadcrumbs; dropdown-меню в навігації;
- картка товару (product_detail.php): фотогалерея, вибір розміру та кольору, кнопка «Додати до кошика» (неактивна при нульовому залишку);
- кошик (cart.php): список товарів, загальна сума, видалення позицій, вибір способу оплати;

- оформлення замовлення (submit_order.php): валідація даних форми, збереження замовлення, вибір способу оплати.

Функціональні вимоги до адміністративної частини:

- аутентифікація з хешуванням паролів (bcrypt) та захистом від SQL-ін'єкцій (prepared statements);
- дашборд зі статистичними картками та оглядом останніх замовлень;
- CRUD-управління товарами (додавання, редагування, видалення, завантаження фото);
- перегляд бази клієнтів та деталей замовлень;
- аналітичні звіти з графіками (Chart.js) та експортом у форматах XLSX (SpreadsheetML) і PDF.

Для збереження даних створено базу даних, яка повинна містити сім таблиць: Category, Product, Product_images, Customer, Sale, Sale_Details, System_User. Ієрархія категорій реалізується через поле Parent_ID, що дозволяє будувати дерево категорій довільної глибини.

За нефункціональними вимогами сайт коректно відображається на екранах різних розмірів (адаптивний дизайн). Захист від SQL-ін'єкцій (prepared statements), XSS (htmlspecialchars), bcrypt для паролів. Хедер фіксований (sticky) на всіх сторінках. Кнопки мають плавні hover-анімації (transition). Сторінки каталогу завантажуються не довше 2 секунд при локальному розгортанні. Фільтри каталогу передаються через GET-параметри в URL для можливості копіювання та збереження результатів пошуку.

2.2 Діаграма прецедентів

Формалізація вимог і моделювання системи різними засобами нотації UML є основними етапами та основними елементами проєктування. Діаграми прецедентів, класів, послідовності та розгортання описують поведінку, структуру та взаємодію компонентів; технічне завдання визначає

функціональні та нефункціональні вимоги до системи. Проєктування дозволяє заздалегідь узгодити архітектурні рішення та виявити розбіжності між вимогами до початку реалізації програми [16].

Діаграма прецедентів (Use Case Diagram) показує взаємодію акторів системи (покупця, адміністратора та касира) і функціональні можливості, які система надає кожному з них. Прецедент описує те, що система робить для актора, тобто показує функціональні вимоги без подробиць їх реалізації [16].

У системі «Aura» визначено є три актори (рис. 2.1). Покупець – це неавторизований користувач публічної частини сайту, який може переглядати каталог, скласти кошик і зробити замовлення. Адміністратор – це авторизований співробітник, який має повний доступ до адміністраторської панелі. Касир – авторизований співробітник з обмеженим доступом, який не може використовувати налаштування системи. Перелік прецедентів для Покупця: переглянути головну сторінку; відкрити каталог з фільтрами; переглянути картку товару; додати товар до кошика; оформити замовлення з вибором способу оплати.

Перелік прецедентів для Адміністратора та Касира: авторизуватися в системі; переглянути дашборд зі статистикою; управляти товарами (CRUD); переглядати замовлення та клієнтів; формувати звіти; експортувати дані. Адміністратор, крім того, має доступ до налаштувань облікового запису.

Особливу увагу заслуговує зв’язок «включити», який використовується між прецедентами «Оформити замовлення» і «Вибрати спосіб оплати». Цей зв’язок включає послідовність підпорядкованих дій: перевірку наявності товарів, вибір способу оплати та збереження замовлення в базі даних. Крім того, оскільки фільтрація є необов’язковим розширенням основного перегляду, зв’язок «розширити» розташований між «Переглянути каталог» і «Застосувати фільтри».

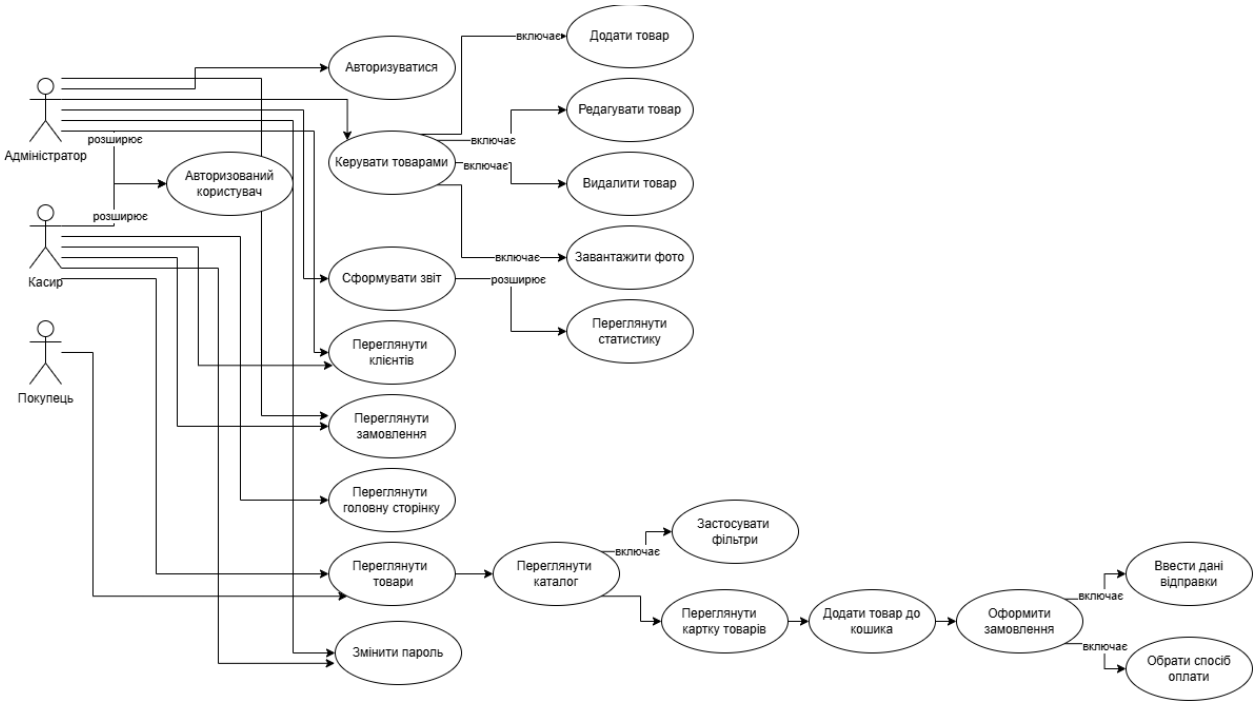


Рисунок 2.1 – Діаграма прецедентів системи «Aura»

2.3 Діаграма класів

Діаграма класів (Class Diagram), що представлена на рисунку 2.3, відображає структуру об'єктів системи, їхні атрибути, методи та зв'язки між собою. Клієнтська логіка проєкту «Aura» реалізована процедурно на PHP без використання ООП-фреймворку, тому класи на діаграмі відповідають логічним модулям (сторінкам або компонентам системи), а не лише файлам [16].

Основою системи є клас Product, який зберігає повні відомості про товар, включно з JSON-масивами розмірів та кольорів. Клас Category реалізує ієрархічну структуру через автономний зовнішній ключ Parent_ID, що дозволяє створювати дерево категорій. Класи Sale та SaleDetails реалізують патерн «замовлення – рядки замовлення»: один об'єкт Sale містить метадані покупки, а множина об'єктів SaleDetails описує кожну позицію. Для реалізації класу Cart використовується PHP-сесія ($_SESSION), а не база даних. Це робить тимчасове зберігання кошика на сервері.

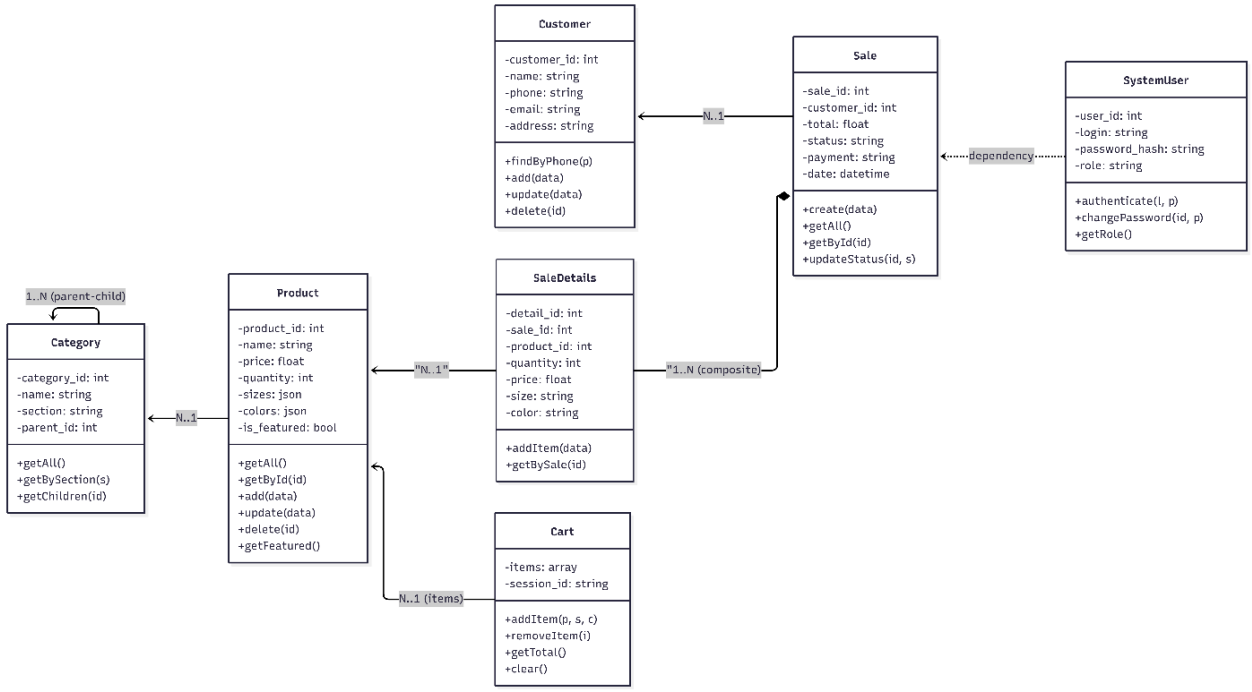


Рисунок 2.2 – Діаграма класів системи «Aura»

Існує кілька типів зв’язків між класами: Product і Category пов’язані з асоціацією «багато до одного», що означає, що кілька товарів належать до однієї категорії; Sale і Customer пов’язані з асоціацією «багато до одного», що означає, що кілька замовлень можуть належати одному клієнту; Sale і SaleDetails пов’язані з асоціацією «один до багатьох» з каскадним з каскадним видаленням. Product і SaleDetails – асоціація «один до багатьох»; SystemUser є незалежним класом, який управляє доступом до адмін-панелі.

