

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

СИВАК ДЕНИС ІВАНОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
д-р. техн. наук, професор
_____ Наталія ВЕСЕЛОВСЬКА
«_____» _____ 2026 р.

**ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ АДАПТИВНОГО ВЕБСАЙТУ
МАГАЗИНУ З ОПТИМІЗОВАНИМ UX/UI ДИЗАЙНОМ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

Олексій МОСКВІН, доцент кафедри
інформаційних технологій,

К. Т. Н.

Оцінка: _____ / _____ / _____

(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

АНОТАЦІЯ

Сивак Д. І. Проєктування та реалізація адаптивного вебсайту магазину з оптимізованим UX/UI дизайном. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця 2026.

У кваліфікаційній (бакалаврській) роботі досліджено та проаналізовано сучасні підходи до створення адаптивних вебзастосунків, принципи UX/UI дизайну та особливості розробки вебресурсів електронної комерції. За допомогою таких технологій як HTML, CSS, JavaScript, React, Node.js, Express.js та PostgreSQL було розроблено адаптивний вебсайт інтернет-магазину вина, основною метою якого є забезпечення зручної взаємодії користувача із системою та реалізація сучасного функціоналу електронної комерції.

Ключові слова: вебзастосунок, інтернет-магазин, UX/UI дизайн, адаптивний дизайн, Node.js, PostgreSQL, електронна комерція.

80 ст. 34 рис., 1 табл., 28 джерел.

ABSTRACT

Syvak D. I. Design and implementation of a responsive online store website with optimized UX/UI design. Specialty 122 “Computer Science”, educational program “Computer Science”. Vasyl’ Stus Donetsk National University, Vinnytsia, 2026.

In the qualification (bachelor’s) thesis, modern approaches to the development of responsive web applications, UX/UI design principles, and features of e-commerce web resource development were studied and analyzed. Using technologies such as HTML, CSS, JavaScript, React, Node.js, Express.js, and PostgreSQL, a responsive wine e-commerce website was developed. Its main purpose is to ensure convenient user interaction with the system and implement modern e-commerce functionality.

Keywords: web application, online store, UX/UI design, responsive design, Node.js, PostgreSQL, e-commerce.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ АДАПТИВНИХ ВЕБСАЙТІВ	7
1.1. Поняття та призначення вебсайтів електронної комерції.....	7
1.2. Основи адаптивного вебдизайну	9
1.3. UX/UI дизайн у сучасних вебзастосунках	11
1.4. Технології створення сучасних вебзастосунків	13
1.5. Використання баз даних PostgreSQL у вебсистемах	15
1.6. Аналіз аналогів інтернет-магазинів	17
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБСАЙТУ ІНТЕРНЕТ МАГАЗИНУ	24
2.1. Постановка задачі та визначення функціональних вимог	24
2.2. Проєктування структури вебсайту.....	33
2.3. Проєктування та реалізація бази даних PostgreSQL	45
2.4. Реалізація клієнтської частини вебсайту.....	55
2.5. Реалізація адаптивності та оптимізації UX/UI дизайну	60
РОЗДІЛ 3. ТЕСТУВАННЯ ТА АНАЛІЗ РОБОТИ ВЕБСАЙТУ	63
3.1. Тестування функціоналу вебсайту	63
3.2. Аналіз результатів тестування вебзастосунку	73
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	78

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій та цифровізації суспільства вебтехнології стали невід’ємною складовою бізнесу, комунікації та електронної комерції. Значна кількість компаній використовує вебзастосунки для реалізації товарів і послуг, забезпечення взаємодії з клієнтами та автоматизації бізнес-процесів. Особливо активно розвивається сфера інтернет-магазинів, яка дозволяє користувачам швидко та зручно здійснювати покупки через мережу Інтернет.

Одним із найважливіших факторів успішності сучасного вебресурсу є не лише його функціональність, але й якість користувацького досвіду. Саме тому особлива увага приділяється UX/UI дизайну, адаптивності інтерфейсу, швидкості роботи вебзастосунку та зручності навігації. У сучасних умовах користувачі використовують різні типи пристроїв для перегляду вебресурсів, зокрема смартфони, планшети, ноутбуки та персональні комп’ютери. Це зумовлює необхідність створення адаптивних вебсайтів, які забезпечують коректне відображення інтерфейсу на різних розмірах екранів.

Актуальність теми дипломної роботи полягає у необхідності створення сучасного адаптивного вебзастосунку інтернет-магазину із оптимізованим UX/UI дизайном, який забезпечує ефективну взаємодію користувача із системою, швидке оформлення замовлення та стабільну роботу вебресурсу. Сучасні інтернет-магазини повинні не лише надавати можливість придбання товарів, а й формувати позитивний користувацький досвід, що безпосередньо впливає на ефективність електронної комерції та рівень задоволеності користувачів.

Об’єктом дослідження є процес проектування та розробки адаптивного вебзастосунку електронної комерції.

Предметом дослідження є методи та засоби реалізації адаптивного вебсайту інтернет-магазину з використанням сучасних frontend та backend технологій, а також принципів UX/UI дизайну.

Метою дипломної роботи є проектування та реалізація адаптивного вебсайту інтернет-магазину вина з оптимізованим UX/UI дизайном, який

забезпечує зручну навігацію, ефективну взаємодію користувача із системою та підтримку основного функціоналу електронної комерції.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати сучасні підходи до розробки вебзастосунків;
- дослідити особливості UX/UI дизайну вебресурсів;
- виконати аналіз сучасних вебтехнологій та інструментів розробки;
- спроектувати структуру вебзастосунку;
- розробити frontend частину вебсайту;
- реалізувати backend частину та REST API;
- створити базу даних PostgreSQL;
- реалізувати систему кошика та оформлення замовлення;
- забезпечити адаптивність вебінтерфейсу;
- провести тестування вебзастосунку та аналіз результатів роботи системи.

У процесі виконання кваліфікаційної роботи використовувалися сучасні вебтехнології та інструменти розробки. Для реалізації frontend частини застосовувалися HTML, CSS, JavaScript та бібліотека React. Backend частина була реалізована за допомогою Node.js та Express.js. Для зберігання інформації використовувалася реляційна система керування базами даних PostgreSQL.

Практичне значення роботи полягає у створенні повноцінного адаптивного вебзастосунку інтернет-магазину вина, який може використовуватися для електронної комерції та демонструє сучасні підходи до реалізації UX/UI дизайну, frontend та backend розробки.

Структура дипломної роботи складається зі вступу, трьох розділів, висновків та списку використаних посилань.

У першому розділі розглянуто теоретичні аспекти веброзробки, сучасні технології створення вебзастосунків, принципи UX/UI дизайну та особливості адаптивної верстки.

У другому розділі описано процес проектування та реалізації вебзастосунку інтернет-магазину, розробку frontend та backend частини, створення бази даних, а також реалізацію функціоналу електронної комерції.

У третьому розділі проведено тестування вебзастосунку, аналіз адаптивності, перевірка UX/UI дизайну та оцінку продуктивності створеного вебресурсу.

Таким чином, результатом кваліфікаційної роботи є створення сучасного адаптивного вебсайту інтернет-магазину вина із оптимізованим UX/UI дизайном, який забезпечує зручну взаємодію користувача із системою та відповідає сучасним вимогам веброзробки.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ АДАПТИВНИХ ВЕБСАЙТІВ

1.1. Поняття та призначення вебсайтів електронної комерції

Електронна комерція є одним із найважливіших напрямів розвитку сучасних інформаційних технологій та бізнесу. Вона охоплює процес купівлі, продажу товарів і послуг через мережу Інтернет із використанням спеціалізованих вебсайтів та онлайн-платформ. Завдяки активному розвитку цифрових технологій електронна комерція стала невід’ємною частиною сучасної економіки, оскільки дозволяє підприємствам автоматизувати процеси продажу, розширювати аудиторію клієнтів та забезпечувати швидку взаємодію між продавцем і покупцем [1].

Основним інструментом електронної комерції є вебсайт інтернет-магазину. Такий вебресурс забезпечує користувачам можливість переглядати каталог товарів, оформлювати замовлення та виконувати оплату через мережу Інтернет. Для власників бізнесу вебсайт виступає ефективним засобом автоматизації продажів, маркетингу та комунікації з клієнтами [2].

Сучасні інтернет-магазини значно відрізняються від перших вебсайтів електронної комерції. Якщо раніше основною функцією сайту було лише відображення інформації про товари, то сьогодні вебмагазини являють собою складні програмні системи з інтегрованими базами даних, системами авторизації, онлайн-оплати, адаптивним дизайном та розширеним функціоналом взаємодії з користувачем. Особливу увагу приділяють швидкості роботи вебсайту, зручності інтерфейсу та адаптації під мобільні пристрої.

Однією з головних переваг вебсайтів електронної комерції є можливість забезпечення цілодобового доступу до товарів і послуг. На відміну від традиційних магазинів, онлайн-магазини працюють без обмежень у часі та дозволяють користувачам здійснювати покупки з будь-якого місця, де є доступ до

Інтернету. Це значно підвищує рівень комфорту для клієнтів та сприяє збільшенню кількості продажів.

Інтернет-магазини також забезпечують суттєве зниження витрат для підприємств. Власнику вебмагазину не потрібно орендувати великі торгові приміщення або утримувати значну кількість персоналу. Більшість процесів автоматизуються за допомогою програмного забезпечення, а обробка замовлень здійснюється через серверну частину вебзастосунку та базу даних.

Важливу роль у функціонуванні вебсайтів електронної комерції відіграють бази даних. Саме вони забезпечують зберігання інформації про товари, користувачів, замовлення та інші елементи системи. Для сучасних вебзастосунків часто використовуються реляційні системи керування базами даних, зокрема PostgreSQL, яка характеризується високою продуктивністю, надійністю та безпекою.

Ще одним важливим аспектом сучасних вебмагазинів є UX/UI дазайн. UX відповідає за зручність взаємодії користувача із системою, тоді як UI визначає зовнішній вигляд інтерфейсу. Від якості UX/UI дазайну залежить швидкість пошуку товарів, простота оформлення замовлення та загальне враження користувача від роботи із сайтом. Саме тому під час розробки сучасних вебсайтів велика увага приділяється створенню інтуїтивно зрозумілого та адаптивного інтерфейсу[3].

Адаптивний дизайн дозволяє вебсайту коректно відображатися на різних пристроях: персональних комп'ютерах, планшетах та смартфонах. Це особливо важливо в умовах зростання кількості мобільних користувачів. Використання адаптивної верстки забезпечує комфортну роботу із сайтом незалежно від розміру екрана пристрою[4].

Таким чином, вебсайти електронної комерції є важливим інструментом сучасного бізнесу, який забезпечує автоматизацію продажів, зручність для користувачів та ефективне управління інформацією. Розробка адаптивних вебмагазинів із використанням сучасних технологій програмування, UX/UI дизайну та систем керування базами даних дозволяє створювати функціональні

та конкурентоспроможні вебзастосунки, здатні задовольнити потреби сучасних користувачів.

1.2. Основи адаптивного вебдизайну

У сучасних умовах розвитку вебтехнологій адаптивний вебдизайн є одним із найважливіших елементів створення якісних вебзастосунків. З кожним роком кількість користувачів мобільних пристроїв стрімко зростає, тому вебсайти повинні коректно працювати на смартфонах, планшетах, ноутбуках та персональних комп'ютерах. Саме адаптивний дизайн забезпечує правильне відображення інтерфейсу на різних типах пристроїв незалежно від розміру екрана або роздільної здатності.

Адаптивний вебдизайн – це підхід до створення вебсайтів, при якому структура, елементи інтерфейсу та контент автоматично адаптивного дизайну є забезпечення зручності користування вебсайтом без необхідності створення окремих версій для мобільних пристроїв та комп'ютерів [5].

Одним із головних принципів адаптивного дизайну є використання гнучкої структури сторінки. Для цього застосовуються сучасні можливості CSS, зокрема Flexbox та CSS Grid. Вони дозволяють створювати динамічне розташування блоків, яке автоматично змінюється залежно від ширини екрана. Завдяки цьому елементи вебсайту можуть коректно переноситися, масштабуватися або змінювати своє розташування [6].

Важливу роль у реалізації адаптивного дизайну відіграють *media queries*. Це спеціальні CSS-правила, які дозволяють задавати різні стилі залежно від параметрів пристрою, наприклад ширини екрана. За допомогою *media queries* можна змінювати розміри тексту, розташування блоків, меню навігації та інші елементи інтерфейсу для забезпечення максимальної зручності користувача [7].

Однією з основних концепцій сучасного адаптивного дизайну є принцип *Mobile First*. Його суть полягає у тому, що розробка інтерфейсу спочатку виконується для мобільних пристроїв, а потім поступово адаптується для

більших екранів. Такий підхід дозволяє створювати більш оптимізовані та швидкі вебзастосунки, оскільки мобільна версія містить лише найнеобхідніші елементи інтерфейсу [8].

Для адаптивного вебдизайну важливим є також правильне використання графічного контенту. Зображення повинні автоматично масштабуватися та не виходити за межі контейнерів. Для цього використовуються відносні розміри, наприклад `width 100%`, що дозволяє зображенням адаптуватися до ширини екрана. Крім того, сучасні формати графіки та оптимізація розміру файлів дозволяють покращити швидкість завантаження сторінок.

Окрему увагу приділяють адаптації навігації вебсайту. На мобільних пристроях класичне горизонтальне меню часто замінюється так званим «бургер-меню», яке займає менше місця та забезпечує ергономіку інтерфейсу та підвищити комфорт користування вебресурсом [9].

Важливою перевагою адаптивного дизайну є покращення користувацького досвіду. Якщо вебсайт некоректно відображається на смартфоні або має незручний інтерфейс, користувачі швидко залишають ресурс. Адаптивність забезпечує комфортне перегортання сторінок, читання тексту, перегляд товарів та оформлення замовлення незалежно від типу пристрою.

Крім зручності для користувачів, адаптивний дизайн позитивно впливає на пошукову оптимізацію вебсайту. Пошукові системи, зокрема Google, надають перевагу адаптивним вебресурсам, що може покращити позиції сайту у результатах пошуку. Це особливо важливо для інтернет-магазинів, оскільки видимість сайту безпосередньо впливає на кількість потенційних клієнтів.

Під час створення сучасних вебзастосунків адаптивний дизайн реалізується за допомогою HTML, CSS та JavaScript. Для спрощення процесу розробки також використовуються сучасні бібліотеки та фреймворки, React [10].