2.4 Діаграма послідовності

Діаграма послідовності (Sequence Diagram) показує хронологічний порядок взаємодії між компонентами системи під час виконання певного сценарію (рис. 2.3). На відміну від діаграми прецедентів, вона показує повідомлення між учасниками про те, як процес відбувається крок за кроком [16].

За основу обрано сценарій «Оформлення замовлення», оскільки він охоплює найбільшу кількість компонентів системи та ілюструє взаємодію між клієнтським браузером, серверною логікою та базою даних.

У сценарії беруть участь чотири учасники: Відвідувач (браузер), Вебсервер Apache, PHP-обробники та MySQL. Взаємодія розпочинається з GET-запиту до catalog.php та завершується збереженням підтвердженого замовлення в базі даних. Після завершення платежу сервер очищує кошик у сесії та повертає браузеру сторінку підтвердження замовлення.

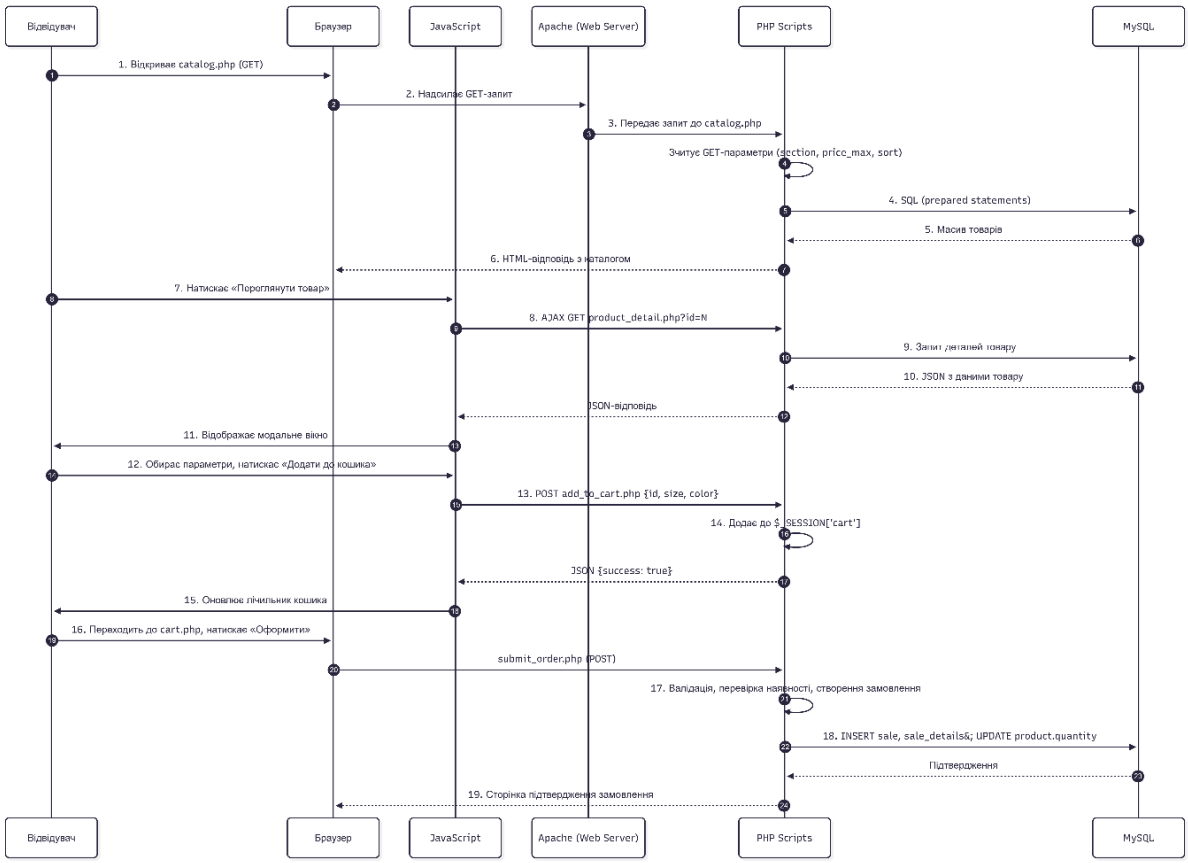


Рисунок 2.3 – Діаграма послідовності: сценарій «Перегляд каталогу та оформлення замовлення»

2.5 Діаграма розгортання

Діаграма розгортання (Deployment Diagram) описує фізичну або логічну архітектуру системи з точки зору вузлів, на яких розміщуються компоненти, та протоколів взаємодії між ними [16]. В проєкті «Aura» є чотири вузли, що представлені на рисунку 2.4. Це клієнтський вузол (браузер), вебсервер Apache + PHP, база даних MySQL і середовище розробки XAMPP.

Клієнтський вузол відповідає за візуалізацію HTML/CSS та виконання JavaScript у браузері. Взаємодія з сервером відбувається по протоколу HTTP/1.1 через стандартні GET- та POST-запити, а також AJAX-запити для динамічних елементів (кошик, модальні вікна). Центральним компонентом вебсервера є Apache, який приймає вхідні запити, PHP-інтерпретатор виконує логіку сервера, а PHP-сесії забезпечують тимчасове зберігання стану кошика. Розширення MySQL, яке використовує готові заяви, дозволяє вузлу вебсервера та базі даних взаємодіяти через протокол TCP/IP.

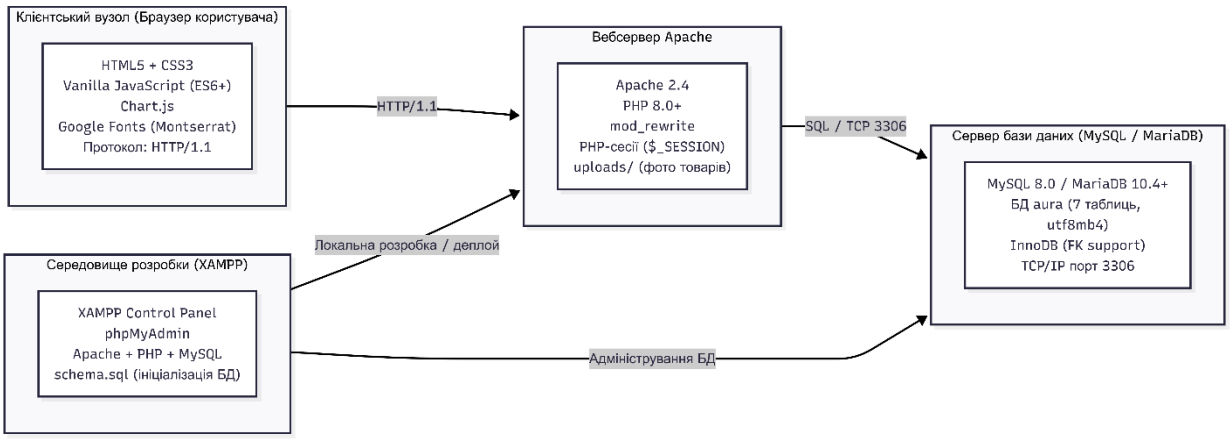


Рисунок 2.4 – Діаграма розгортання системи «Aura»

2.6 Проєктування бази даних

Проєктування бази даних – це процес визначення структури таблиць, їхніх характеристик, типів даних, первинних і зовнішніх ключів, а також

зв’язків між сутностями. Коректно спроєктована база даних забезпечує цілісність даних, мінімізує надлишковість (нормалізація) та спрощує написання SQL-запитів.

База даних системи «Aura» побудована відповідно до вимог третьої нормальної форми (3НФ). Розроблено сім взаємопов'язаних таблиць Каскадне видалення залежних записів забезпечується зовнішніми ключами, які мають обмеження ON DELETE CASCADE. Наприклад, коли товар видаляється, всі зображення, пов’язані з ним, автоматично видаляються з таблиці product_images. Усі таблиці використовують рушій InnoDB і кодування utf8mb4 для підтримки транзакцій і зовнішніх ключів.

Ієрархія категорій реалізована через самопосилальний зовнішній ключ Parent_ID у таблиці category (рис. 2.5). Категорії верхнього рівня (Жінкам, Чоловікам, Аксесуари) мають Parent_ID = NULL. Підкатегорії посилаються на батьківський розділ, що дозволяє будувати дерево категорій необмеженої глибини без додаткових таблиць.

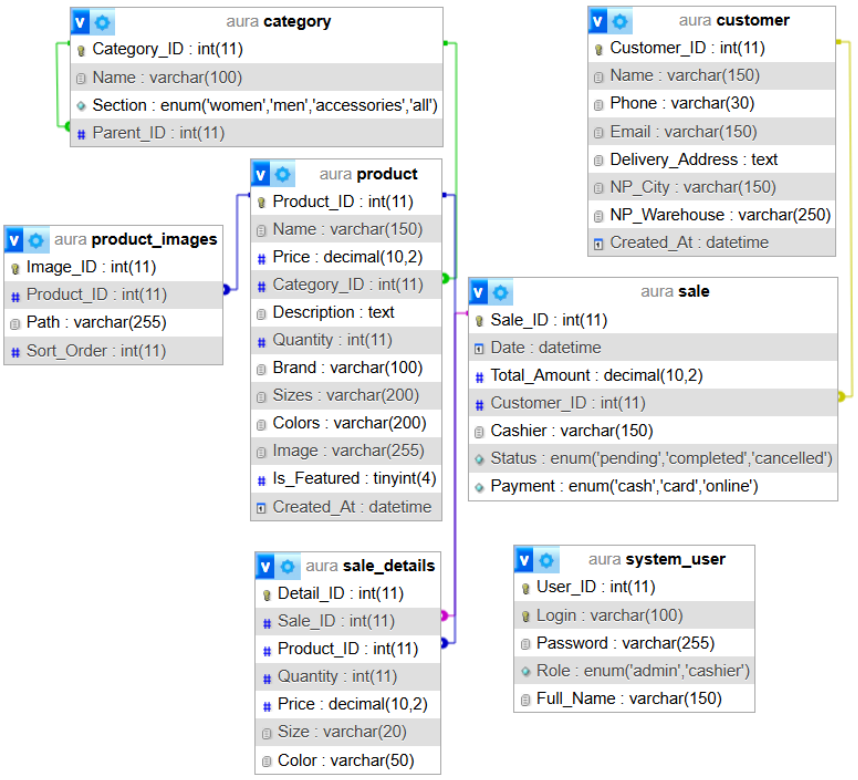


Рисунок 2.5 – Схема бази даних системи «Aura» (ключові таблиці)

Поля Sizes та Colors таблиці product зберігають JSON-масиви безпосередньо у полі VARCHAR(200). Це рішення полегшує читання та запис вибору товару, не створюючи окремих таблиць для кожного атрибута. Коли користувач робить замовлення, розмір і колір, які він вибрав, записується в рядках таблиці sale_details. Це гарантує, що дані залишаться незмінними навіть після редагування асортименту товарів.

У таблиці system_user використовуються лише bcrypt-хеші для зберігання паролів, яке представлено полям Password VARCHAR(255). Функція PHP password_hash() з алгоритмом PASSWORD_BCRYPT створює окрему «сіль» для кожного хешу, що унеможливлює використання таблиць атак, які були обчислені раніше. password_verify() використовується для перевірки пароля при вході, і він повертає true лише за умови, що хеш з даними, які були введені, збігається.

Таблиця sale реалізує часткову денормалізацію: поле Total зберігає загальну суму замовлення, хоча її теоретично можна отримати підсумовуванням рядків sale_details. Це свідоме рішення дозволяє швидко отримувати статистику замовлень без ресурсомістких агрегуючих запитів при формуванні звітів та дашборду.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Архітектура та технології реалізації

Система «Aura» побудовано на класичній серверній архітектурі без застосування фреймворків, тобто PHP-код виконується безпосередньо на сервері та генерує HTML-відповідь, яка передається у браузер клієнта (табл. 3.1). Такий метод, відомий як «процедурний PHP», широко використовується для навчальних проєктів, оскільки він надає повний контроль над кожним рядком коду, не вимагаючи вивчення конкретного синтаксису фреймворку.

Таблиця 3.1 – Технологічний стек системи «Aura»

Технологія / Інструмент	Роль у проєкті
PHP	Серверна логіка, обробка HTTP-запитів, робота з БД
MySQL / MariaDB	Реляційна база даних
HTML5	Семантична розмітка сторінок
CSS3	Зовнішній вигляд, анімації, адаптивність
JavaScript	Інтерактивність: кошик, модальні вікна, динамічні фільтри
Chart.js	Інтерактивні графіки у звітах адмін-панелі
SpreadsheetML	Генерація Excel-файлів без сторонніх бібліотек
XAMPP	Локальне середовище розробки (Apache + MySQL + PHP)

Структура файлів проєкту представлена в таблиці 3.2. Архітектурна система складається з двох окремих зон: публічної (для покупців) і адміністративної (для працівників магазину). Файл login.php служить точкою ходу в адміністративну частину. На початку виконання кожна захищена сторінка перевіряє наявність активної сесії та роль користувача. Якщо авторизації немає, сторінка перенаправляє автоматично на сторінку входу.

Підключення до бази даних зосереджено в файлі db.php. Це дозволяє змінювати параметри з’єднання в одному місці, не торкаючись решти файлів

проєкту. З'єднання встановлюється з чітким зазначенням кодування utf8mb4, що гарантує правильну роботу з кириличним текстом.

Таблиця 3.2 – Структура файлів проєкту

Файл / Папка	Призначення
index.php	Головна сторінка: hero-секція, категорії, популярні товари
catalog.php	Каталог: фільтри, сортування, breadcrumbs, dropdown-меню
product_detail.php	Модальне вікно з деталями товару
cart.php	Кошик, форма доставки, вибір способу оплати
submit_order.php	Обробка та підтвердження замовлення
login.php	Авторизація в адмін-панелі
home.php	Дашборд з KPI-картками та статистикою
products.php / edit_product.php	CRUD-управління товарами
customers.php / edit_customer.php	Управління клієнтами
sales.php / order_view.php	Перегляд замовлень та їх деталей
report.php / export.php	Аналітика та експорт у Excel/PDF
db.php	Підключення до MySQL
style.css	Всі стилі сайту
schema.sql	Структура бази даних та тестові дані
uploads/categories/	Фото тайлів категорій (women.jpg, men.jpg, acc.jpg)
uploads/products/	Фотографії товарів, завантажені через адмін-панель

Для всіх PHP-файлів проєкту та баз даних використовується кодування UTF-8, також відоме як utf8mb4. У серверній частині є понад 20 PHP-скриптів, а файл стилів style.css містить 1469 рядків. У базі даних є 7 таблиць і 13 тестових товарних позицій.

У дизайні використовується преміальна темна колірна схема; основний фон – темний антрацит (#1A1A1A), золотистий (#C5A059) – колір кнопок і деталей, а основний текст – #FFFFFF/#E0E0E0. Через Google Fonts доступні чотири накреслення шрифту Montserrat (400, 500, 600, 700). Кольорові

значення, відступи та параметри анімації знаходяться в центрі файлу стилів через CSS-змінні (:root).

3.2 Реалізація функціональних вимог

Функціональні вимоги визначають конкретну поведінку системи, включаючи її дії, прийом даних і відповідь. У межах кваліфікаційної роботи функціональні вимоги було поділено на дві категорії: одна стосується публічної частини вебсайту, доступної для всіх користувачів, а інша стосується адміністративної панелі, яка призначена для працівників магазину. Нижче описано реалізовані функціональні вимоги до публічної та адміністративної частин.

Три структурні блоки складають головну сторінку (рис. 3.1), яка є точкою входу для відвідувача. На першому місці знаходиться hero-секція, яка складається з великого банеру з назвою магазину, слоганом і кнопкою переходу до каталогу. Другий блок містить три тайли-категорії: «Жінкам», «Чоловікам» і «Аксесуари» (рис. 3.2). Кожен тайл веде до відфільтрованого каталогу відповідного розділу. Третій – перелік популярних товарів. Вісім позицій у базі даних відображаються з прапорцем Is_Featured = 1. За допомогою оператора ORDER BY RAND() в SQL можна рандомізувати порядок виведення товарів для кожного завантаження сторінки, щоб запобігти одноманітності.

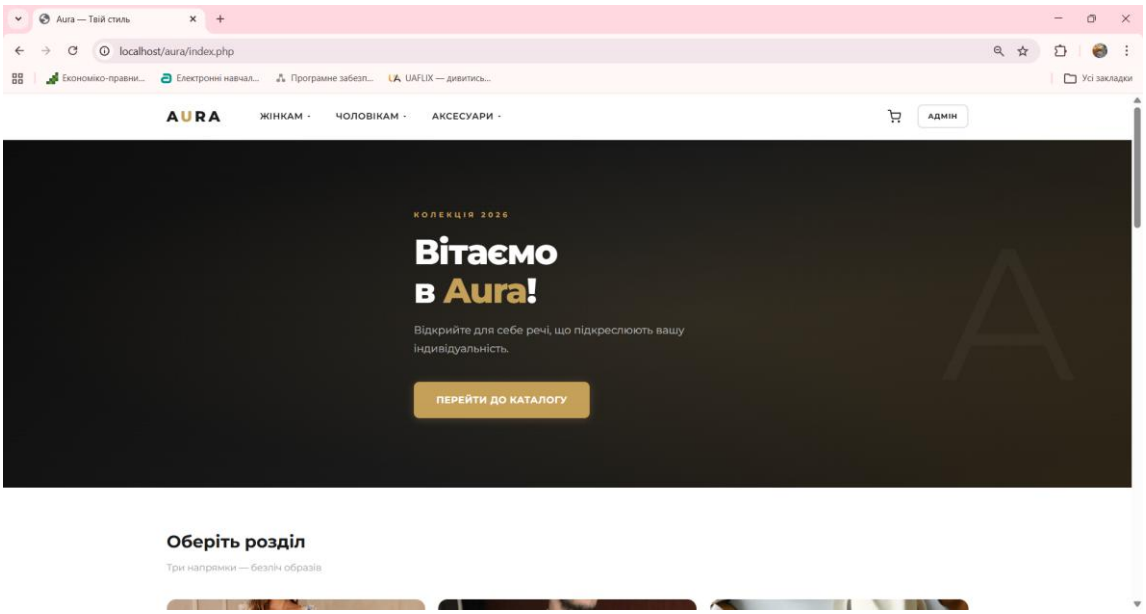


Рисунок 3.1 – Головна сторінка вебсайту

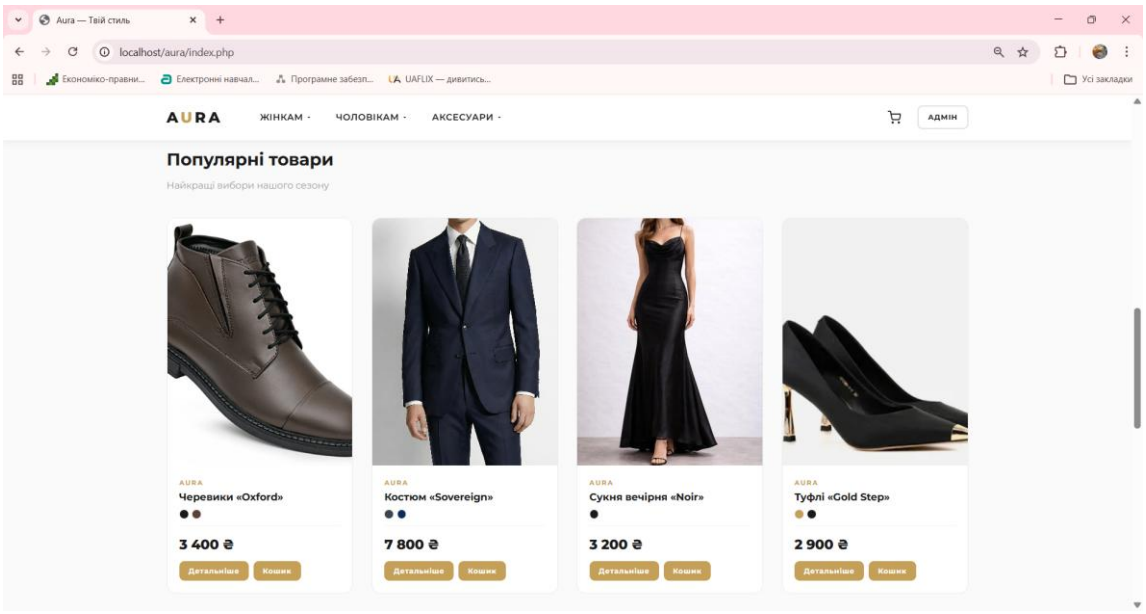


Рисунок 3.2 – Секція з популярними товарами на головній сторінці

Каталог пропонує повну систему навігації та фільтрації (рис. 3.3). Відвідувач може одночасно звужувати вибірку за різними параметрами, наприклад розділом, підкатегорією, брендом і ціновим діапазоном. Передбачено сортування результатів за новизною, назвою або ціною. Це було зроблено в порядку зростаючого чи спадного зросту.