Отже, адаптивний вебдизайн є необхідною складовою сучасних вебсайтів та вебзастосунків. Його використання дозволяє забезпечити коректну роботу сайту на різних пристроях, підвищити зручність користування, покращити швидкість роботи вебресурсу та забезпечити позитивний користувацький досвід.

Саме тому адаптивність є одним із ключових критеріїв під час розробки сучасних інтернет-магазинів.

1.3. UX/UI дизайн у сучасних вебзастосунках

У процесі розробки сучасних вебзастосунків важливу роль відіграє UX/UI дизайн, оскільки саме він визначає зручність використання вебсайту та формує загальне враження користувача від роботи із системою. Для інтернет-магазинів якісний UX/UI дизайн є особливо важливим, адже від нього залежить швидкість пошуку товарів, зручність оформлення замовлення та ефективність взаємодії користувача із вебресурсом.

UX (User Experience) – це користувацький досвід, який описує загальне сприйняття користувачем роботи вебсайту або застосунку. UX включає зручність навігації, швидкість виконання дій, логічність структури сторінок, доступність функцій та комфорт взаємодії із системою. Основною метою UX-дизайну є створення такого вебресурсу, який буде максимально зрозумілим та зручним для користувача [11].

UI (User Interface) – це користувацький інтерфейс, тобто візуальна складова вебзастосунку. UI охоплює оформлення сторінок, кольорові схеми, шрифти, кнопки, анімації, розташування елементів та інші візуальні компоненти. Завдання UI-дизайну полягає у створенні сучасного, естетичного та інтуїтивно зрозумілого інтерфейсу [12].

Одним із головних принципів UX/UI дизайну є простота використання. Користувач повинен швидко знаходити необхідну інформацію та без труднощів виконувати потрібні дії. Для цього вебсайт має містити зрозумілу структуру сторінок, логічне меню навігації та мінімальну кількість зайвих елементів, які можуть відволікати увагу.

Особливу увагу у UX/UI дизайні приділяють навігації вебсайту. Для інтернет-магазинів важливо, щоб користувач міг швидко перейти до каталогу товарів, сторінки кошика, контактної інформації або оформлення замовлення.

Наявність зрозумілого меню та пошукової системи значно покращує взаємодію користувача із сайтом та зменшує час пошуку необхідного товару [13].

Важливим елементом UI-дизайну є кольорова гама вебсайту. Кольори впливають на емоційне сприйняття користувача та формують стиль вебресурсу. Для інтернет-магазинів вина часто використовують темні, насичені або теплі кольори, які асоціюються з елегантністю, преміальністю та атмосферою винної культури. Правильно підібрана кольорова схема дозволяє створити привабливий та сучасний інтерфейс.

Не менш важливою складовою UI є типографіка. Шрифти повинні бути читабельними, сучасними та гармонійно поєднуватися із загальним дизайном сайту. Надмірна кількість різних шрифтів або неправильний розмір тексту можуть погіршити сприйняття інформації користувачем.

Для сучасних вебзастосунків характерним є використання інтерактивних елементів інтерфейсу. До них належать кнопки, анімації, ефекти покращують візуальне сприйняття та роблять взаємодію користувача із системою більш комфортною [14].

Важливою складовою UX/UI дизайну є також адаптивність інтерфейсу. Сайт повинен бути однаково зручним як на персональному комп'ютері, так і на мобільних пристроях. Для цього використовуються адаптивні компоненти, гнучкі сітки та спеціальні стилі для різних розмірів екранів.

Під час створення UX/UI дизайну велика увага приділяється швидкості роботи вебсайту. Якщо сторінки завантажуються повільно або інтерфейс працює нестабільно, користувачі можуть залишити вебресурс. Саме тому сучасні вебзастосунки оптимізуються шляхом зменшення розміру зображень, використання кешування та оптимізації коду.

У сучасній веброзробці UX/UI дизайну реалізується за допомогою HTML, CSS, JavaScript та різноманітних бібліотек і фреймворків. Одним із популярних інструментів є React, який дозволяє створювати компонентний інтерфейс та забезпечує швидке оновлення елементів сторінки без повного перезавантаження сайту [15].

Таким чином, UX/UI дизайну є важливою складовою сучасних вебзастосунків та безпосередньо впливає на ефективність роботи інтернет-магазину. Використання сучасних принципів UX/UI дозволяє створити зручний, привабливий та функціональний вебресурс, який забезпечує позитивний користувацький досвід та підвищує рівень взаємодії користувачів із системою.

1.4. Технології створення сучасних вебзастосунків

Сучасні вебзастосунки є складними програмними системами, які забезпечують взаємодію користувача із сервером через веббраузер. Для створення функціональних, швидких та адаптивних вебресурсів використовуються різноманітні технології та інструменти, що дозволяють реалізувати як клієнтську, так і серверну частину системи. Розробка сучасного вебзастосунку передбачає використання мов розмітки, стилізації, програмування, серверних платформ та систем керування базами даних.

Основою будь-якого вебсайту є HTML (HyperText Markup Language). HTML використовується для створення структури вебсторінок та визначення їх основних елементів: заголовків, тексту, таблиць, зображень, кнопок, форм та інших компонентів інтерфейсу. Саме HTML формує каркас вебсайту, на який у подальшому накладаються стилі та функціональні можливості [16].

Для оформлення зовнішнього вигляду сторінок використовується CSS (Cascading Style Sheets). CSS дозволяє задавати кольори, шрифти, розміри елементів, анімації, розташування блоків та адаптивність інтерфейсу. За допомогою CSS реалізується сучасний дизайн вебсайту, який забезпечує привабливий зовнішній вигляд та комфортну взаємодію користувача із системою [17].

Однією з найважливіших технологій сучасної веброзробки є JavaScript. Дана мова програмування забезпечує інтерактивність вебсторінок та дозволяє створювати динамічні елементи інтерфейсу. За допомогою JavaScript реалізується випадаючі меню, анімації, перевірка введених даних, оновлення

інформації без перезавантаження сторінки та інші функціональні можливості вебзастосунку [18].

У сучасних вебпроектах широко використовуються JavaScript-фреймворки та бібліотеки. Однією з найпопулярніших бібліотек є React. React дозволяє створювати компонентний інтерфейс, де кожна частина вебсторінки представлена у вигляді окремого компонента. Це значно спрощує процес розробки, повторного використання коду та підтримки вебзастосунку.

Перевагою React є можливість швидкого оновлення окремих елементів сторінки без її повного перезавантаження. Це забезпечує високу швидкість роботи вебзастосунку та покращує користувацький досвід. Крім того, React підтримує створення адаптивних інтерфейсів та добре підходить для розробки сучасних інтернет-магазинів [19].

Для реалізації серверної частини вебзастосунків часто використовується Node.js. Node.js є середовищем виконання JavaScript-коду на сервері та дозволяє створювати швидкі та масштабовані серверні застосунки. Однією з основних переваг Node.js є використання однієї мови програмування як для frontend, так і для backend частини вебсистеми [20].

Разом із Node.js часто застосовується фреймворк Express.js. Він спрощує створення серверної логіки, маршрутизації та API. За допомогою Express.js можна реалізувати обробку HTTP-запитів, авторизацію користувачів, взаємодію із базою даних та інші серверні функції [21].

Для обміну даними між клієнтською та серверною частинами використовується REST API. API (Application Programming Interface) дозволяє frontend частині отримувати або передавати дані на сервер. Наприклад, при відкритті каталогу товарів frontend надсилає запит до сервера, а backend повертає інформацію про товари з бази даних у форматі JSON.

Важливою складовою сучасних вебзастосунків є система керування базами даних. Для зберігання інформації про користувачів, товари, замовлення та інші дані часто використовуються реляційні бази даних. Однією з найбільш

популярних систем є PostgreSQL. Вона забезпечує високу продуктивність, надійність та підтримку складних SQL-запитів.

Для взаємодії сервера з базою даних використовуються спеціальні бібліотеки та драйвери. Вони дозволяють виконувати SQL-запити, додавати нові записи, змінювати інформацію та отримувати дані для відображення на вебсайті. У вебмагазинах база даних відіграє ключову роль, оскільки саме в ній зберігається основна інформація про роботу системи.

Таким чином, створення сучасних вебзастосунків базується на комплексному використанні frontend та backend технологій. HTML, CSS та JavaScript забезпечують побудову інтерфейсу користувача, React дозволяє створювати динамічні компоненти, Node.js та Express.js реалізують серверну логіку, а PostgreSQL забезпечує надійне зберігання даних. Поєднання цих технологій дозволяє створювати сучасні адаптивні вебсистеми з високим рівнем функціональності та зручності використання.

1.5. Використання баз даних PostgreSQL у вебсистемах

У сучасних вебзастосунках бази даних є однією з найважливіших складових програмної системи. Саме вони забезпечують зберігання, обробку та керування інформацією, необхідною для функціонування вебресурсу. Для інтернет-магазинів база даних використовується для зберігання інформації про товари, користувачів, замовлення, категорії продукції, кошик покупця, та інші елементи системи. Без використання бази даних реалізація повноцінного вебмагазину є практично неможливою.

Однією з найбільш популярних систем керування базами даних є PostgreSQL. PostgreSQL це об'єктно-реляційна система керування базами даних з відкритим вихідним кодом, яка характеризується високою продуктивністю, стабільністю та надійністю. Вона підтримує складні SQL- запити, механізми безпеки, багатокористувацький режим роботи та ефективну обробку великих обсягів інформації [22].

Основною перевагою PostgreSQL є використання реляційної моделі даних. У такій моделі інформація зберігається у вигляді таблиць, які можуть бути пов'язані між собою за допомогою спеціальних ключів. Це дозволяє уникати дублювання інформації, забезпечувати цілісність даних та спростувати процес їх обробки.

Кожна таблиця у базі даних складається зі стовпців та рядків. Стовпці визначають тип інформації, яка буде зберігатися, а рядки містять конкретні записи. Наприклад, у вебмагазині може існувати таблиця товарів, де зберігаються назва товару, ціна, опис, категорія, кількість на складі та зображення продукції. Окремо можуть існувати таблиці користувачів та замовлень.

Для забезпечення зв'язків між таблицями використовуються primary key та foreign key. Primary key є унікальним ідентифікатором запису в таблиці, тоді як foreign key використовується для встановлення зв'язку між таблицями. Наприклад, таблиця замовлень може містити зовнішній ключ, який посилається на таблицю користувачів, що дозволяє визначити, який саме користувач оформив замовлення [23].

Важливу роль у роботі PostgreSQL відіграє мова SQL (Structured Query Language). SQL використовується для створення таблиць, додавання нових записів, зміни інформації, видалення даних та виконання пошуку. За допомогою SQL-запитів можна швидко отримувати необхідну інформацію з бази даних та передавати її до вебзастосунку [24].

У сучасних вебсистемах PostgreSQL часто використовується разом із серверними технологіями Node.js та Express.js. Серверна частина вебзастосунку надсилає SQL-запити до бази даних, отримує необхідну інформацію та передає її фронтенд частині для відображення користувачу. Такий підхід дозволяє створювати динамічні вебсайти з автоматичним оновленням інформації.

Для інтернет-магазину база даних виконує велику кількість важливих функцій. Вона забезпечує зберігання каталогу товарів, даних користувачів, інформації про замовлення, історії покупок та контактної інформації. Крім того,

база даних дозволяє реалізувати систему авторизації користувачів, кошик покупця та функціонал оформлення замовлення.

Однією з переваг PostgreSQL є високий рівень безпеки. Система підтримує механізми авторизації, розмежування прав доступу та захисту даних. Це особливо важливо для вебзастосунків електронної комерції, де зберігається конфіденційна інформація користувачів та дані про замовлення.

Ще однією важливою особливістю PostgreSQL є підтримка масштабованості та стабільності роботи. Система здатна ефективно працювати навіть при великій кількості користувачів та значному навантаженні на сервер. Саме тому PostgreSQL широко використовується як у невеликих вебпроектах, так і у великих комерційних системах.

Для роботи з PostgreSQL часто використовуються спеціальні графічні інструменти, наприклад pgAdmin. Вони дозволяють створювати таблиці, виконувати SQL-запити, переглядати структуру бази даних та керувати інформацією через зручний інтерфейс. Використання таких інструментів значно спрощує процес розробки та адміністрування бази даних [25].

Таким чином, PostgreSQL є потужною та надійною системою керування базами даних, яка широко використовується у сучасних вебзастосунках. Використання PostgreSQL у вебмагазинах дозволяє забезпечити ефективне зберігання та обробку інформації, підтримувати стабільну роботу системи та реалізувати складний функціонал електронної комерції. Завдяки своїм можливостям PostgreSQL є оптимальним рішенням для створення сучасних адаптивних вебсайтів інтернет-магазинів.

1.6. Аналіз аналогів інтернет-магазинів

Аналіз аналогів є обов'язковим етапом проєктування будь-якого вебзастосунку, оскільки дозволяє вивчити існуючі рішення у відповідній предметній галузі, виявити їхні переваги та недоліки, запозичити ефективні UX-патерни та уникнути типових помилок. Результати аналізу безпосередньо

впливають на формування вимог до власного проєкту та обґрунтовують прийняті проєктні рішення.

Для аналізу було обрано три інтернет-магазини вина, що представляють різні підходи до проєктування інтерфейсу, функціонального наповнення та технічної реалізації. Кожен аналог оцінювався за такими критеріями:

- дизайн та візуальна привабливість інтерфейсу;
- зручність навігації та пошуку товарів;
- функціональність каталогу та карток товарів;
- реалізація кошика та процесу оформлення замовлення;
- наявність системи авторизації та особистого кабінету;
- адаптивність на мобільних пристроях;
- швидкість завантаження сторінок;
- додаткові функціональні можливості.

Winetime є одним із найбільших українських інтернет-магазинів вина із широким асортиментом продукції та розвинутою інфраструктурою доставки. Сайт функціонує з 2010 року та має значну аудиторію постійних покупців [26].

UX/UI дизайн. Інтерфейс Winetime витримано у класичному стилі з переважанням темно-червоних та бордових кольорів, що відповідає тематиці магазину. Головна сторінка містить великий банер із акційними пропозиціями, горизонтальне меню категорій та сітку рекомендованих товарів. Дизайн є функціональним, але дещо застарілим – відсутні сучасні анімації та мікровзаємодії, що знижує загальне враження від використання.

Каталог та пошук. Каталог організований за типом вина (червоне, біле, рожеве, ігристе, десертне), країною походження та ціновим діапазоном. Реалізована система фільтрів із можливістю одночасного застосування кількох параметрів. Пошук за ключовим словом функціонує, але не забезпечує підказок у режимі реального часу – результати відображаються лише після натискання кнопки пошуку, що є суттєвим UX-недоліком.