Також реалізовано breadcrumbs, тобто навігаційний ланцюжок, схожий на «Каталог → Жінкам → Сукні», який використовується для орієнтації в ієрархії категорій. У шапці вебсайту є спадне меню, яке дозволяє відкрити спадне підменю, яке містить відповідні підкатегорії, коли ви натискаєте на назву розділу (рис. 3.4). Усі параметри фільтрації передаються через GET-рядок URL, а PHP-скрипт динамічно створює SQL-запит на їх основі, використовуючи готові заяви для захисту від SQL-ін’єкцій.

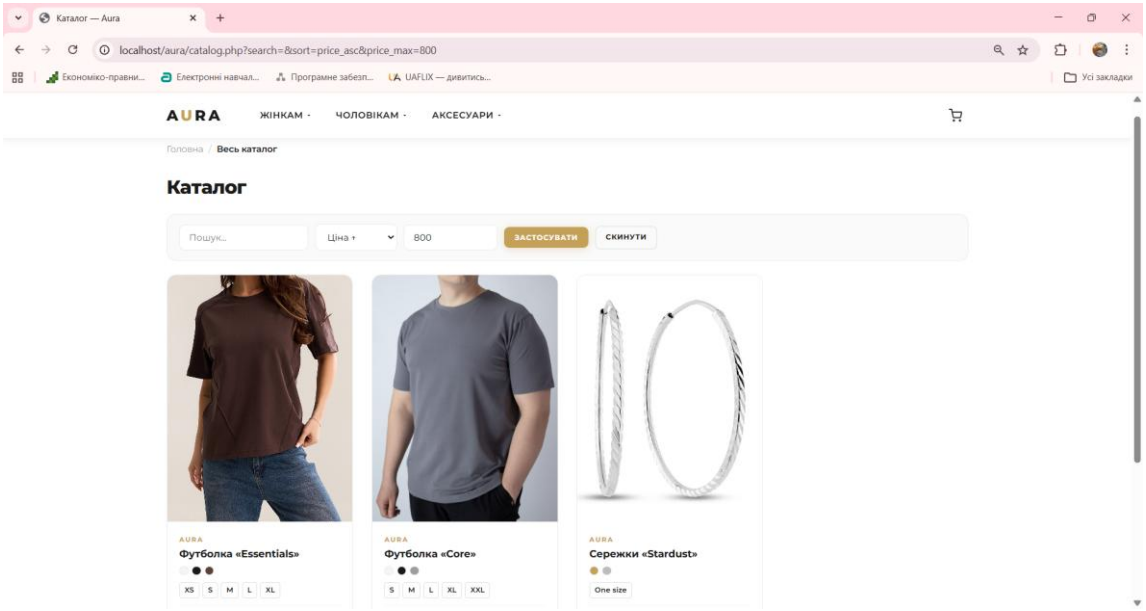


Рисунок 3.3 – Каталог сайту

У модальному вікні поверх каталогу можна відвідати сторінку з деталями товару, не переходячи на нову адресу (рис. 3.5). Вона містить галерею фотографій із перемиканням між ними, а також повний опис місця, назву бренду та поточну ціну. Покупець обирає розмір і колір із запропонованих відтінків.

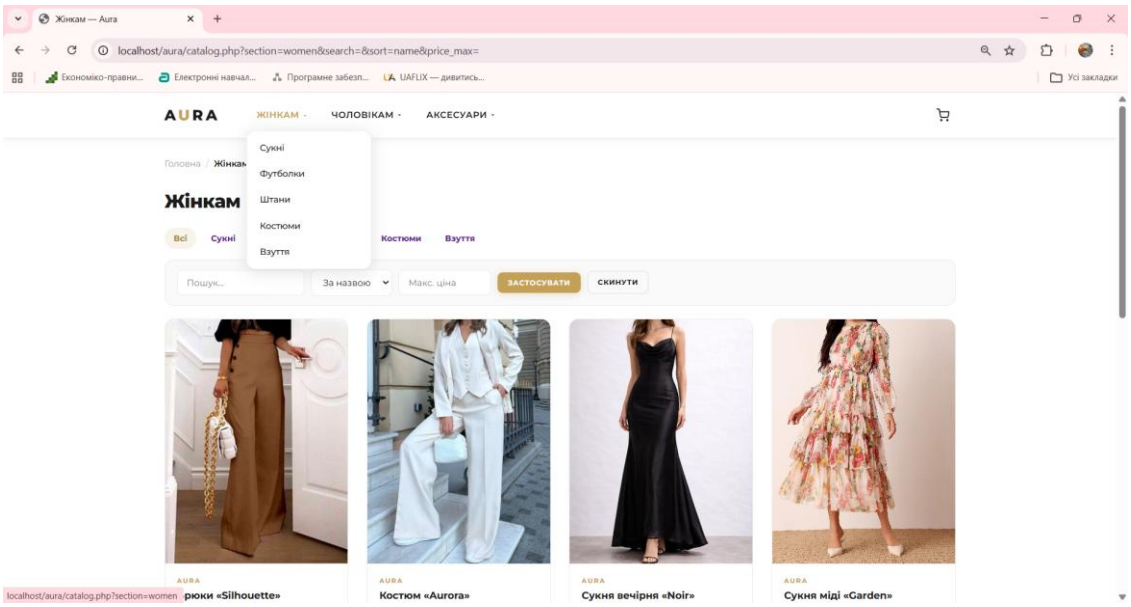


Рисунок 3.4 – Спадне меню навігаційної панелі сайту

Залишок на складі конкретної позиції відображається безпосередньо на картці, що запобігає оформленню замовлення на товар, який не є на складі. Кнопка «Додати до кошика» працює лише коли немає нічого в кошику. Якщо товар закінчився, кнопка блокується та відображається повідомлення про це (рис. 3.6).

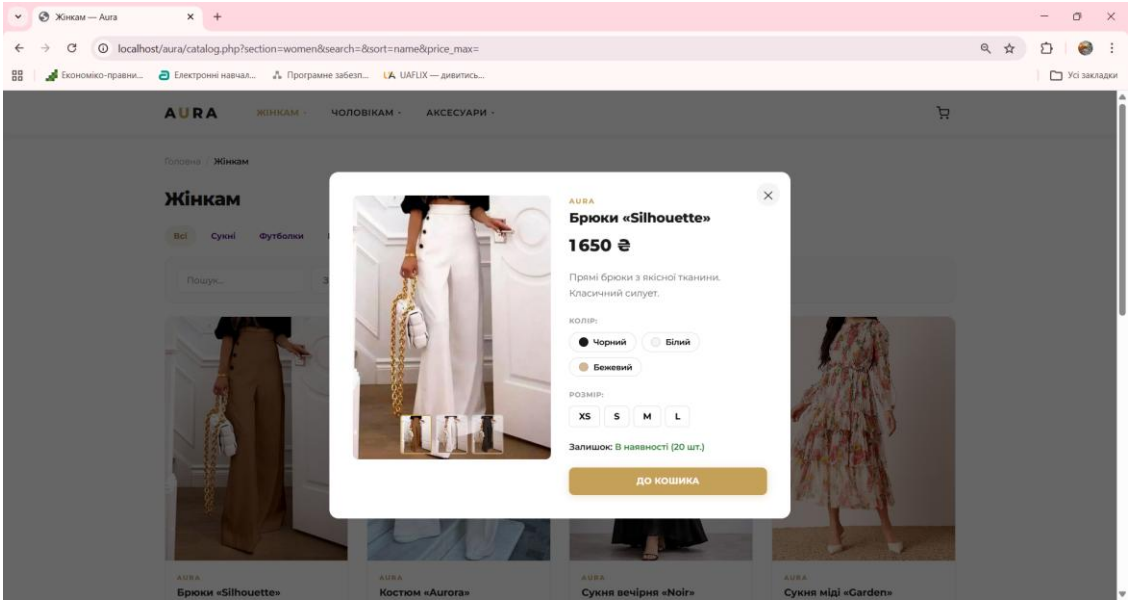


Рисунок 3.5 – Спливаюче вікно з деталями товару

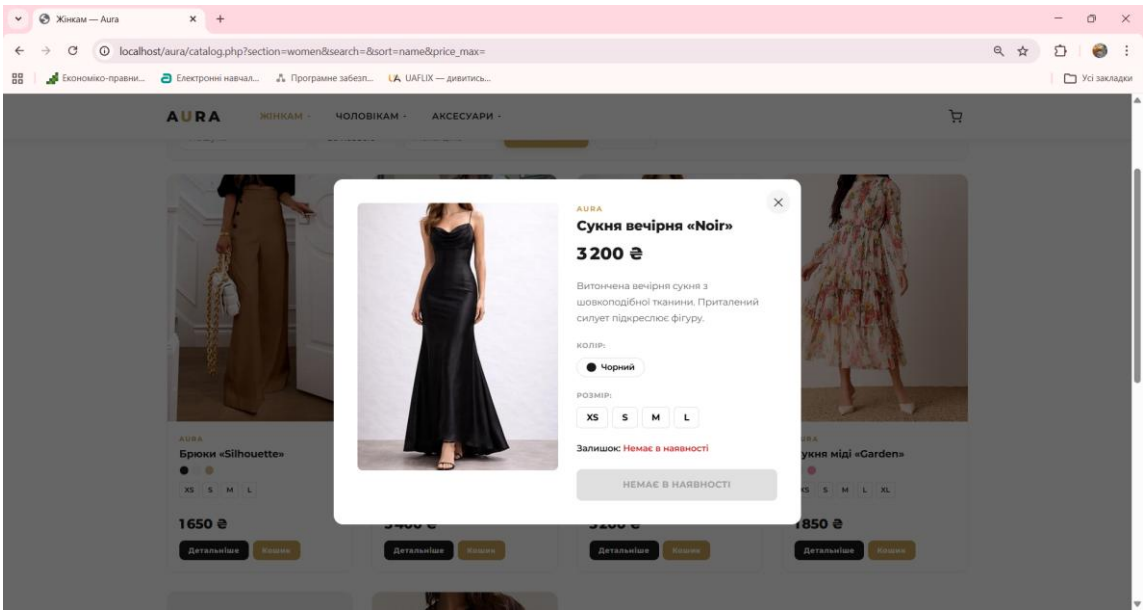


Рисунок 3.6 – Повідомлення про закінчення товару

На сторінці кошика відображається список усіх доданих товарів разом із фотографією, назвою, обраним розміром і кольором і ціною (рис. 3.7). Загальна вартість замовлення визначається автоматично. Клієнт має право видалити будь-яку позицію зі списку. Перед оформленням замовлення ви можете вибрати оплату готівкою при отриманні або карткою онлайн.

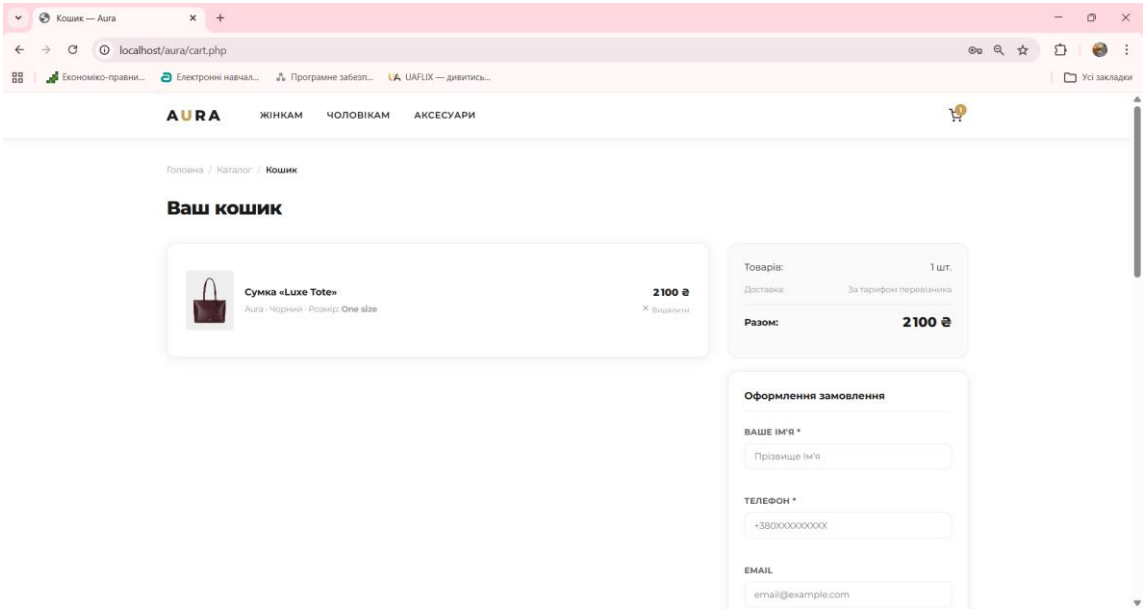


Рисунок 3.7 – Сторінка кошика

На сторінці відображається UI-картка користувача, яка містить поля для введення реквізитів, автоматичне визначення бренду платіжної картки Visa або MasterCard, а також симуляція обробки платежу (рис. 3.8). Технічно вміст кошика зберігається в PHP-сесії, яка прив’язана до браузера відвідувача.

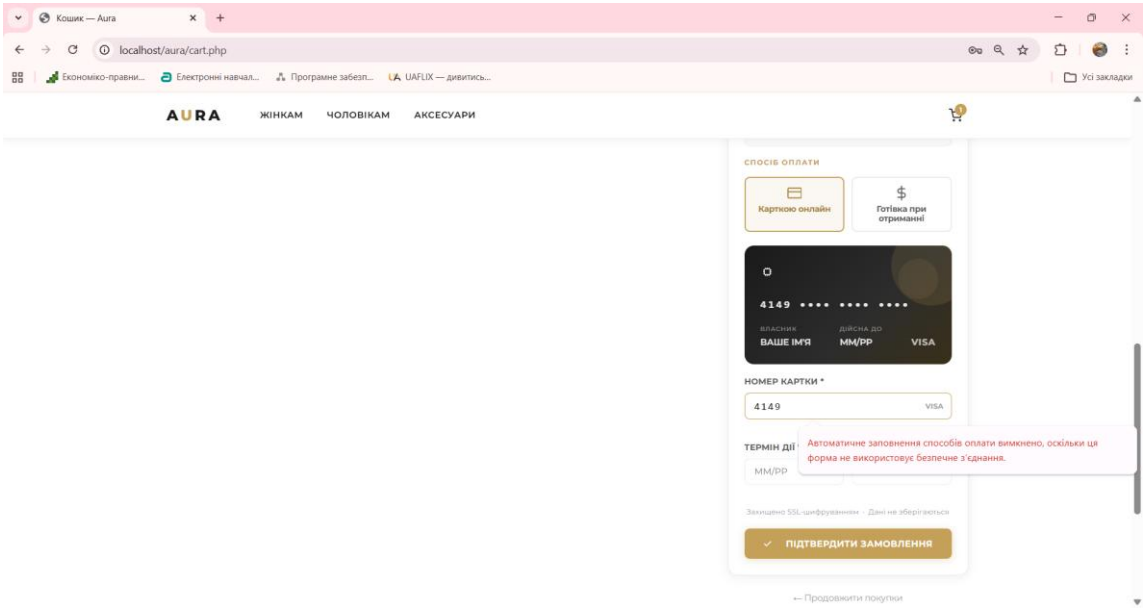


Рисунок 3.8 – UI-картка з полями для введення реквізитів

Після підтвердження покупки файл-обробник запускає та виконує повний цикл обробки. Він перевіряє правильність введених даних у формі (наприклад, ім’я, телефон і адреса доставки), перевіряє наявність вибраних товарів на складі, ідентифікує або створює запис клієнта в таблиці клієнта, зберігає замовлення в таблиці продажів із способом оплати, сумою та статусом, а також створює окремі рядки для кожної позиції (рис. 3.9). По завершенні процесу обробки кількість відповідних товарів у таблиці product зменшується на замовлену кількість, сесійні дані кошика знищуються, а покупцю надсилається повідомлення про підтвердження замовлення.

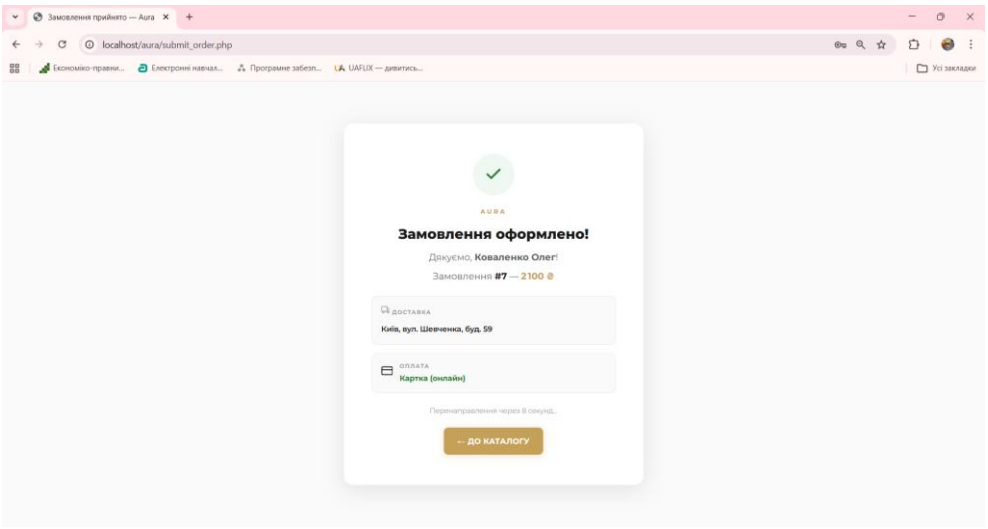


Рисунок 3.9 – Повідомлення про підтвердження замовлення

Вхід до адміністративної панелі здійснюється через форму логін/пароль (рис. 3.10). Паролі зберігаються у хешованому вигляді за алгоритмом bcrypt – криптографічною хеш-функцією, навмисно уповільненою для ускладнення атак методом перебору. Вбудована PHP-функція password_verify виконує перевірку. Під час авторизації всі SQL-операції виконуються через підготовлені заяви, що унеможливлює SQL-ін’єкції. Кожна сторінка адмін-панелі містить перевірку сесії та ролі користувача. Якщо хтось отримує несанкціонований доступ, він буде перенаправлений на сторінку входу. Крім того, форма містить індикатор надійності та кнопку перемикання видимості поля пароля.

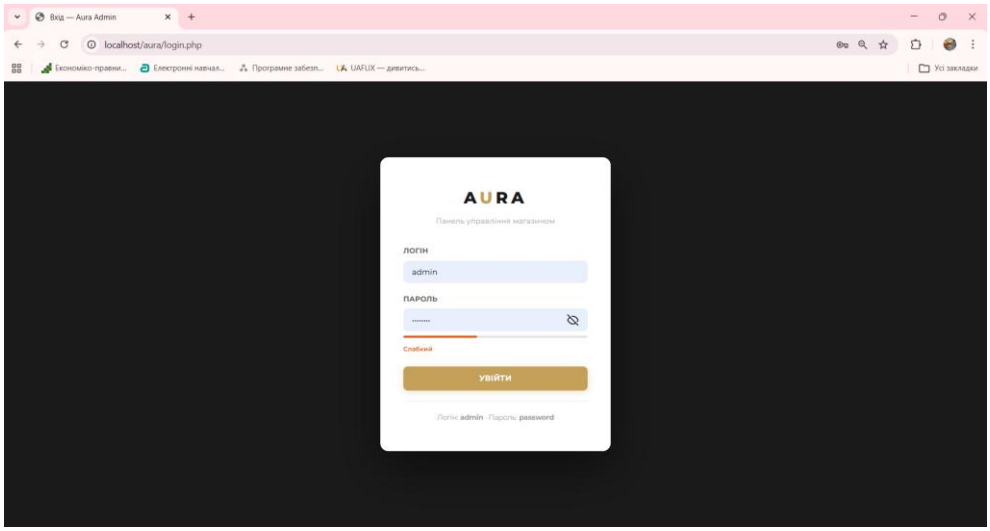


Рисунок 3.10 – Вхід до адміністтативної панелі

Як раніше було зазначено, адміністративна панель містить перевірку ролі користувача. Якщо входити адміністратором, в доступі будуть всі функції, такі як перегляд і статистика замовлень, редагування/додавання/видалення товарів, перегляд та експорт звітів (рис. 3.11). Але для касира буде доступний тільки перегляд замовлень (рис. 3.12).