Кошик та оформлення замовлення. Процес оформлення замовлення складається з кількох кроків та вимагає обов'язкової реєстрації або входу перед оформленням – гостьове замовлення не підтримується. Це є значним бар'єром для нових покупців, оскільки частина з них залишить сайт, не бажаючи реєструватися.

Адаптивність. Мобільна версія сайту є базово адаптивною, проте деякі елементи інтерфейсу (зокрема, розширена панель фільтрів) відображаються некоректно на малих екранах і вимагають горизонтальної прокрутки.

Переваги: великий асортимент товарів, детальні характеристики вин, розвинена система фільтрації, наявність відгуків покупців.

Недоліки: застарілий дизайн, відсутність пошуку в режимі реального часу, обов'язкова реєстрація, часткові проблеми з мобільною адаптивністю, низька швидкість завантаження окремих сторінок.

Аромат вина є спеціалізованим інтернет-магазином з акцентом на українських та грузинських винах, що є нішевим рішенням для відповідної аудиторії.

UX/UI дизайн. Інтерфейс сайту є найбільш простим серед розглянутих українських аналогів. Дизайн функціональний, але не має чіткої концепції – суміш різних стилів та кольорів створює відчуття неузгодженості. Головна сторінка перевантажена текстовим контентом, що ускладнює швидке орієнтування.

Каталог та пошук. Каталог організований за країнами та виробниками, що зручно для аудиторії, яка вже знає, що шукає, але незручно для нових покупців, які обирають за типом вина. Пошук базовий, без автодоповнення.

Функціональність. Сайт надає детальну текстову інформацію про кожне вино, включаючи рекомендації щодо поєднання з їжею та умови зберігання. Це є перевагою для досвідченої аудиторії, але може перевантажувати новачків.

Адаптивність. Мобільна версія має суттєві проблеми: деякі блоки виходять за межі екрана, розміри елементів керування занадто малі для комфортного використання пальцем.

Переваги: детальні описи товарів; акцент на локальному виробнику; унікальний асортимент.

Недоліки: застарілий та неузгоджений дизайн, суттєві проблеми з мобільною адаптивністю, відсутність сучасних UX-рішень, повільне завантаження.

Vivino є найбільшою у світі онлайн-платформою для вина з аудиторією понад 50 мільйонів користувачів. Сервіс поєднує функції інтернет-магазину та соціальної мережі для любителів вина [27].

UX/UI дизайн. Інтерфейс Vivino є одним із найбільш продуманих у галузі. Платформа використовує чистий мінімалістичний дизайн з якісними зображеннями та зрозумілою ієрархією елементів. Кольорова схема – переважно білий фон із виноградно-фіолетовими акцентами - формує чіткий і впізнаваний бренд.

Каталог та пошук. Vivino реалізує один із найкращих у галузі алгоритмів пошуку та рекомендацій. Пошук підтримує розпізнавання етикеток вина через камеру смартфона - революційна функція, що є одним із ключових конкурентних переваг платформи. Фільтрація за рейтингом, ціновим діапазоном, країною та сортом винограду реалізована через інтуїтивні повзунки та чекбокси.

Соціальні функції. Vivino інтегрує елементи соціальної мережі: користувачі можуть залишати відгуки та оцінки для кожного вина, стежити за іншими користувачами та отримувати персоналізовані рекомендації на основі своїх попередніх оцінок. Алгоритм рекомендацій є одним із найсильніших конкурентних переваг платформи.

Мобільний застосунок. Vivino має окремий мобільний застосунок (IOS та Android) із ще кращим UX, ніж вебверсія. Застосунок дозволяє сканувати етикетки вин у ресторані та миттєво отримувати рейтинг та відгуки.

Адаптивність. Вебверсія повністю адаптивна, зручна у використанні на всіх типах пристроїв.

Переваги: передовий UX/UI дизайн, унікальна функція розпізнавання етикеток, соціальні функції та персоналізовані рекомендації, висока швидкість завантаження, відмінна мобільна адаптивність.

Недоліки: недоступність для покупки в Україні, складна екосистема може бути надмірною для простого інтернет-магазину, обов'язкова реєстрація для більшості функцій.

В таблиці 1.1 наведено зведену порівняльну таблицю за ключовими критеріями, складено на основі проведеного аналізу.

Таблиця 1.1 Порівняльний аналіз аналогів

Критерій	Winetime	Аромат вина	Vivino
Дизайн (1-5)	3	2	5
Пошук у реальному часі	Ні	Ні	Так
Фільтрація	Розвинена	Базова	Розвинена
Порівняння товарів	Ні	Ні	Ні
Кошик	Ні	Ні	Ні
Гостьове замовлення	Ні	Так	Ні
Особистий кабінет	Так	Базовий	Так
Адаптивність	Часткова	Слабка	Повна
Швидкість	~55	~48	~88
Відгуки та рейтинги	Так	Ні	Так

За результатами проведеного порівняльного аналізу трьох магазинів вина – сформовано перелік ключових висновків, які безпосередньо вплинули на проєктні рішення власного вебзастосунок.

Перш за все, аналіз виявив суттєву різницю у рівні UX/UI дизайну між розглянутими платформами. Vivino демонструє еталонний підхід до проєктування інтерфейсу – мінімалістична кольорова схема, чітка візуальна ієрархія та якісні зображення формують відчуття преміальності та сучасності. Winetime та Аромат вина, натомість, використовують застарілі дизайнерські рішення, що негативно впливає на загальне сприйняття. Це підтверджує

необхідність застосування сучасного мінімалістичного дизайну у власному проєкті з продуманою кольоровою гамою, що відповідає тематиці магазину вина.

По-друге, жоден із трьох розглянутих аналогів не реалізує повноцінну функцію порівняння товарів. Для Winetime та Аромат вина це є очевидним функціональним недоліком, оскільки покупець вина нерідко обирає між кількома схожими позиціями та потребує зручного інструменту для їх зіставлення за характеристиками. Vivino частково компенсує цю відсутність через систему рейтингів та відгуків, проте повноцінного порівняння пліч-о-пліч не пропонує. Включення функції порівняння товарів до власного вебзастосунку є обґрунтованим рішенням, що заповнює виявлену прогалину та формує конкурентну перевагу.

По-третє, аналіз пошукового функціоналу виявив значні відмінності між платформами. Winetime та Аромат вина реалізують лише базовий пошук без підказок у режимі реального часу, що суттєво уповільнює процес знаходження потрібного товару. Vivino, навпаки, пропонує розвинений пошук із миттєвими результатами та унікальну функцію розпізнавання етикеток через камеру. Реалізація пошуку з динамічною фільтрацією результатів у режимі реального часу є обов'язковим елементом власного проєкту відповідно до сучасних стандартів UX.

По-четверте, питання обов'язковості реєстрації є важливим чинником конверсії. Winetime вимагає обов'язкової авторизації перед оформленням замовлення, що є суттєвим бар'єром для нових покупців. Аромат вина допускає гостьове замовлення, але не надає жодних переваг зареєстрованим користувачам. Vivino робить реєстрацію фактично обов'язковою для доступу до більшості функцій платформи. У власному проєкті реалізовано збалансований підхід: реєстрація надає реальні переваги – збережені дані профілю, автоматичне заповнення форми замовлення, доступ до історії покупок, але не є обов'язковою умовою для здійснення покупки.

По-п'яте, мобільна адаптивність є одним із найбільш критичних недоліків українських аналогів. Аромат вина має серйозні проблеми з відображенням на

мобільних пристроях, Winetime демонструє лише часткову адаптивність із некоректним відображенням окремих елементів. Vivino, будучи міжнародною платформою, реалізує повноцінну адаптивність та має окремий мобільний застосунок. Повна адаптивність інтерфейсу на всіх типах пристроїв є обов'язковою вимогою до власного проєкту, що підтверджується як результатами аналізу, так і загальними тенденціями ринку.

Таким чином, власний вебзастосунок проєктується з урахуванням виявлених недоліків кожного з аналогів та запозиченням їхніх сильних сторін. Від Vivino – сучасний мінімалістичний дизайн, пошук у режимі реального часу та детальні характеристики товарів. Від Winetime – розвинена система особистого кабінету з історією замовлень. Від Аромат вина – можливість гостьового замовлення без обов'язкової реєстрації. Виявлена у всіх трьох аналогів відсутність функції порівняння товарів та недостатній рівень мобільної адаптивності у вітчизняних платформах стали ключовими орієнтирами для формування конкурентних переваг власного вебзастосунку.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБСАЙТУ ІНТЕРНЕТ МАГАЗИНУ

2.1. Постановка задачі та визначення функціональних вимог

У сучасних умовах стрімкого розвитку інформаційних технологій та цифрової економіки вебзастосунки інтернет-магазинів перетворилися на один із найефективніших та найпоширеніших інструментів ведення комерційної діяльності. Зростання частки онлайн-продажів у загальному обсязі роздрібної торгівлі свідчить про те, що споживачі все частіше надають перевагу дистанційному придбанню товарів через вебресурси. Це зумовлює необхідність створення сучасних, адаптивних та функціонально насичених вебсайтів із високим рівнем зручності використання.

Ринок алкогольних напоїв, зокрема вин, є специфічним сегментом електронної комерції, де особливу роль відіграє презентація товару, якість контенту та зручність оформлення замовлення. Покупець, який обирає вино онлайн, потребує детального опису продукції, якісних фотографій, зрозумілої навігації та простого процесу оформлення покупки. Саме тому під час розробки вебзастосунку особлива увага приділяється UX/UI дизайну, адаптивності інтерфейсу, швидкості роботи та стабільності системи.

У процесі постановки задачі було визначено, що створений вебзастосунок повинен виконувати не лише функцію продажу товарів, а й забезпечувати сучасний користувацький досвід. Інтерфейс вебсайту повинен бути інтуїтивно зрозумілим, адаптивним та візуально привабливим. Крім того, система повинна забезпечувати стабільну взаємодію між frontend частиною, backend сервером та базою даних PostgreSQL.

У процесі постановки задачі було визначено такі основні цілі створення вебзастосунку:

- реалізація сучасного адаптивного інтерфейсу для всіх типів пристроїв;

- створення зручної та логічно структурованої системи навігації;
- реалізація каталогу товарів із розширеними фільтрами та пошуком;
- створення повноцінної системи кошика;
- реалізація функціоналу оформлення замовлення з вибором способу доставки та оплати;
- інтеграція реляційної бази даних для зберігання та обробки даних;
- забезпечення взаємодії frontend та backend частини через REST API;
- оптимізація UX/UI дизайну відповідно до сучасних стандартів та патернів;
- забезпечення швидкої роботи вебресурсу та оптимізація продуктивності.

Одним із головних завдань є створення адаптивного вебінтерфейсу, який коректно відображається та функціонуватиме на таких типах пристроїв:

- персональних комп'ютерах (роздільна здатність від 1280px і вище);
- ноутбуках (роздільна здатність від 1024px-1280px і вище);
- планшетах (роздільна здатність від 768px-1024px і вище);
- смартфонах (роздільна здатність до 768px і вище);

Актуальність адаптивного дизайну пояснюється тим, що значна частка сучасних користувачів переглядає товари та здійснює покупки саме через мобільні пристрої. Пошуковий алгоритм Google також надає перевагу мобільно-адаптованим сайтам у результатах пошуку (принцип Mobile First Indexing), що безпосередньо впливає на відвідуваність ресурсу. Саме тому вебресурс повинен автоматично адаптувати структуру сторінок, розмір шрифтів, відступи та розташування елементів залежно від розміру екрана користувача. Для реалізації адаптивності використовується підхід Responsive Web Design на основі медіа-запитів CSS та гнучкої сіткової системи.

Важливою складовою сучасного інтернет-магазину є система реєстрації та авторизації користувачів. Наявність особистого облікового запису суттєво

покращує користувацький досвід, дозволяє персоналізувати взаємодію із платформою та спрощує повторне оформлення замовлень.

Система реєстрації та авторизації повинна забезпечувати такі можливості:

- реєстрацію нового користувача із заповненням обов'язкових полів (ім'я, прізвище, email-адреса, пароль);
- перевірку коректності Валідацію введених даних під час реєстрації формату email, відповідності паролю встановленим вимогам надійності, а також унікальності електронної адреси у системі;
- авторизацію зареєстрованого користувача за допомогою email-адреси та пароля;
- безпечне зберігання паролів у базі даних у хешованому вигляді з використанням алгоритму bcrypt, що унеможливорює їх зчитування у відкритому вигляді навіть при компрометації бази даних [28];
- підтримку сесій або токенів авторизації для збереження стану входу між сеансами;
- можливість виходу з облікового запису із завершенням активної сесії;
- захист особистих сторінок та дій (перегляд історії замовлень, збережені дані профілю) від неавторизованого доступу.

Для авторизованих користувачів передбачено розширений функціонал особистого кабінету, який включає:

- перегляд персональних даних із можливістю їх редагування (ім'я, прізвище, контактний номер, адреса доставки);
- перегляд історії оформлених замовлень із відображенням статусу кожного з них;
- можливість повторного замовлення раніше придбаних товарів.

Інформація про зареєстрованих користувачів зберігається в окремій таблиці бази даних PostgreSQL та пов'язується із таблицею замовлень через

зовнішній ключ, що дозволяє ефективно відстежувати всю активність конкретного покупця.

Однією з ключових функцій системи є каталог товарів, який є центральним елементом будь-якого інтернет-магазину. Він повинен забезпечувати такі можливості:

- перегляд усього асортименту продукції;
- Відображення якісних фотографій товарів із можливістю детального перегляду;
- виведення вартості кожної позиції із підтримкою різних форматів відображення ціни;
- короткий інформативний опис товару (назва, країна походження, рік врожаю, тип вина та рейтинг);
- можливість додавання продукції до кошика безпосередньо з картки товару;
- відображення детальної сторінки товару з повним описом та характеристиками.

Каталог формується на основі даних, що зберігаються у базі даних PostgreSQL, та передається на клієнтську частину через REST API. Завдяки цьому підходу оновлення асортименту не потребує втручання у програмний код – достатньо змінити відповідні записи у базі даних.

Для суттєвого покращення користувацького досвіду у вебзастосунку реалізується система пошуку товарів. Пошук є особливо важливим при великій кількості позицій у каталозі, коли ручне прокручування сторінок значно ускладнює процес вибору товару.