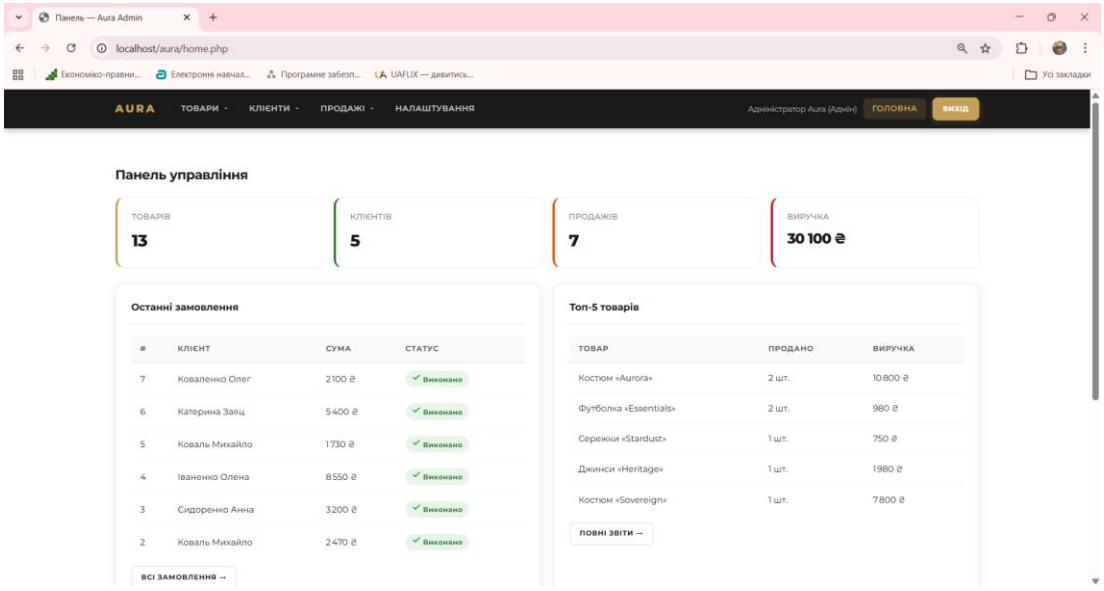


Рисунок 3.11 – Головна сторінка адмін-панелі для адміністратора

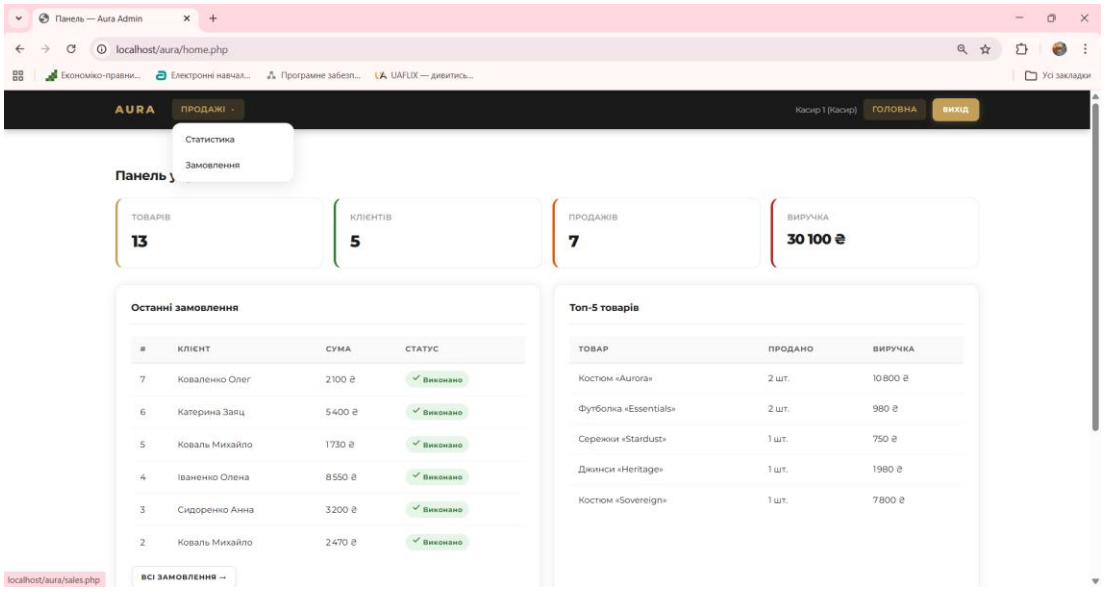


Рисунок 3.12 – Головна сторінка адмін-панелі для касира

Чотири KPI-картки на головному екрані панелі показують оперативну статистику: загальна кількість товарів в асортименті, кількість зареєстрованих клієнтів, кількість замовлень і загальна виручка в гривнях. Нижче представлена таблиця останніх шести замовлень, яка містить дату, ім’я клієнта, суму та поточний статус (рис. 3.13). Крім того, вона містить список товарів із критично низьким залишком, який необхідний для своєчасного поповнення запасів, що складається з трьох або менше одиниць.

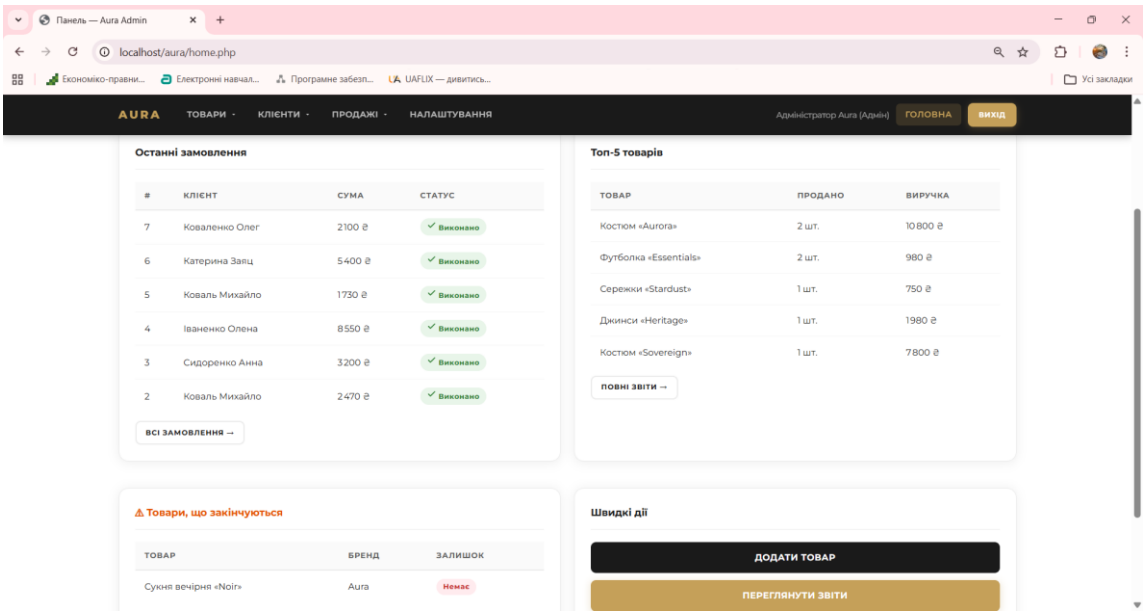


Рисунок 3.13 – Головна сторінка адмін-панелі home.php

Адміністратор може виконувати широкий спектр операцій щодо асортименту, включаючи додавання нових товарів із зазначенням назви, ціни, кількості, категорії, опису, розмірів і кольорів, а також видалення товарів із каталогу (рис. 3.14). Можна завантажувати фотографії для кожного продукту. Файли зберігаються в папці uploads/, а шляхи до них записуються в таблицю product_images. Прапорець Is_Featured дозволяє маркувати продукт як рекомендований; ці позиції відображаються на головній сторінці. Усі дії виконуються через захищені форми, а вхідні дані валідуються сервером.

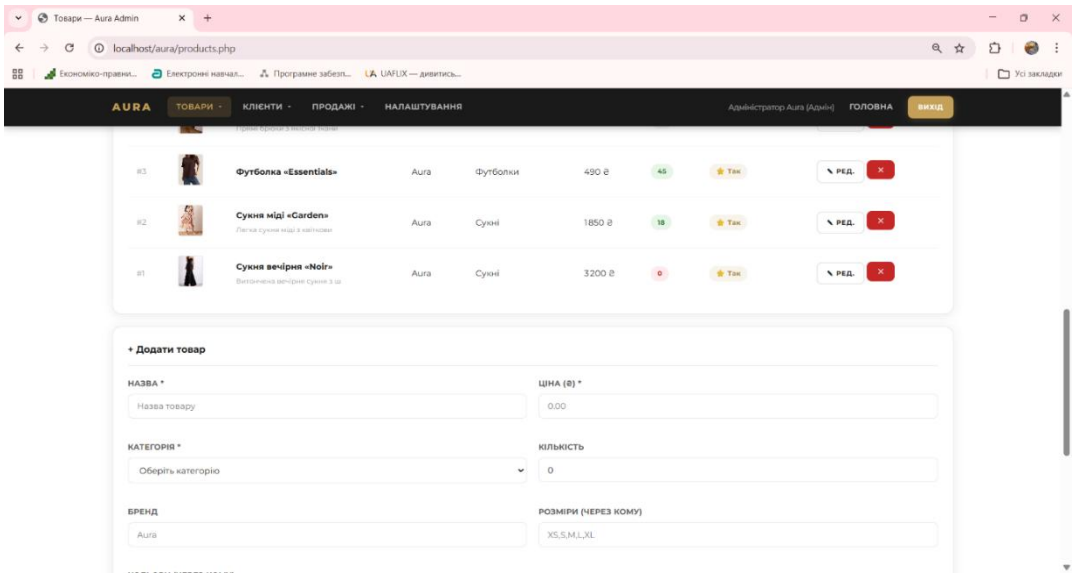


Рисунок 3.14 – CRUD-управління товарами

Також адміністратор має доступ до списку покупців, які зареєструвалися, включаючи імена, контактні дані та адресу доставки. Кожне замовлення міститься в окремому розділі, де наведено перелік позицій, обрані розміри та кольори, загальну суму, спосіб оплати та поточний статус (рис. 3.15). Це дозволяє робітникам стежити за станом відправлень і оперативно реагувати на будь-які зміни.