Система пошуку повинна надавати такі можливості:

- пошук товарів за назвою або ключовими словами у режимі реального часу;
- миттєве оновлення результатів без перезавантаження сторінки;

- відображення підказок або результатів у спадному списку під час введення запиту;
- коректна обробка порожніх запитів та відсутності результатів із відповідним повідомленням для користувача.

Важливою функціональною складовою вебресурсу є кошик покупок — елемент, через який безпосередньо формується намір покупки. Система кошика повинна підтримувати:

- додавання товарів із каталогу одним натисканням;
- зміну кількості обраної продукції з автоматичним перерахунком вартості;
- видалення окремих товарів або очищення всього кошика;
- автоматичний підрахунок загальної вартості замовлення з урахуванням кількості кожної позиції;
- збереження стану кошика між переходами між сторінками під час поточної сесії;
- відображення кількості доданих позицій у навігаційній панелі для зручності користувача.

Реалізація кошика на клієнтській стороні здійснюється із використанням механізмів управління станом, що забезпечує миттєве оновлення інтерфейсу без затримок та перезавантаження сторінки.

Наступним важливим функціональним модулем є система оформлення замовлення. Від зручності цього процесу безпосередньо залежить коефіцієнт конверсії — частка відвідувачів, які фактично здійснюють покупку. Під час оформлення покупки користувач повинен мати можливість:

- вводити особисту інформацію (ім'я, прізвище, контактний номер телефону,
- email-адресу);
- вказувати адресу доставки або відділення поштового оператора;
- обирати зручний спосіб доставки серед доступних варіантів;

- переглядати підсумок замовлення перед його підтвердженням;
- підтверджувати замовлення та отримувати зворотний зв'язок про успішне його оформлення;
- Отримувати автоматичне повідомлення в особистому кабінеті з деталями замовлення після його успішного оформлення.

Для забезпечення гнучкості вебзастосунку передбачено підтримку кількох способів доставки:

- Доставка через оператора «Нова пошта» із зазначенням номера відділення або адресної доставки;
- Самовивіз із зазначеного пункту видачі.

Вибір способу доставки динамічно змінює поля форми, які необхідно заповнити користувачу, що спрощує процес оформлення та зменшує кількість кроків.

Ще однією важливою та конкурентною перевагою вебзастосунку є функціонал порівняння товарів. Ця можливість особливо актуальна для сегменту вин, де покупець нерідко обирає між кількома схожими позиціями та потребує зручного інструменту для їх зіставлення за ключовими характеристиками.

Сторінка порівняння товарів повинна забезпечувати такі можливості:

- додавання товарів до списку порівняння безпосередньо з картки у каталозі за допомогою відповідної кнопки або іконки;
- відображення кількості доданих до порівняння позицій у навігаційній панелі, аналогічно до лічильника кошика;
- перехід на окрему сторінку порівняння, де обрані товари відображаються у вигляді таблиці;
- наочне паралельне зіставлення характеристик товарів за такими параметрами: назва, ціна, країна походження, рік врожаю, тиш вина (червоне, біле, рожеве, ігристе), міцність, виробник, об'єм пляшки;
- можливість видалення окремої позиції зі списку порівняння безпосередньо на сторінці порівняння без переходу до каталогу;

- можливість додавання порівнюваного товару до кошика прямо зі сторінки порівняння, що скорочує кількість дій для оформлення покупки;
- обмеження максимальної кількості товарів, які можна порівнювати одночасно (рекомендовано не більше 3-4 позицій), із відповідним повідомленням при спробі перевищити ліміт.

З технічної точки зору список товарів для порівняння зберігається у стані клієнтської частини застосунку та підтримується між переходами та сторінками протягом поточної сесії. При переході на сторінку порівняння клієнтська частина надсилає запит до серверного API для отримання повних характеристик обраних товарів за їхніми ідентифікаторами, після чого формує таблицю порівняння на основі отриманих даних.

Сторінка порівняння також повинна коректно обробляти крайні випадки:

- відображення відповідного повідомлення, якщо користувач перейшов на сторінку порівняння, не додавши жодного товару;
- адаптивне відображення таблиці порівняння на мобільних пристроях із горизонтальною прокруткою або перестроєнням макету.

Функціонал порівняння товарів є значущим інструментом підтримки прийняття рішення про покупку, що підвищує рівень залученості користувача та позитивно впливає на конверсію інтернет-магазину.

У процесі визначення функціональних вимог особлива увага приділяється інтеграції backend частини та бази даних. Серверна частина вебзастосунку є посередником між клієнтом та базою даних і повинна забезпечувати:

- обробку HTTP-запитів від клієнтської частини (GET, POST, PUT, DELETE);
- реалізацію маршрутизації через REST API для доступу до ресурсів системи;
- передачу інформації у форматі JSON між frontend та базою даних;
- отримання та збереження інформації про товари;

- обробку та збереження замовлень у базі даних;
- валідацію вхідних даних перед записом у базу;
- обробку помилок та повернення відповідних HTTP-статус-кодів.

Для реалізації серверної частини обрано технологічний стек Node.js у поєднанні з фреймворком Express.js. Використання цих технологій дозволяє створити стабільний, асинхронний серверний застосунок із підтримкою REST API та ефективною обробкою клієнтських запитів. Node.js забезпечує неблокуючу модель введення-виведення, що є особливо важливим при обробці одночасних запитів від кількох користувачів.

Важливу роль у роботі вебзастосунку відіграє реляційна база даних PostgreSQL, яка забезпечує надійне зберігання та ефективну обробку всіх даних системи. База даних повинна:

- зберігати повну інформацію про товари (назва, опис, ціна, зображення, категорія, характеристики);
- зберігання інформацію про замовлення та їхній статус;
- зберігати дані покущів, введені під час оформлення замовлення;
- підтримувати зв'язки між таблицями за допомогою зовнішніх ключів (наприклад, між таблицями товарів, замовлень та позицій замовлення);
- забезпечувати швидке виконання SQL-запитів завдяки правильній індексації полів;
- підтримувати транзакції для забезпечення цілісності даних при обробці замовлень.

Схема бази даних проєктується із дотриманням принципів нормалізації для уникнення надмірності даних та забезпечення їхньої цілісності.

У процесі постановки задачі визначено детальні вимоги до UX/UI дизайну вебресурсу. Якісний дизайн є одним із ключових чинників, що впливають на рішення користувача про покупку та загальне враження від взаємодії з вебсайтом. Інтерфейс вебсайту повинен бути:

- сучасним, з використанням актуальних трендів у вебдизайні;
- мінімалістичним, де пріоритет надається контенту, а не декоративним елементам;
- логічно структурованим із чіткою візуальною ієрархією;
- зручним та інтуїтивно зрозумілим для широкої аудиторії;
- адаптивним та коректно відображуваним на всіх типах пристроїв.

Для покращення взаємодії користувача із системою передбачено використання:

- плавних CSS-анімацій під час переходу між сторінками та відображення елементів;
- hover-ефектів на інтерактивних елементах (кнопки, картки товарів, пункти меню);
- адаптивного мобільного меню з підтримкою жестів на сенсорних пристроях,
- динамічного оновлення інформації (кошик, результати пошуку) без перезавантаження сторінки;
- цілісної кольорової гами, що відповідає тематиці магазину вин та викликає відповідні асоціації у покупця.

Особлива увага приділяється швидкості роботи вебресурсу. Повільна завантаженість сторінок негативно впливає на користувацький досвід та призводить до відтоку відвідувачів. Вебзастосунок повинен:

- швидко завантажувати сторінки завдяки оптимізації розміру зображень та мінімізації HTTP-запитів;
- забезпечувати плавну взаємодію без помітних затримок навіть при великій кількості динамічних елементів;
- ефективно обробляти запити до бази даних завдяки оптимізованим SQL-запитам та індексуванню.

Поряд із продуктивністю необхідно враховувати базові вимоги до безпеки системи:

- валідація та санація введених користувачем даних на клієнтській та серверній стороні;
- базовий захист серверної частини від поширених вразливостей;
- безпечне зберігання чутливих даних;
- стабільна та передбачувана робота API-маршрутів із коректною обробкою виняткових витуацій.

2.2. Проєктування структури вебсайту

Проєктування структури навігації є одним із перших та найважливіших етапів розробки вебзастосунку. Якісна навігаційна структура забезпечує логічний та інтуїтивно зрозумілий шлях користувача від моменту входу на сайт до завершення покупки. Під час проєктування навігації необхідно враховувати принципи UX-дизайну, зокрема мінімальну кількість кроків для досягнення цільової дії, чітку візуальну ієрархію та передбачувану поведінку інтерфейсу.

Структура навігації вебсайту інтернет-магазину вина організована за ієрархічним принципом та поділяється на кілька рівнів. Перший рівень — це головне навігаційне меню, яке є доступним з будь-якої сторінки вебсайту та містить посилання на ключові розділи. Другий рівень — внутрішня навігація окремих сторінок, яка дозволяє здійснювати дії всередині конкретного розділу.

Головне навігаційне меню включає такі основні розділи:

- головна – посилання на головну сторінку із загальною інформацією про магазин та акційними пропозиціями;
- каталог – перехід до повного асортименту акційних товарів;
- акції – перехід до асортименту акційних товарів;
- «Про нас» - сторінка з інформацією про компанію, історію магазину, основні принципи роботи та переваги сервісу;
- доставка – сторінка з описом умов доставки, термінів отримання замовлення та доступних служб;

- оплата – сторінка з інформацією про доступні способи оплати товарів, включаючи оплату онлайн та при отриманні;
- контакти – сторінка з контактною інформацією магазину, адресою, номером телефону, електронною поштою та формою зворотнього зв'язку;
- порівняння – перехід до сторінки порівняння товарів із лічильником доданих позицій;
- обране – сторінка зі списком товарів, які користувач додав до обраного для подальшого перегляду або покупки;
- світла та темна тема – можливість перемикання між світлим і темним оформленням інтерфейсу для підвищення зручності використання сайту;
- кошик – перехід до кошика покупок із динамічним лічильником кількості товарів;
- профіль – перехід до особистого кабінету для авторизованих користувачів або до сторінки входу/реєстрації для неавторизованих.

Приклад головного навігаційного меню наведено на рис. 2.1.



Рисунок 2.1 – Головне навігаційне меню

Для мобільних пристроїв головне меню трансформується у компактну кнопку з трьома горизонтальними лініями, як показано на рис. 2.2. Це рішення є загальноприйнятим стандартом мобільного UX-дизайну та забезпечує зручність використання на малих екранах.

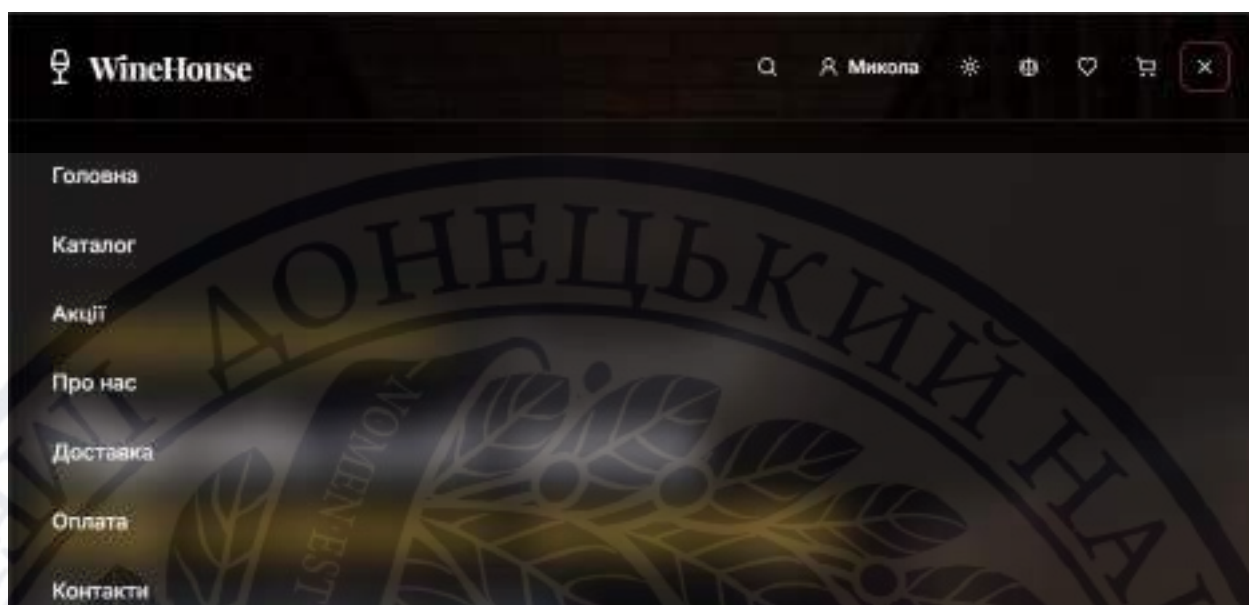


Рисунок 2.2 – Трансформоване головне меню для мобільних пристроїв

Окрім головного меню, вебсайт містить допоміжні навігаційні елементи. Навігаційні ланцюжки відображаються на внутрішніх сторінках каталогу та сторінках товарів, допомагаючи користувачу орієнтуватися у структурі сайту та швидко повертатися на попередні рівні. Нижній колонтитул містить додаткові посилання та контактну інформацію, як на рис. 2.3.



Рисунок 2.3 – Нижній колонтитул

Навігація між кроками оформлення замовлення реалізована у вигляді покрокового індикатора прогресу, який наочно показує поточний етап процесу покупки та дозволяє повертатися на попередні кроки без втрати введених даних.

Для наочного відображення архітектури вебсайту та взаємозв'язків між його сторінками використовується UML-діаграма структури. Діаграма

інформувати про магазин, залучати до перегляду асортименту та спонукати до здійснення покупки.

Структура головної сторінки включає такі блоки:

- Него-секція – повноекранний банер із якісним зображенням, коротким слоганом магазину та кнопкою заклику до дії «Переглянути каталог». Него секція формує емоційне враження від бренду та задає загальний стиль сторінки;
- секція спеціальних пропозицій – горизонтальна стрічка, що дозволяє відвідувачу швидко перейти до акційних товарів;

Приклад головної сторінки наведено на рис 2.5.

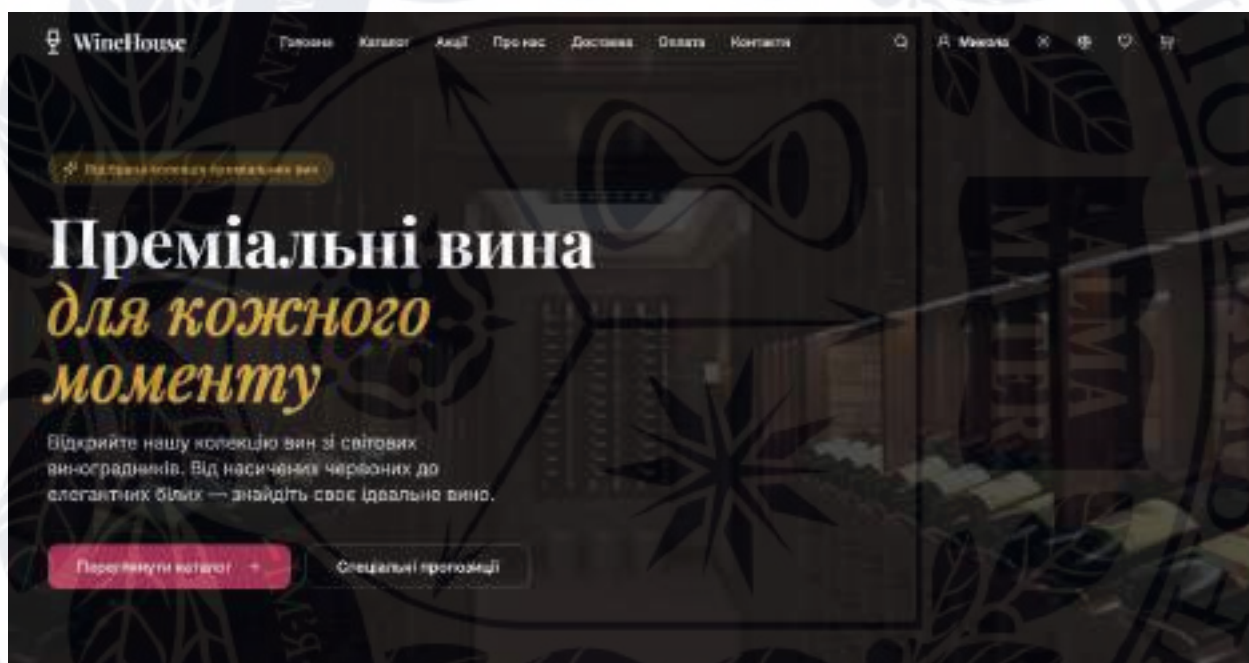


Рисунок 2.5 – Головна сторінка

Сторінка каталогу є центральним функціональним розділом інтернет-магазину, де користувач безпосередньо взаємодіє з асортиментом продукції.

Сторінка каталогу містить такі елементи:

- пошуковий рядок – розміщений у верхній частині сторінки, забезпечує фільтрацію товарів у режимі реального часу за введеним запитом без перезавантаження сторінки;

- сітка карток товарів – основний контентний блок, де кожна картка містить фотографію товару, його назву, коротку характеристику, ціну, кнопку «Додати до кошика» та кнопку «Додати до порівняння»;
- індикатор завантаження – відображається під час отримання даних із сервера, забезпечуючи візуальний зворотний зв'язок для користувача;
- повідомлення про відсутність результатів – відображається у разі, якщо пошуковий запит не повернув жодного товару, із пропозицією змінити запит.

Приклад сторінки каталогу наведено на рис 2.6.

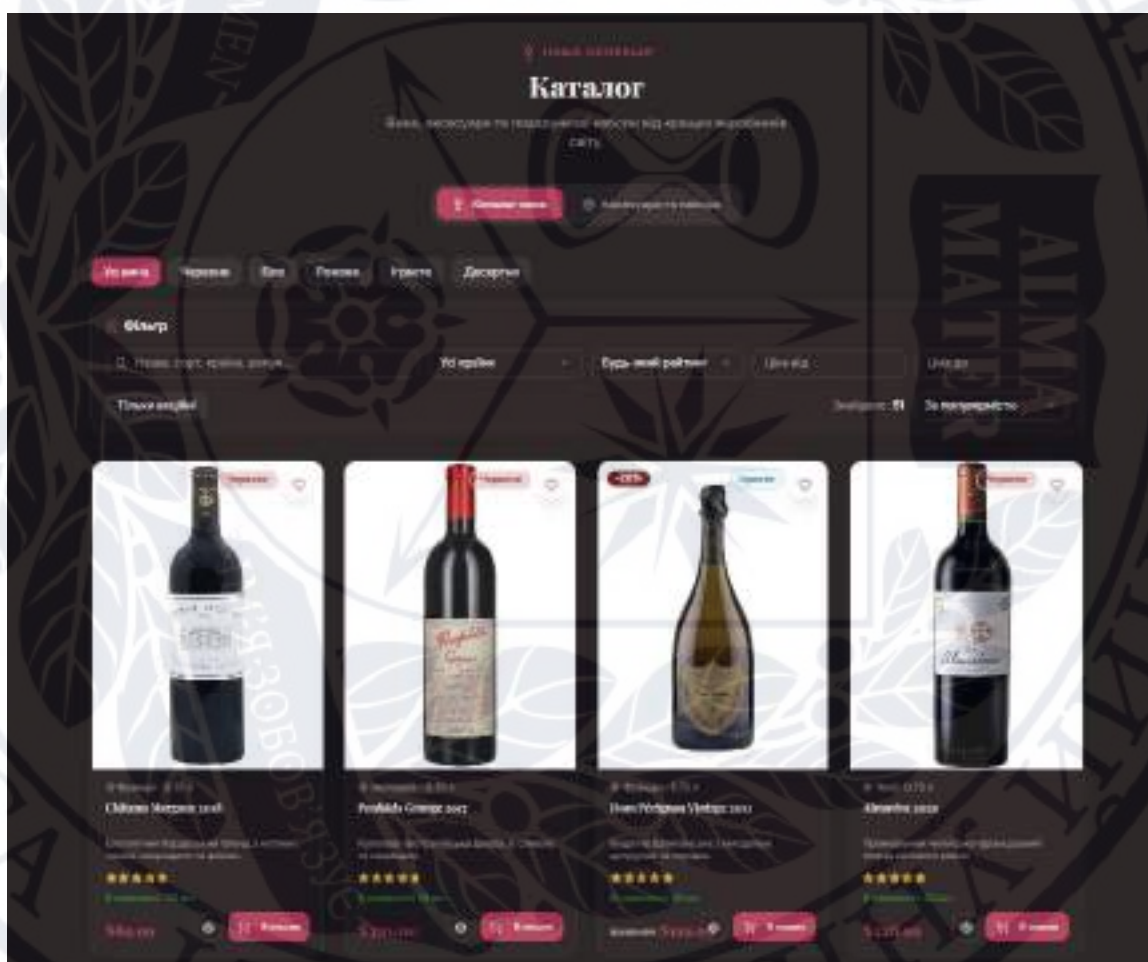


Рисунок 2.6 – Каталог товарів

Картка товару у каталозі проєктується таким чином, щоб надавати максимум корисної інформації при мінімальному розмірі. При наведенні

курсора, картка може розгортати додаткові елементи керування або змінювати свій вигляд для привернення уваги, як позначено на рис. 2.7.



Рисунок 2.7 – Розгорнута картка товару

Сторінка окремого товару надає вичерпну інформацію про конкретну позицію та є ключовим місцем прийняття рішення про покупку.

Сторінка товару включає такі блоки:

- галерея зображень – основне фото товару з можливістю перегляду у збільшеному форматі;
- назва та основні характеристики – повна назва, тип вина, країна походження, рік врожаю, виробник, міцність та об'єм пляшки;
- ціна та кнопки дій – чітко виділена ціна, кнопка «Додати до кошика» з вибором кількості та кнопка «Додати до порівняння»;
- детальний опис – розширений текстовий опис товару, смакові характеристики, рекомендації щодо поєднання з їжею та умови зберігання;

Сторінка кошика відображає всі товари, додані користувачем, та є проміжним етапом перед оформленням замовлення.

Сторінка кошика містить:

- синсок доданих товарів – кожна позиція відображається із фотографією, назвою, ціною за одиницю та можливістю зміни кількості або видалення;
- дільник кількості – кнопки збільшення та зменшення кількості кожного товару з автоматичним перерахунком вартості;
- підсумок замовлення – блок із загальною вартістю всіх товарів, що динамічно оновлюється при зміні вмісту кошика;
- кнопка «Оформити замовлення» – основна кнопка заклик до дії, що переводить користувача на наступний крок;
- повідомлення про порожній кошик – відображається, якщо користувач перейшов на сторінку кошика без доданих товарів, із пропозицією перейти до каталогу.

Приклад сторінки кошика наведено на рис. 2.8.

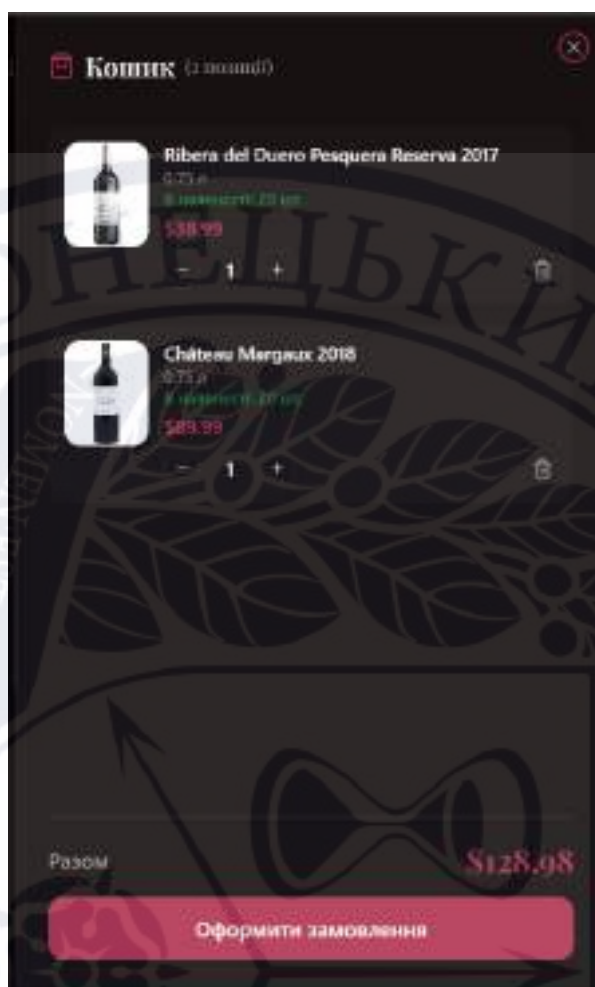


Рисунок 2.8 – Сторінка кошика

Сторінка оформлення замовлення є критично важливим етапом у воронці продажів. Зайва складність або незрозумілість цього процесу безпосередньо призводить до відмови від покупки, тому інтерфейс проєктується максимально простим та зрозумілим. Сторінка оформлення замовлення включає:

- форму введення особистих даних – поля для введення імені, прізвища, номера телефону та email-адреси. Для авторизованих користувачів поля заповнюються автоматично на основі даних профілю;
- блок вибору способу доставки – перемикач між варіантами «Нова пошта» та «Самовивіз». При виборі «Нова пошта» додатково відображаються поля для введення міста та номера відділення або адреси адресної доставки;

- промокод – користувач вводить промокод і застосовується знижка при покупці.
- підсумок замовлення – компактний перелік товарів із кошика із загальною вартістю для підтвердження перед переходом до оплати;
- кнопка «Перейти до оплати» - після натискання система валідує введені дані та, у разі їх коректності, переводить користувача на сторінку оплати.

Приклад оформлення замовлення наведено на рис. 2.9.

Оформлення замовлення
Доставка, дані для доставки

Особисті дані

Ім'я та прізвище *
Микола Скоросад

Телефон *
0955541130

Емейл *
miko@ukr.net

Доставка

☒ Кур'єр ☐ Ново Пошта ☐ Самовивіз

Введіть адресу самовивізу *

м. Вінниця, вул. Пирогова, 70

Спосіб оплати

☒ Картокою онлайн ☐ Преквитання ☐ Банківський переказ

Повне деталізація

Вино для Діда Різника (Вино для Діда Різника 2017) 1 л	2100.00
Вино для Діда Різника (Вино для Діда Різника 2017) 1 л	500.00
Промокод	
активний	
Введіть промокод (максимум 20 символів)	
Сума замовлення	2100.00
Всього до оплати	2600.00
Результат	2600.00

[Перейти до оплати](#)

Рисунок 2.9 – Оформлення замовлення

Сторінка оплати є фінальним кроком перед підтвердженням замовлення. Вона відображає підсумкову інформацію про замовлення та надає користувачу можливість обрати зручний спосіб оплати. Сторінка оплати містить:

- підсумок замовлення – фінальний перелік товарів, спосіб доставки та загальна сума до сплати;
- блок вибору способу оплати – перемикач між доступними варіантами, зокрема оплата карткою онлайн або оплата при отриманні (накладений платіж);
- форму введення платіжних даних – відображається при виборі оплати карткою та містить захищені поля для введення номера картки, терміну дії та CVV-коду;
- кнопку «Підтвердити та оплатити» – після натискання система обробляє платіж та переводить користувача на сторінку підтвердження замовлення;
- індикатори безпеки – візуальні елементи, що підтверджують захищеність з'єднання та платіжних даних (іконки SSL, логотипи платіжних систем), що підвищують довіру користувача.

Як показано на рис. 2.10.