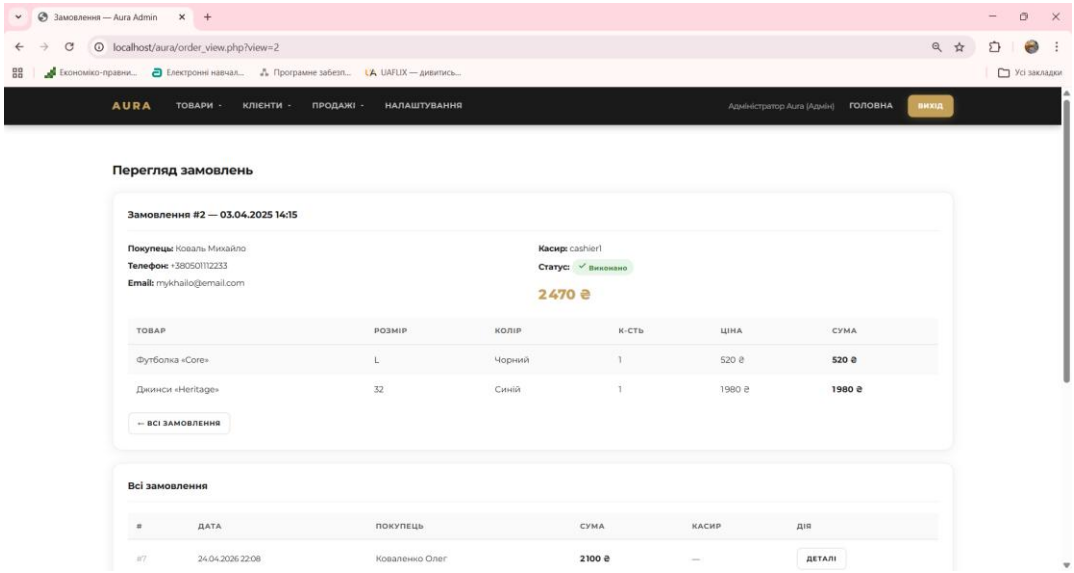


Рисунок 3.15 – Деталі замовлення

На основі бібліотеки Chart.js, аналітичний відділ надає наочні графіки: показники продажів за певний період часу, структура виручки за категоріями та найкращі позиції за кількістю реалізованих одиниць. Дані для побудови графіків отримуються шляхом агрегування запитів SQL до таблиць продажу та деталей продажу. Звітні дані можна експортувати в два формати: XLSX (у форматі SpreadsheetML, який не потребує сторонніх бібліотек на стороні сервера) та PDF (рис. 3.16). Завантажені файли готові до друку або передачі керівництву без додаткової обробки.

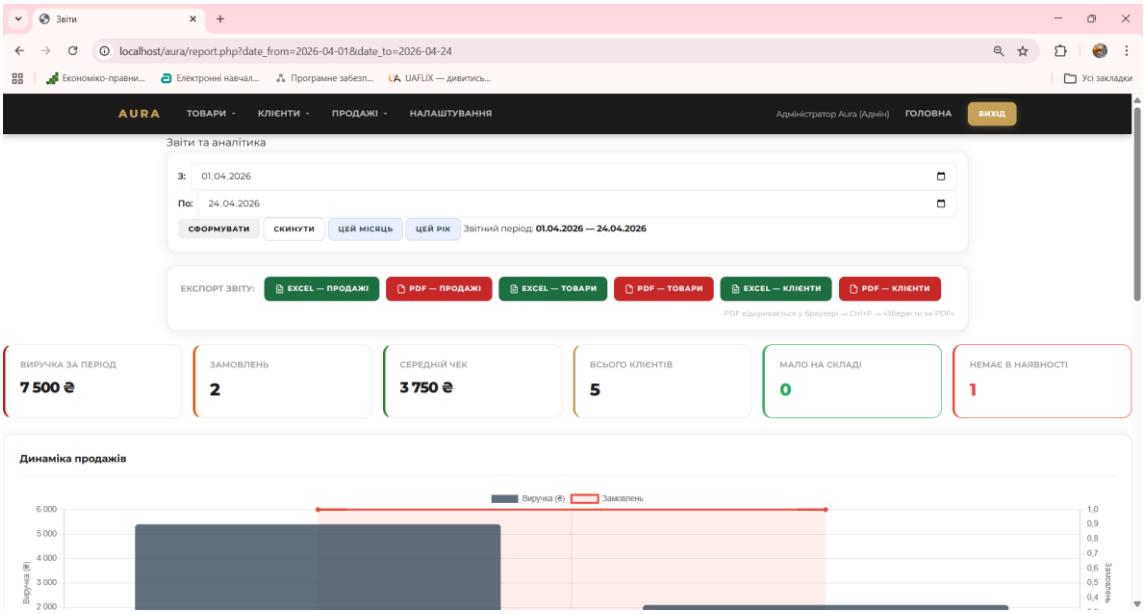


Рисунок 3.16 – Сторінка з аналітичними звітами

У результаті всі функціональні вимоги, які були заявлені, будо реалізовано за допомогою окремих PHP-файлів, об’єднаних спільною базою даних та одним шаром стилізації. Клієнти можуть легко переглядати каталог і підтверджувати замовлення, а адміністративна панель надає працівникам інструменти для аналізу продажів, відстеження замовлень і керування асортиментом.

3.3 Тестування вебсайту

Обов'язковим етапом розробки програмного продукту є тестування. Це дозволяє виявити відмінності між вимогами, сформульованими на стадії проєктування, і тим, як система поводиться в реальному житті. Для оцінки кваліфікації використовувалися два основні методи: модульне та інтеграційне.

Модульне тестування (unit testing) передбачає перевірку окремих функціональних одиниць системи незалежно від решти коду. У цьому проєкті окремо перевірено основні PHP-функції та JavaScript-компоненти.

На сервері використовувалися функції password_hash() і password_verify для перевірки правильності хешування та верифікації паролів, валідація форми (перевірка типів і діапазонів значень за допомогою intval(), floatval(), і trim()), а також функції формування динамічних SQL-запитів із фільтрами у файлі catalog.php.

На клієнтській сторінці окремо протестовано логіку JavaScript. Протестовано функції управління кошиком, такі як додавання, видалення та перерахунок суми; дії модального вікна картки товару; і перевірка форми оформлення замовлення перед її відправкою на сервер.

Інтеграційне тестування оцінює взаємодію між компонентами системи, такими як серверна PHP-логіка та MySQL, PHP-сесії та захист сторінок адмін-панелі, JavaScript та серверні обробники AJAX-запитів (add_to_cart.php, remove_from_cart.php).

Зокрема перевірено наскрізний сценарій, починаючи з відкриття каталогу покупцем і закінчуючи фіксацією замовлення в базі даних: використання фільтрів, відкриття картки товару, додавання її до кошика, заповнення форми та збереження за допомогою submit_order.php. Паралельно перевірялася адміністративна частина, яка складалася з авторизації, CRUD-операцій з товарами, перегляду замовлень і створення звітів. Крім того, перевірено рольову модель доступу. Підтвердилося, що касир не має доступу

до сторінки звітів і редагувати товари за допомогою прямого введення URL або меню.

На рисунку 3.17 наведено тест-кейси, що охоплюють функціональність публічної частини, адміністративної панелі та механізмів безпеки вебсайту «Aura».