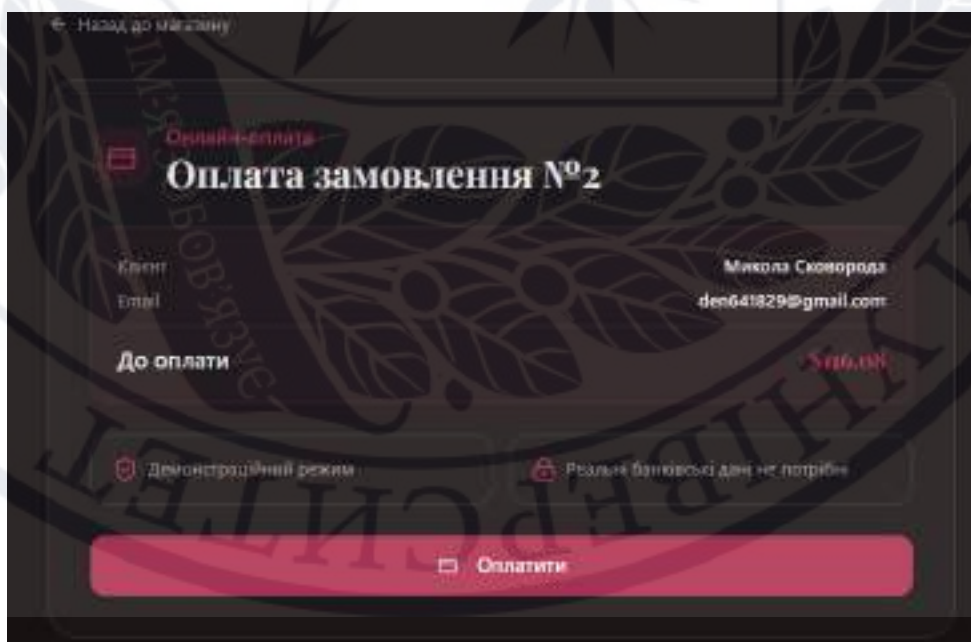


Рисунок 2.10 – Оплата замовлення

Після успішного підтвердження замовлення система автоматично надсилає повідомлення в особистий кабінет з деталями покупки на вказану адресу, а користувач перенаправляється на сторінку підтвердження із номером замовлення та орієнтовними строками доставки, як показано на рис. 2.11.



Рисунок 2.11 – Успішне оформлення оплати

Сторінка профілю доступна виключно для авторизованих користувачів та є персональним центром управління обліковим записом.

Сторінка профілю організована у вигляді розділів із бічною навігацією між ними:

- розділ «Особисті дані» – відображає поточну інформацію користувача (ім'я, прізвище, email, номер телефону) з можливістю редагування кожного поля. Зміни зберігаються після підтвердження та одразу відображаються у формі оформлення замовлення;
- розділ «Мої замовлення» – хронологічний перелік усіх оформлених замовлень із відображенням дати, суми, складу та статусу кожного. При натисканні на конкретне замовлення відкривається детальна сторінка із повною інформацією;
- кнопка «Вийти» – завершення активної сесії із перенаправленням на головну сторінку.

Сторінка профілю проєктується з урахуванням принципу поступового розкриття інформації – користувач бачить лише ту інформацію та ті елементи

керування, які йому необхідні у конкретний момент, без перевантаження інтерфейсу зайвими деталями. Приклад особистого кабінету зображено на рис. 2.12.

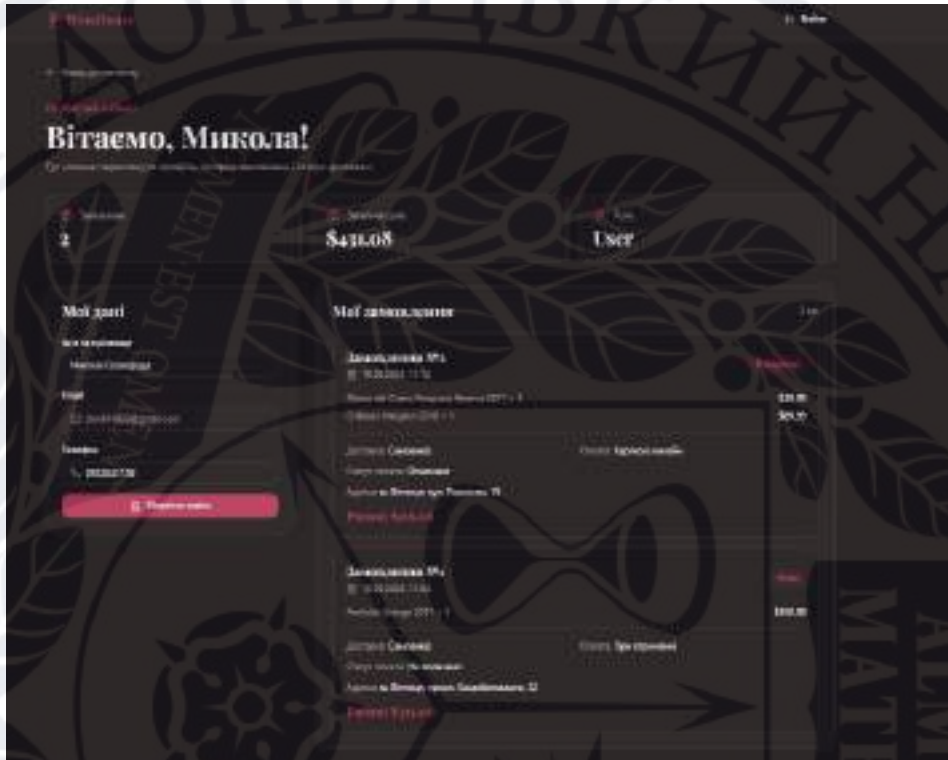


Рисунок 2.12 – Особистий кабінет

2.3. Проектування та реалізація бази даних PostgreSQL

Вибір системи керування базами даних є одним із ключових архітектурних рішень при проектуванні вебзастосунку, оскільки від надійності, продуктивності та функціональності бази даних безпосередньо залежить стабільність роботи всієї системи. У процесі аналізу доступних рішень розглядалися як реляційні СКБД (PostgreSQL, MySQL, SQLite), так і переляційні (MongoDB, Redis). За результатами порівняльного аналізу для реалізації вебзастосунку інтернет-магазину вина було обрано PostgreSQL.

Вибір PostgreSQL обґрунтовується такими перевагами цієї СКБД:

Повна відповідність стандарту SQL. PostgreSQL підтримує стандарт SQL у значно більшому обсязі порівняно з більшістю конкурентів, що забезпечує коректність виконання складних запитів, підзапитів, оконних функцій та

агрегацій. Це особливо важливо для операцій із замовленнями, де необхідно виконувати складні аналітичні запити.

Підтримка складних типів даних PostgreSQL підтримує широкий спектр вбудованих типів даних, зокрема JSONB для зберігання напівструктурованих даних, масиви, перелічувані типи (ENUM), а також геопросторові типи через розширення PostGIS. Для інтернет-магазину це дозволяє гнучко зберігати характеристики товарів різних категорій.

Надійність та підтримка транзакцій. PostgreSQL повністю підтримує властивості ACID (Atomicity, Consistency, Isolation, Durability, що гарантує цілісність даних навіть у разі збоїв систем. Це є критично важливим для обробки замовлень, де часткове збереження даних є неприпустимим.

Розширені можливості індексування. PostgreSQL підтримує різноманітні типи індексів: B-tree, Hash, GIN, GiST, що дозволяє суттєво прискорити виконання запитів до таблиць із великою кількістю записів.

Підтримка зовнішніх ключів та обмежень цілісності. На відміну від деяких NoSQL-систем, PostgreSQL забезпечує декларативне визначення зв'язків між таблицями та автоматичне дотримання обмежень цілісності даних на рівні бази даних.

Масштабованість та продуктивність. PostgreSQL ефективно працює як на невеликих проєктах, так і при великих обсягах даних, забезпечуючи стабільну продуктивність завдяки розвиненому планувальнику запитів.

Безкоштовність та відкритий вихідний код. PostgreSQL поширюється під ліцензією PostgreSQL License, що є дозвоільною ліцензією з відкритим кодом, яка не накладає обмежень на використання у комерційних проєктах.

Порівняно з MySQL, PostgreSQL забезпечує кращу підтримку стандарту SQL та більш розвинені механізми роботи з транзакціями. Порівняно з MongoDB, PostgreSQL гарантує строгу цілісність даних через зовнішні ключі та обмеження, що є важливим для фінансових операцій інтернет-магазину.

Розробка структури бази даних розпочинається з аналізу предметної області та визначення сутностей, які необхідно відображати у системі. На основі

функціональних вимог, були виявлені такі основні сутності: користувачі, товари, категорії товарів, замовлення, елементи замовлень, способи оплати, список обраних товарів, список порівняння товарів та промокоди.

При проєктуванні структури бази даних дотримувалися принципів нормалізації до третьої нормальної форми (3НФ), що забезпечує відсутність надмірності даних, логічну організацію таблиць та мінімізацію аномалій при операціях вставки, оновлення та видалення записів.

Перша нормальна форма (1НФ) дотримується завдяки тому, що всі атрибути таблиць є атомарними – кожна комірка містить лише одне значення, а кожен рядок є унікальним завдяки первинному ключу.

Друга нормальна форма (2НФ) забезпечується відсутністю часткових залежностей – усі неключові атрибути повністю залежать від первинного ключа таблиці, а не від його частини.

Третя нормальна форма (3НФ) досягається виключенням транзитивних залежностей – неключові атрибути залежать виключно від первинного ключа, а не від інших неключових атрибутів.

Загальна структура бази даних складається з десяти взаємопов'язаних таблиць, між якими визначені відносини «один до одного», «один до багатьох» та «багато до багатьох» через проміжні таблиці.

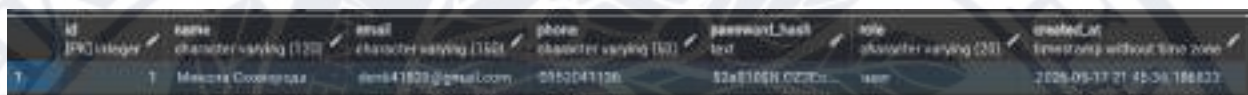
ER-діаграма є графічним відображенням структурн бази даних, що показує сутності, їхні атрибути та зв'язки між ними, як показано на рис. 2.13.



Рисунок 2.13 – ER-діаграма бази даних

ER-діаграма наочно демонструє, що центральними сутностями системи є таблиці `users` та `products`, з якими так чи інакше пов'язані всі інші таблиці бази даних.

Таблиця `users` є однією з ключових таблиць бази даних і призначена для зберігання облікових записів зареєстрованих покупців. Саме до цієї таблиці прив'язані замовлення, списки обраних товарів та списки порівняння. Приклад таблиці `users` наведено на рис. 2.14.



id	name	email	phone	password_hash	role	created_at
1	Микола Коцюк	denis41823@gmail.com	0952041136	\$2a\$10\$H02T...	user	2026-05-17 21:45:58.186233

Рисунок 2.14 – Таблиця `users`

Опис полів таблиці `users`:

- `id` – унікальний ідентифікатор користувача, генерується автоматично;
- `name` – ім'я та прізвище користувача, обов'язкові для заповнення;
- `email` – електронна адреса, використовується як логін; обмеження `UNIQUE` гарантує відсутність дублікатів;
- `phone` – контактний номер телефону;
- `password_hash` – хеш пароля, отриманий алгоритмом `bcrypt`; пароль у відкритому вигляді ніколи не зберігається;
- `role` – роль користувача на сайті;
- `created_at` – дата та час реєстрації, встановлюється автоматично.

Таблиця `products` зберігає повну інформацію про кожну позицію асортименту інтернет-магазину. Вона є центральною сутністю каталогу та пов'язана із таблицями категорії, елементів замовлень, списків обраних товарів та порівняння, як показано на рис. 2.15.

id	name	type	country	region	grape	volume
integer	varchar varying (255)	varchar varying (50)	varchar varying (100)	varchar varying (100)	varchar varying (255)	integer
1	Chateau Margaux 2018	red	France	Bordeaux	Cabernet Sauvignon, Merlot	6.75 n
2	Barolo Rosso 2016	red	Italy	Piedmont	Barolo	6.75 n
3	Tuscanella 2019	red	Italy	Tuscany	Sangiovese, Cabernet Sauvignon	6.75 n
4	Penfolds Shiraz 2017	red	Australia	South Australia	Shiraz	6.75 n
5	Marques de Val de Mar 2018	red	Spain	Rioja	Tempranillo	6.75 n
6	Castello del Duilio Cabernet Sauvignon 2021	red	Italy	Central Valley	Cabernet Sauvignon	6.75 n
7	Colares Zinfandel 2018	red	America	Monterey	Zinfandel	6.75 n
8	Ornellaia di Montalcino 2017	red	Italy	Tuscany	Sangiovese, Cabernet	6.75 n
9	Antinori Pinot 2021	red	Italy	Tuscany	Pinot Noir	6.75 n
10	Koblenz Merlot Reserve 2020	red	Germany	Rheinhessen	Merlot	6.75 n
11	Gaja Barolo 2018	red	Italy	Piedmont	Barolo	6.75 n
12	Cloudy Bay Sauvignon Blanc 2022	white	New Zealand	Marlborough	Sauvignon Blanc	6.75 n
13	Charles Heidsieck 2020	white	France	Burgundy	Chardonnay	6.75 n
14	Pinot Noir 2021	white	France	Bordeaux	Pinot Noir	6.75 n
15	Santa Margherita Pinot Grigio 2022	white	Italy	Abruzzo	Pinot Grigio	6.75 n
16	Pazo do Gallego Albariño 2022	white	Spain	Rias Baixas	Albariño	6.75 n
17	Barossa Valley Shiraz 2022	white	Australia	Barossa Valley	Shiraz	6.75 n
18	Grado Pinot Noir 2022	white	USA	Washington	Pinot Noir	6.75 n
19	Freixas Pinot Noir 2022	white	France	Bordeaux	Pinot Noir	6.75 n
20	El Estero de los Hornos 2022	white	Argentina	Mendoza	Pinot Noir	6.75 n
21	Whispering Angel Rose 2022	rose	France	Provence	Rosé	6.75 n
22	Muscat Rose 2022	rose	France	Provence	Rosé	6.75 n
23	Donatello Pinot 2022	white	France	Provence	Pinot Noir	6.75 n

Рисунок 2.15 – Таблица products

Опис полів таблиці products:

- id – унікальний ідентифікатор товару;
- name – повна назва товару,
- type – тип вина (наприклад: червоне, біле, рожеве, ігристе);
- country – країна походження вина;
- region – регіон виробництва вина;
- grape – сорт винограду, з якого виготовлено вино;
- volume – об'єм пляшки у літрах;
- alcohol – вміст алкоголю у відсотках;
- price – поточна ціна товару,
- old price – попередня ціна (використовується для знижок/акцій);
- rating – середній рейтинг товару (наприклад, від 1 до 5);
- popularity – показник популярності товару, рейтинг;
- is promotion – логічне значення (true/false), чи бере участь товар в акції;
- short_description – короткий опис товару для карток/каталогу;
- description – детальний текстовий опис, смакові характеристики та рекомендації;

- `image_url` – URL-адреса фотографії товару;
- `contents` – склад або додаткова інформація про продукт (наприклад: сульфіти, витримка, цукор, додаткові характеристики);
- `stock` – кількість одиниць товару на складі з перевіркою на невід’ємне значення;
- `created_at` – дата та час створення запису про товар;
- `updated_at` – дата та час останнього оновлення запису про товар.

Таблиця `product_categories` призначена для класифікації товарів за типом вина та забезпечує можливість фільтрації асортименту у каталозі, як показано на рис 2.16.

	<code>id</code> [PK] integer	<code>slug</code> character varying (50)	<code>name</code> character varying (100)	<code>product_group</code> character varying (30)
1	1	red	Червоне вино	wine
2	2	white	Біле вино	wine
3	3	rose	Рожеве вино	wine
4	4	sparkling	Ігристе вино	wine
5	5	dessert	Десертне вино	wine
6	6	accessory	Акcesуар	accessory
7	7	gift	Подарунковий набір	gift

Рисунок 2.16 – Таблиця `product_categories`

Опис полів таблиці `categories`:

- `id` – унікальний ідентифікатор категорії;
- `slug` – URL-індетифікатор категорії у зручному для адреси форматі;
- `name` – назва категорії (наприклад, «Червоне вино», «Біле вино», «Ігристе вино», «Рожеве вино», «Десертне вино»); обмеження `UNIQUE` виключає дублювання назв;
- `product group` – група товарів, до якої належить категорія.

Таблиця `orders` зберігає загальну інформацію про кожне оформлене замовлення. Вона є сполучною ланкою між покупцем, способом оплати та переліком придбаних товарів, як показано на рис. 2.17.

id	customer_id	user_id	delivery_method	payment_method	payment_status	paid_at	city
[PK] integer	integer	integer	character varying (30)	character varying (30)	character varying (30)	timestamp without time zone	character varying (100)
1	1	1	pickup	cod	unpaid	null	null
2	2	2	pickup	card	paid	2026-05-19 11:34:12.376682	null

Рисунок 2.17 – таблиця orders

Опис полів таблиці orders:

- id – унікальний ідентифікатор замовлення;
- customer_id – ідентифікатор клієнта;
- user id – зовнішній ключ на таблицю користувачів; при видаленні користувача встановлюється значення NULL (ON DELETE SET NULL), зберігаючи історію замовлень;
- delivery_method – спосіб доставки;
- payment method – спосіб оплати;
- payment _status – статус оплати;
- paid_at – дата та час фактичної оплати замовлення;
- city – місто доставки замовлення;
- address – повна адреса доставки;
- nova_post_address – адреса відділення «Нової Пошти»;
- nova_post number – номер відділення «Нової Пошти»;
- pickup_address – адреса точки самовивозу;
- total price – загальна сума замовлення;
- subtotal_price – сума товарів до застосування знижок;
- discount_amount – сума знижки;
- promocode id – ідентифікатор застосованого промокоду;
- promocode _code – код використаного промокоду;
- status – поточний статус замовлення з чітко визначеним переліком допустимих значень;
- created _at – дата та час створення замовлення.

Таблиця order_items реалізує відношення «багато до багатьох» між замовленнями та товарами, зберігаючи конкретний перелік позицій у кожному замовленні, як позначено на рис. 2.18.

	id [PK] integer	order_id integer	product_id integer	product_name character varying (255)	quantity integer	price numeric (10,2)
1	1	1	4	Penfolds Grange 2017	1	350.00
2	2	2	39	Ribera del Duero Pesquera Reserva 20...	1	38.99
3	3	2	1	Château Margaux 2018	1	89.99

Рисунок 2.18 – таблиця order_items

Опис полів таблиці order_items:

- order item id – унікальний ідентифікатор елемента замовлення;
- order id – зовнішній ключ на замовлення;
- product_id – зовнішній ключ на товар;
- product_name – назва товару, що зберігаються незалежно від можливих змін у таблиці товарів;
- quantity – кількість одиниць товару з перевіркою на позитивне значення;
- price – ціна товару.

Таблиця user_wishlist реалізує функціонал списку бажань, дозволяючи авторизованим користувачам зберігати товари для подальшого розгляду, як позначено на рис. 2.19.

	id [PK] integer	user_id integer	product_id integer	created_at timestamp without time zone
1	1	1	50	2026-05-19 13:34:19.372842
2	2	1	12	2026-05-19 13:34:23.093764

Рисунок 2.19 – таблиця user_wishlist

Опис полів таблиці wishlist:

- id – унікальний ідентифікатор запису;
- user_id – зовнішній ключ на користувача;
- product_id – зовнішній ключ на товар;
- created_at – дата та час додавання товару до списку;

Таблиця reviews призначена для зберігання відгуків та оцінок вив, що дозволяє користувачам ознайомлюватися з досвідом інших покупців і робити більш обґрунтований вибір продукції, як показано на рис. 2.20.

	id [PK] integer	user_id integer	product_id integer	rating integer	comment text	created_at timestamp without time zone
1	1	1	1	4	Відмінний смак, відчувається...	2026-05-19 13:43:34.717347

Рисунок 2.20 – Таблиця reviews

Опис полів таблиці reviews:

- id – унікальний ідентифікатор відгуку;
- user_id – зовнішній ключ на таблицю користувачів, який визначає автора відгуку;
- product_id – зовнішній ключ на таблицю товарів, до якого залишено відгук;
- rating – оцінка товару у вигляді числового значення;
- comment – текстовий коментар користувача з його враженнями про продукт;
- created_at – дата та час створення відгуку.

Таблиця customers призначена для зберігання інформації про клієнтів, які оформлюють замовлення в інтернет-магазині. Вона дозволяє ідентифікувати покупців та забезпечує можливість комунікації з ними, приклад таблиці зображено на рис. 2.21.

	id [PK] integer	full_name character varying (150)	phone character varying (50)	email character varying (150)	created_at timestamp without time zone
1	1	Микола Сковорода	0952041136	den541829@gmail.com	2026-05-18 11:04:55.565799
2	2	Микола Сковорода	0952041136	den541829@gmail.com	2026-05-19 11:12:22.57884

Рисунок 2.21 – Таблиця customers

Опис полів таблиці customers:

- id – унікальний ідентифікатор клієнта;
- full_name – повне ім'я клієнта;
- phone – контактний номер телефону;
- email – електронна адреса клієнта;

- created at – дата та час створення запису про клієнта в системі.

Таблиця `promocodes` призначена для зберігання інформації про промокоди, які використовуються для надання знижок користувачам. Вона дозволяє керувати акційними пропозиціями, обмеженнями використання та термінами дії промокодів, як показано на рис. 2.22.

id	code	discount_type	discount_value	is_active	usage_limit	used_count	expires_at	created_at
1	WINE10	percent	10.00	true	100	2	[null]	2026-05-17 21:43:31.878304
2	VIP20	percent	20.00	true	50	0	[null]	2026-05-17 21:43:31.878304
3	GIFT5	fixed	5.00	true	100	0	[null]	2026-05-17 21:43:31.878304

Рисунок 2.22 – Таблиця `promocodes`

Опис полів таблиці `promocodes`:

- `id` – унікальний ідентифікатор промокоду;
- `code` – унікальний текстовий код промокоду;
- `discount_type` – тип знижки;
- `discount_value` – значення знижки;
- `is_active` – статус активності промокоду;
- `usage_limit` – максимальна кількість використань промокоду;
- `used_count` – кількість вже використаних активацій промокоду;
- `expires_at` – дата та час закінчення дії промокоду;
- `created_at` – дата та час створення промокоду.

Таблиця `cart_items` призначена для зберігання товарів, які користувач додає до кошика. Вона дозволяє відстежувати вибрані продукти, їх кількість та оновлення в процесі формування замовлення, приклад таблиці зображено на рис. 2.23.

id	user_id	product_id	quantity	created_at	updated_at
1	8	1	32	2026-05-19 14:03:08.649301	2026-05-19 14:03:08.649301

Рисунок 2.23 – Таблиця `cart_items`

Опис полів таблиці `cart_items`:

- `id` – унікальний ідентифікатор елемента кошика;

- `user_id` – зовнішній ключ на таблицю користувачів, що визначає власника кошика;
- `product_id` – зовнішній ключ на таблицю товарів;
- `quantity` – кількість одиниць обраного товару у кошику;
- `created_at` – дата та час додавання товару до кошика;
- `updated_at` – дата та час останнього оновлення запису.

2.4. Реалізація клієнтської частини вебсайту

Клієнтська частина вебзастосунку інтернет-магазину вина реалізована з використанням бібліотеки React – однієї з найпопулярніших та найбільш розвинених JavaScript-бібліотек для побудови користувацьких інтерфейсів.

Вибір React як основного інструменту для побудови клієнтської частини обґрунтовується такими перевагами:

Компонентна архітектура. React побудована на концепії компонентів – самодостатніх, повторно використовуваних блоків інтерфейсу, кожен з яких відповідає за відображення певної частини сторінки та управління власним станом. Це дозволяє розбити складний інтерфейс на логічні, незалежні частини, спрощує підтримку коду та забезпечує його масштабованість.

Віртуальний DOM. React використовує концепцію віртуального DOM – легковагового представлення реального DOM-дерева у пам'яті. При зміні стану компонента React спочатку оновлює віртуальний DOM, потім порівнює його з попереднім станом та вносить лише мінімально необхідні зміни до реального DOM.

Це суттєво підвищує продуктивність інтерфейсу, особливо при частих оновленнях, наприклад при динамічному відображенні кошика або результатів пошуку.

У React дані передаються від батьківських компонентів до дочірніх через механізм `props`, що забезпечує передбачуваність поведінки інтерфейсу та спрощує відстеження джерела змін у разі виникнення помилок.

Хуки (Hooks), починаючи з версії 16.8, React підтримує хуки – функції, що дозволяють використовувати стан та інші можливості React у функціональних компонентах без необхідності писати класи. Використання хуків `useState`, `useEffect`, `useContext` та `useCallback` значно спрощує логіку компонентів та робить код більш читабельним.

Розвинена екосистема React має широку екосистему допоміжних бібліотек: `React Router` для клієнтської маршрутизації, `Axios` для HTTP-запитів, `Context API` та `Redux` для управління глобальним станом застосунку.

Проект ініціалізується за допомогою інструменту `Create React App` або `Vite`, що надає готову конфігурацію середовища розробки із підтримкою гарячого перезавантаження, транспіляції сучасного JavaScript через `Babel` та збірки проєкту через `Vite`.

Компонентний підхід є основоположним принципом архітектури React-застосунку. Весь інтерфейс вебсайту розбивається на ієрархію компонентів, де кожен компонент відповідає за відображення конкретної частини сторінки та управління власною логікою.

У проєкті виділяються три рівні компонентів залежно від їхньої ролі та рівня абстракції.

Компоненти розмітки (`Layout Components`) визначають загальну структуру сторінок та включають елементи, спільні для всіх або більшості сторінок застосунку.

Компонент `Header` відповідає за відображення навігаційного меню, логотипу, пошукового рядка, іконок кошика, порівняння та профілю із відповідними лічильниками. Він підписаний на глобальний стан кошика та авторизації через `Context API` і реагує на їхні зміни в режимі реального часу.

Компоненти сторінок (`Page Components`) відповідають за відображення конкретних сторінок застосунку та організацію їхнього вмісту. Кожен `page-компонент` отримує необхідні дані через `API-запити` або з глобального стану та передає їх дочірнім компонентам через `props`.

Атомарні компоненти (UI Components) є найдрібнішими повторно використовуваними елементами інтерфейсу. Прикладом є компонент ProductCard, який відображає картку товару в каталозі.

Компонентний підхід забезпечує повторне використання коду – наприклад, компонент ProductCard використовується як у каталозі, так і на головній сторінці у секції рекомендованих товарів, що виключає дублювання розмітки та логіки.

Для управління глобальним станом застосунку використовується React Context API у поєднанні з хуком useReducer. Це рішення є достатнім для масштабу поточного проєкту та не потребує підключення зовнішніх бібліотек управління станом, таких як Redux. Глобальний стан організований у три незалежні контексти: CartContext (стан кошика), AuthContext (стан авторизації) та ComparisonContext (стан списку порівняння).

Адаптивний дизайн клієнтської частини реалізується через поєднання гнучкої сіткової системи CSS Grid та Flexbox, медіа-запитів CSS і принципу Mobile First – спочатку визначаються стилі для мобільних пристроїв, а потім послідовно розширюються для більших екранів.

Система контрольних точок (breakpoints) визначена як набір глобальних CSS-змінних.

Сітка каталогу товарів реалізована через CSS Grid із автоматичною зміною кількості стовпців залежно від ширини екрана.

Навігаційне меню трансформується у бургер-меню на мобільних пристроях. На екранах із шириною менше 768px кнопка бургер-меню стає видимою, а навігаційні посилання приховуються та відображаються у вигляді бічної панелі при натисканні.

Сторінка порівняння товарів на мобільних пристроях реалізує горизонтальну прокрутку таблиці, оскільки відображення трьох-чотирьох стовпів на малому екрані в стандартному форматі є неможливим.

Зображення у всіх блоках визначені як max-width: 100% та height: auto, що забезпечує їхнє коректне масштабування на будь-якому екрані без спотворення пропорцій.

Тапографіка також адаптується до розміру екрана: розміри шрифтів, відступи та міжрядкові інтервали визначені з використанням відносних одиниць `rem` та `em`, а також функції `CSS clamp()` для плавного масштабування між мінімальним та максимальним значеннями.

Кошик покупок реалізований як глобальний стан застосунку через `CartContext`, що забезпечує доступ до його вмісту з будь-якого компонента без необхідності передавати дані через довгий ланцюг `props`.

Логіка кошика реалізована у `CartContext` з використанням хука `useReducer`. Загальна вартість кошика обчислюється як похідна величина від масиву товарів і не зберігається у стані окремо, що виключає можливість неузгодженості між вмістом кошика та відображуваною сумою.

Процес оформлення замовлення реалізований як багатокроковий інтерфейс (`multi-step form`). Стан форми зберігається у локальному стані компонента `CheckoutPage` і не скидається при переходах між кроками.

Валідація полів форми виконується на клієнтській стороні перед кожним переходом до наступного кроку. Перевіряється коректність формату `email`, довжина та формат номера телефону, обов'язковість заповнення необхідних полів. При виявленні помилок поля підсвічуються червоним кольором із відповідним повідомленням під полем введення.

Після успішного підтвердження замовлення клієнтська частина надсилає `POST`-запит до серверного `API`, очищає кошик та перенаправляє користувача на сторінку підтвердження замовлення з відображенням його номера.

Сторінка каталогу отримує дані про товари з серверного `API` при першому завантаженні через хук `useEffect`. Для уникнення блокування інтерфейсу під час очікування відповіді сервера відображається компонент-завантажувач (`Loader`).