Сайт http://localhost/aura/	Google Chrome Версія	Microsoft Edge Версія 147.0.3912.72	Opera One Версія: 113.0.5230.86	Примітки/Посилання на баг-репорт
Авторизація				
Перевірити успішну авторизацію адміністратора (admin/password)	Passed	Passed	Passed	
Перевірити успішну авторизацію касира (cashier1/password)	Passed	Passed	Passed	
Валідаційні повідомлення:				
Порожнє поле логіну або паролю	Passed	Passed	Passed	
Невалідний пароль (неправильний, але непорожній)	Passed	Passed	Passed	
Прямий перехід на захищену сторінку без авторизації, редирект на login.php	Passed	Passed	Passed	
Рольова модель доступу				
Касир не бачить розділ «Звіти» у меню	Passed	Passed	Passed	
Касир не може відкрити report.php напряму через URL	Passed	Passed	Passed	
Касир не бачить розділ «Управління товарами» у меню	Passed	Passed	Passed	
Адміністратор має доступ до всіх розділів панелі	Passed	Passed	Passed	
Головна сторінка (index.php)				
Відображення hero-секції, трьох тайлів категорій та 8 рекомендованих товарів	Passed	Passed	Passed	
Клік на тайл «Жінкам», перехід до каталогу з фільтром section=women	Passed	Passed	Passed	
Перевірити адаптивне відображення на ширині 375 px (мобільний)	Passed	Passed	Passed	
Каталог товарів (catalog.php)				
Фільтрація за розділом «Жінкам» — URL містить ?section=women	Passed	Passed	Passed	
Фільтрація за підкатегорією «Сукні»	Passed	Passed	Passed	
Фільтрація за максимальною ціною (повзунок 1000 грн)	Passed	Passed	Passed	
Пошук товару за назвою «Сукня»	Passed	Passed	Passed	
Сортування за ціною (зростання)	Passed	Passed	Passed	
Breadcrumbs відображаються коректно: Каталог--> Жінкам-->Сукні	Passed	Passed	Passed	
Dropdown-меню з підкатегоріями при наведенні на розділ	Passed	Passed	Passed	
Картка товару (product_detail.php)				
Відкриття модального вікна картки товару при кліку в каталозі	Passed	Passed	Passed	
Галерея фото: перемикання між кількома зображеннями	Passed	Passed	Passed	
Відображення назви, ціни, опису, залишку на складі	Passed	Passed	Passed	
Вибір розміру та кольору перед додаванням у кошик	Passed	Passed	Passed	
Валідаційні повідомлення:				
Спроба «Додати до кошика» без вибору розміру-->підказка	Passed	Passed	Passed	
Кошик (cart.php)				
Додати товар до кошика: лічильник у хедері збільшується на 1	Passed	Passed	Passed	
Видалити товар з кошика: позиція зникає, сума перераховується	Passed	Passed	Passed	
Оновити кількість позиції: загальна сума перераховується	Passed	Passed	Passed	
Відновлення кошика після несподіваного закриття браузера (авторизований користувач)	Passed	Passed	Passed	
Перехід до оформлення замовлення з непорожнім кошиком	Passed	Passed	Passed	
Оформлення замовлення (submit_order.php)				
Оформлення з коректними даними (ім'я, телефон, email, адреса, оплата готівка) --> замовлення в БД	Passed	Passed	Passed	
Оформлення з вибором оплати «Онлайн карткою», анімована картка відображається	Passed	Passed	Passed	
Валідаційні повідомлення:				
Порожнє поле «Телефон» --> HTML-валідація блокує відправку	Passed	Passed	Passed	
Некоректний email --> HTML-валідація блокує відправку	Passed	Passed	Passed	
Адмін-панель: Управління товарами (products.php)				
Додавання нового товару з фото — товар з'являється в каталозі	Passed	Passed	Passed	
Редагування характеристик наявного товару	Passed	Passed	Passed	
Позначення товару як «Рекомендований» (Is_Featured = 1) — відображається на головній	Passed	Passed	Passed	
Видалення товару — каскадне видалення фото з БД (ON DELETE CASCADE)	Passed	Passed	Passed	
Валідаційні повідомлення:				
Спроба додати товар з від'ємною ціною --> повідомлення про помилку	Passed	Passed	Passed	
Спроба зберегти товар без назви --> поле підсвічується	Passed	Passed	Passed	
Адмін-панель: Замовлення (sales.php / order_view.php)				
Список замовлень відображається з датою, клієнтом, сумою та статусом	Passed	Passed	Passed	
Фільтрація замовлень за датою (від–до)	Passed	Passed	Passed	
Деталі замовлення: перелік товарів, кількість, розміри, кольори, дані покупця	Passed	Passed	Passed	
Адмін-панель: Звіти та аналітика (report.php)				
Генерація звіту «Продажі» --> лінійний графік Chart.js відображається	Passed	Passed	Passed	
Генерація звіту «Товари» --> стовпчиковий графік топ-10 товарів відображається	Passed	Passed	Passed	
Завантаження звіту у форматі Excel (.xlsx) — файл коректно відкривається	Passed	Passed	Passed	
Завантаження звіту у форматі PDF (CSS @media print) — файл коректно відкривається	Passed	Passed	Passed	
Налаштування акаунту (settings.php)				
Успішна зміна пароля (поточний --> новий --> підтвердження) --> пароль хешується bcrypt	Passed	Passed	Passed	
Валідаційні повідомлення:				
Невірний поточний пароль --> повідомлення про помилку, зміни не зберігаються	Passed	Passed	Passed	
Новий пароль та підтвердження не збігаються --> повідомлення про помилку	Passed	Passed	Passed	
Безпека (OWASP)				
SQL-ін'єкція у полі пошуку каталогу (' OR '1'='1) --> prepared statement нейтралізує	Passed	Passed	Passed	
XSS у полі імені замовлення (<script>alert('xss')</script>) --> htmlspecialchars() екранує	Passed	Passed	Passed	
Пароль зберігається у БД у вигляді bcrypt-хешу (не у відкритому вигляді)	Passed	Passed	Passed	
Додаткова перевірка				
Адаптивне відображення каталогу на мобільному екрані (375 px) — 1 колонка	Passed	Passed	Passed	
Дашборд (home.php): KPI-блоки та таблиця останніх замовлень відображаються коректно	Passed	Passed	Passed	
Товари з критично низьким залишком (≤ 3 од.) виводяться у відповідному блоці дашборду	Passed	Passed	Passed	

Рисунок 3.17 – Тест-кейси вебсайту «Aura»

Усі 23 тести отримали статус «Пройдено» за результатами тестування.

Захищені механізми, такі як готові заяви, htmlspecialchars() та bcrypt-хешування, підтвердили свою працездатність: спроби SQL-ін’єкції та XSS-

атаки не викликали небажаних результатів. Рольова модель доступу виконує технічне завдання, оскільки касир не може отримати доступ до розділів звітів і управління товарами як на рівні користувача інтерфейсу, так і на рівні серверної перевірки сесії.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було спроєктовано та розроблено вебдодаток «Aura» для управління асортиментом магазину одягу. Поставлені завдання виконано повністю: проведено аналіз предметної області, обрано технологічний стек, створено архітектуру системи, реалізовано програмний продукт та проведено функціональне тестування.

На етапі аналізу предметної області в першому розділі визначено, що саме потребує автоматизації для роботи магазину одягу. Торгівля текстилем має свої особливості: кожен продукт має різні розміри та кольори, асортимент постійно змінюється, а кількість працівників, які можуть отримати доступ до даних, різна. Електроні таблиці Excel не надійні для роботи, оскільки дані дублюються, залишки не точні, а помилки касира або менеджера можуть призвести до збитків для магазину. У результаті порівняльного аналізу існуючих рішень було виявлено, що більшість із них або надто складні для малого бізнесу, або не виконують усі необхідні функції в одному продукті. Це підтвердило, що було б доцільно створити власний додаток.

У другому розділі розглянуто проєктування системи. До написання першого рядка коду можна перевірити логіку системи, знайти потенційні проблеми та узгодити архітектурні рішення за допомогою UML-діаграм прецедентів, класів, послідовностей і розгортання. Реляційна база даних складається з семи взаємопов’язаних таблиць. Принципи нормалізації використовуються для того, щоб кожна сутність, будь то товар, категорія, покупець або продаж, зберігалася окремо і пов’язана через зовнішні ключі. Це гарантує, що дані залишаються безпечними, і запобігає надмірному дублюванню.

У третьому розділі реалізовано клієнтську та серверну частини застосунку. Публічна частина орієнтована на покупців: зручний каталог із фільтрацією, картки товарів із фотографіями і детальними характеристиками,

можливість оформити замовлення. Адміністративна панель призначена для працівників магазину і дозволяє виконувати весь спектр функцій керування, включаючи додавання та зміну товарів, управління категоріями, перегляд і обробка замовлень і аналітику продажів за допомогою графіків, створених за допомогою бібліотеки Chart.js. Трирівнева рольова модель, яка складається з ролей адміністратора, менеджера та касира, реалізується безпосередньо на рівні серверної логіки PHP. Кожна роль має детальний перелік дозволених дій, і жодна з них не може перевищити свої повноваження.

Окрему увагу приділено безпеці застосунку. Механізм підготовлених виразів використовується для захисту від SQL-ін’єкцій у кожному запиті до бази даних. Функція екранування захищає всі точки виведення користувацьких даних від XSS-атак. Усі паролі адміністраторів зберігаються за допомогою bcrypt-хешів. Ці рішення не є надмірною обережністю. Такий захист є життєво важливим для будь-якої системи, яка працює з комерційними даними.

Застосунок вирішує реальну проблему заміни табличного обліку на централізовану систему, забезпечуючи контроль доступу, поточні залишки та аналітику продажів. Продукт готовий до впровадження в малих або середніх магазинах одягу, а модульна структура коду дозволяє розширювати його функції без переробки всієї системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Хто такий веброзробник і чим він займається. IT Generation : освітній портал. URL: https://it-generation.com.ua/stati/khto-takyi-veb-rozrobnyk-i-chym-vin-zaimaietsia/ (дата звернення: 12.02.2026).

2. HTML, CSS, and JavaScript overview. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn (дата звернення: 12.02.2026).

3. About Node.js. Node.js Foundation. URL: https://nodejs.org/en/about (дата звернення: 12.02.2026).

4. SQL vs NoSQL Explained. MongoDB Inc. URL: https://www.mongodb.com/resources/compare/relational-vs-non-relational-databases (дата звернення: 12.02.2026).

5. Share of website traffic from mobile devices worldwide 2024. Statista. URL: https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices (дата звернення: 12.02.2026).

6. Що таке прогресивні вебдодатки: детальна відповідь. WEZOM : блог. URL: https://wezom.com.ua/ua/blog/pwa-prilozheniya-web-progressive-app (дата звернення: 12.02.2026).

7. OWASP Top Ten. OWASP Foundation. URL: https://owasp.org/www-project-top-ten (дата звернення: 12.02.2026).

8. PHP Manual. PHP. URL: https://www.php.net/manual/en/ (дата звернення: 16.02.2026).

9. MySQL. MySQL 8.0 Reference Manual. URL: https://dev.mysql.com/doc/refman/8.0/en/ (дата звернення: 12.02.2026).

10. Chart.js Documentation. Chart.js. URL: https://www.chartjs.org/docs/latest/ (дата звернення: 16.02.2026).

11. Password Storage Cheat Sheet. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html (дата звернення: 16.02.2026).

12. Zara : інтернет-магазин одягу та взуття. URL: https://www.zara.com (дата звернення : 20.02.2026).

13. INTERTOP : інтернет-магазин взуття та одягу. URL: https://intertop.ua (дата звернення : 20.02.2026).

14. VOVK : інтернет-магазин жіночого одягу. URL: https://vovk.com/ua/ (дата звернення : 20.02.2026).

15. Технічне завдання. Вікіпедія : вільна енциклопедія. URL: https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D1%96%D1%87%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F (дата звернення: 20.02.2026).

16. UML для бізнес-моделювання: для чого потрібні діаграми процесів. Evergreens : сайт. URL: https://evergreens.com.ua/ua/articles/uml-diagrams.html (дата звернення: 20.02.2026).

Декларація

академічної доброчесності

здобувача освіти ВСП «Економіко-правничий фаховий коледж ЗНУ»

Я, Заяц Катерина Сергіївна, здобувачка освіти 4 курсу, спеціальності/освітньо-професійної програми 121 Інженерія програмного забезпечення, групи К121-22, адреса електронної пошти katya.zayats0507@gmail.com

- підтверджую, що написана мною кваліфікаційна робота на тему «Розробка вебзастосунку для керування асортиментом магазину одягу» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата _____ Підпис _____ Катерина ЗАЯЦ

Дата _____ Підпис _____ Альона ХОДАКОВСЬКА