Пошук реалізований на клієнтській стороні із фільтрацією вже завантаженого масиву товарів без додаткових запитів до сервера, що забезпечує миттєву відповідь інтерфейсу на введення користувача. Метод `ILIKE`-подібного порівняння через `toLowerCase` забезпечує нечутливість пошуку до регістру символів.

Пошуковий рядок реалізований з функцією `debounce` - затримкою обробки введення на 300 мілісекунд після останнього сніволу, що запобігає надмірно частим перерендерингам компонента при швидкому введенні тексту.

Модальні вікна є важливим елементом інтерфейсу інтернет-магазину, оскільки дозволяють відображати додаткову інформацію або запити підтвердження без переходу на нову сторінку, зберігаючи контекст поточного перегляду.

У проєкті реалізований універсальний компонент `Modal`, який використовується в різних частинах застосунку.

Компонент реалізує декілька важливих UX-рішень: блокування прокрутки фонового вмісту при відкритому модальному вікні, закриття модального вікна при натисканні клавіші `Escape` або кліку на затемнений фон, а також рендеринг через `createPortal` безпосередньо у кореневий елемент документа, що уникає проблем із CSS-стилізацією та `z-index`.

Модальні вікна використовуються у таких сценаріях: підтвердження видалення товару з кошика, повідомлення про успішне додавання товару до кошика, попередження при спробі додати п'ятий товар до списку порівняння, а також відображення збільшеного зображення товару на сторінці продукту.

Динамічний контент реалізується через умовне рендерування та анімації CSS-переходів. Повідомлення про успішні дії (`toast-notifications`) відображаються у куті екрана та автоматично зникають через визначений проміжок часу.

Анімації переходів між сторінками реалізовані через CSS-класи з визначеними властивостями `transition` та `animation`, що створює плавне відчуття навігації між розділами вебсайту без різких переключень. Компоненти-скелетони відображаються замість карток товарів під час завантаження даних із сервера, що забезпечує позитивне сприйняття швидкості роботи застосунку навіть при повільному з'єднанні – користувач бачить структуру майбутнього контенту замість порожньої сторінки або простого спінера.

2.5. Реалізація адаптивності та оптимізації UX/UI дизайну

Однією з основних вимог до сучасних вебзастосунків є забезпечення адаптивності інтерфейсу та високого рівня користувацького досвіду. Для інтернет-магазинів це має особливе значення, оскільки значна кількість користувачів здійснює покупки саме через мобільні пристрої. Саме тому під час розробки вебзастосунку інтернет магазину вина, велика увага приділялася реалізації адаптивного дизайну та оптимізації UX/UI інтерфейсу.

- персональних комп'ютерах;
- ноутбуках;
- планшетах;
- смартфонах;

Для реалізації адаптивного дизайну використовувалися сучасні можливості CSS:

- Flexbox;
- CSS Grid;
- Media queries;
- Responsive layout.

Використання Flexbox дозволило створити гнучку структуру сторінок, яка автоматично змінює розташування елементів залежно від ширини екрана. CSS Grid застосовується для побудови адаптивних блоків каталогу товарів та структури сторінок вебресурсу.

Важливу роль у реалізації адаптивного дизайну відіграють media queries. За допомогою media queries були створені окремі стилі для різних роздільних здатностей екрана. Це дозволило автоматично змінювати:

- розмір тексту;
- структуру меню;
- ширину блоків;
- розташування елементів;
- відступи між компонентами.

Під час реалізації адаптивності особлива увага приділялася мобільній версії вебсайту. Для смартфонів було створено спеціальне адаптивне меню типу burger-тепи, яке займає мінімальну кількість простору та забезпечує швидкий доступ до основних сторінок вебресурсу.

Однією з важливих складових UX/UI дизайну є зручність навігації. Під час розробки вебзастосунку було створено логічну структуру сторінок, яка дозволяє користувачу швидко переходити між основними розділами системи. Для покращення користувацького досвіду меню навігації було зроблено фіксованим, що дозволяє отримувати доступ до основних функцій вебсайту незалежно від положення сторінки.

Важливу роль у UX/UI дизайні відіграє кольорова схема вебсайту. Для інтернет-магазину вина використовувалася темна кольорова гама із використанням бордових, чорних та золотистих відтінків, які створюють атмосферу преміального сервісу та відповідають тематиці вебресурсу.

Окрему увагу було приділено типографіці та читабельності тексту. Для інтерфейсу використовувалися сучасні шрифти із достатнім міжрядковим інтервалом та контрастністю тексту. Це дозволило забезпечити комфортне сприйняття інформації користувачем на різних пристроях.

Для покращення взаємодії користувача із вебсайтом використовувалися:

- плавні анімації;
- hover-ефекти;
- анімовані кнопки;
- плавна поява елементів;
- ефекти наведення на товари.

Такі рішення роблять інтерфейс більш сучасним та покращують візуальне сприйняття вебресурсу.

Важливою складовою UX/UI оптимізації стала швидкість роботи вебзастосунку. Для покращення продуктивності було виконано:

- оптимізацію зображення;
- зменшення розміру CSS та JavaScript файлів;

- компонентну структуру React;
- динамічне оновлення сторінок.

Завдяки використанню React оновлення елементів інтерфейсу виконується значно швидше, що позитивно впливає на користувацький досвід.

Для каталогу товарів використовувалася адаптивна система карток продукції. На великих екранах товари відображаються у вигляді декількох колонок, а на мобільних пристроях автоматично переходять у вертикальне розташування.

Під час реалізації UX/UI дизайну також враховувалися принципи мінімалізму та простоти використання. Інтерфейс вебсайту не перевантажений зайвими елементами, а основні функції доступні користувачу у декілька кліків. Для підвищення зручності оформлення замовлення було реалізовано:

- автоматичне оновлення кошика;
- швидкий перехід до Checkout;
- динамічне відображення способів доставки;
- повідомлення про успішне оформлення покупки.

Усі елементи інтерфейсу були протестовані на різних розмірах екранів для забезпечення стабільної роботи вебресурсу.

Таким чином, у процесі реалізації адаптивності та оптимізації UX/UI дизайну було створено сучасний інтерфейс вебзастосунку, який забезпечує комфортну взаємодію користувача із системою на різних типах пристроїв. Використання адаптивної верстки, сучасних UX/UI рішень та оптимізації продуктивності дозволило створити функціональний, швидкий та зручний вебресурс.

РОЗДІЛ 3. ТЕСТУВАННЯ ТА АНАЛІЗ РОБОТИ ВЕБСАЙТУ

3.1. Тестування функціоналу вебсайту

Після завершення реалізації вебзастосунку було проведено функціональне тестування основних модулів системи. Основною метою тестування є перевірка правильності роботи функціоналу вебсайту, стабільності взаємодії між frontend та backend частинами системи, а також коректності обробки інформації у базі даних.

У процесі тестування перевірялися:

- робота каталогу товарів;
- система пошуку;
- функціонал кошика;
- робота бази даних;
- взаємодії frontend та backend;
- адаптивність сторінок.

Одним із перших етапів тестування була перевірка каталогу товарів. Під час тестування оцінювалася коректність:

- завантаження товарів;
- відображення фотографій;
- виведення цін;
- роботи кнопок додавання до кошика.

Було встановлено, що всі товари коректно завантажуються із бази даних та відображаються у каталозі без помилок, як зображено на рис. 3.1.

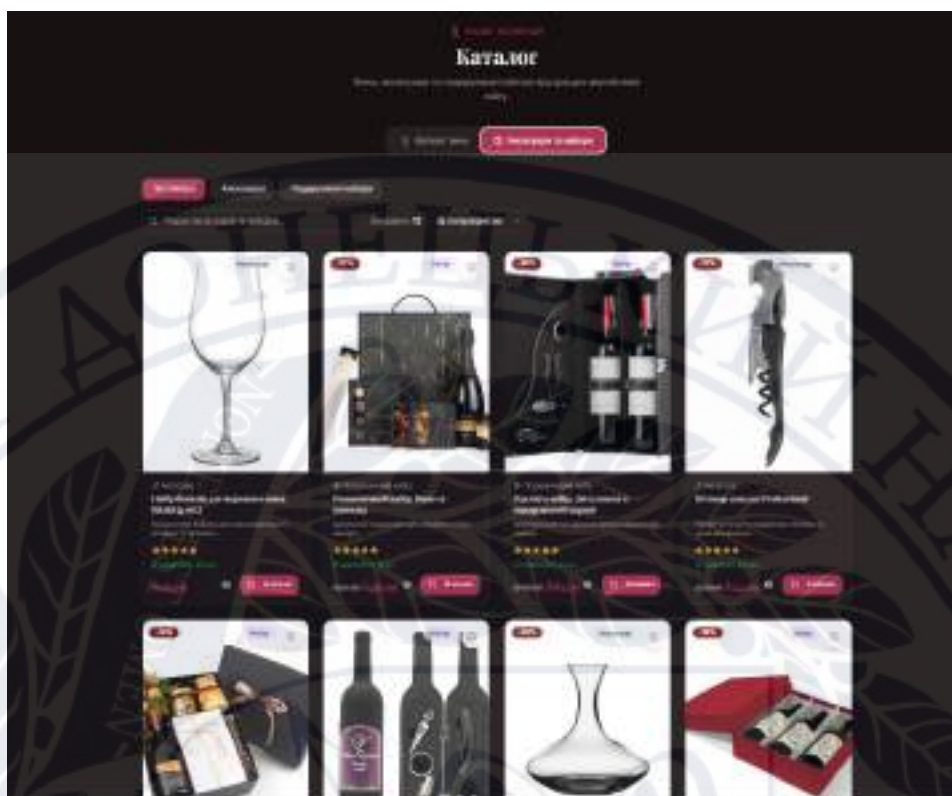


Рисунок 3.1 – Тестування каталогу товарів

Наступним етапом стало тестування системи пошуку товарів. Користувач вводив назву продукції у пошуковий рядок, після чого система автоматично відображала відповідні товари. Пошук виконувався динамічно без перезавантаження сторінки, що забезпечує швидку взаємодію користувача із вебресурсом. Під час тестування було перевірено:

- пошук повної назви товару;
- пошук частини назви;
- відображення результатів;
- швидкість роботи пошукової системи.

Було встановлено, що товари по запиту коректно відображаються у каталозі без помилок, як зображено на рис. 3.2.

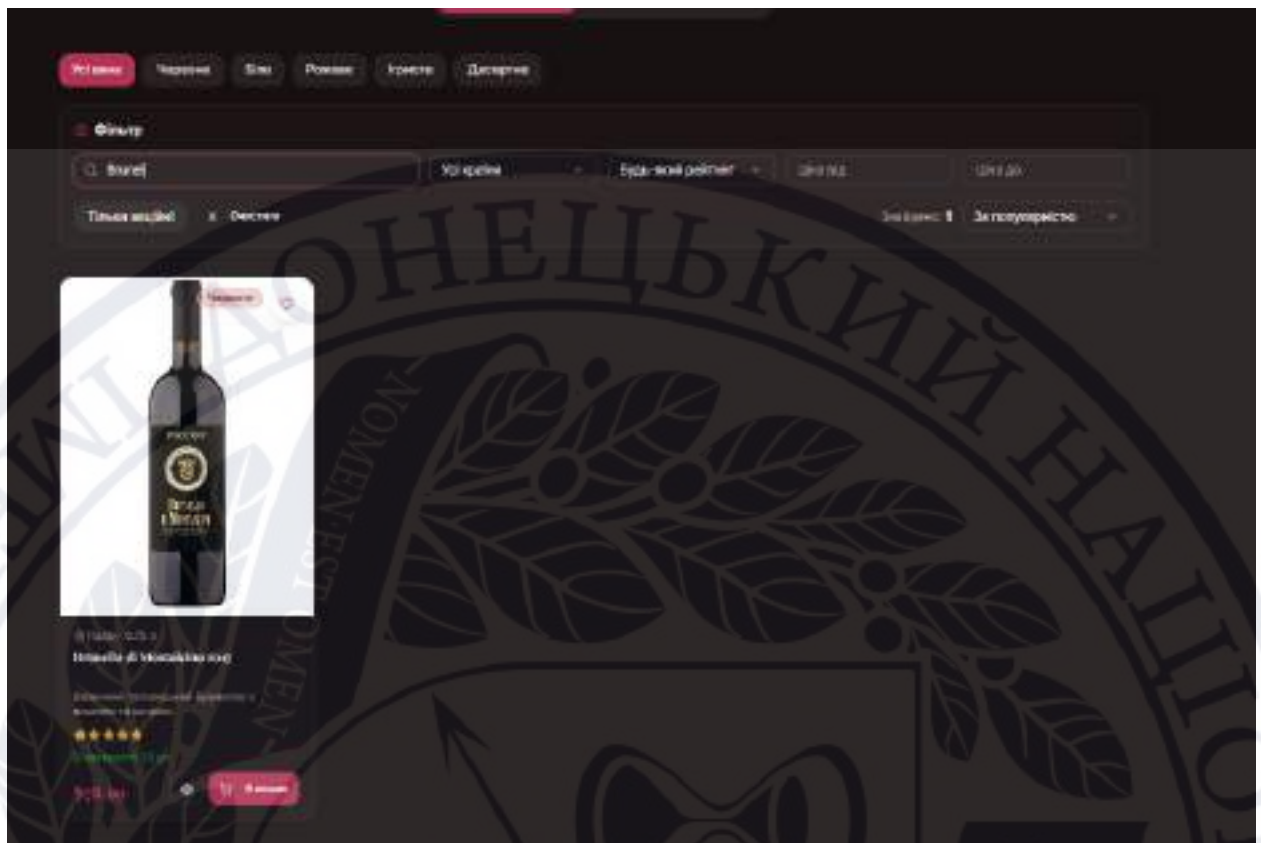


Рисунок 3.2 – Тестування системи пошуку

Важливою складовою тестування стала перевірка функціоналу кошика. Під час тестування користувач додавав товари до кошика, змінював кількість продукції та видаляв товари із списку покупок. Було встановлено, що система коректно:

- оновлює кількість товарів;
- обчислює загальну суму замовлення;
- зберігає інформацію про кошик;
- очищує кошик після оформлення покупки.

Було встановлено, що функціонал кошика працює коректно та без помилок. Товари успішно додаються до кошика, кількість продукції правильно оновлюється, а загальна сума замовлення автоматично перераховується. Також система коректно видаляє товари із кошика та очищує його після оформлення покупки, як зображено на рис. 3.3.

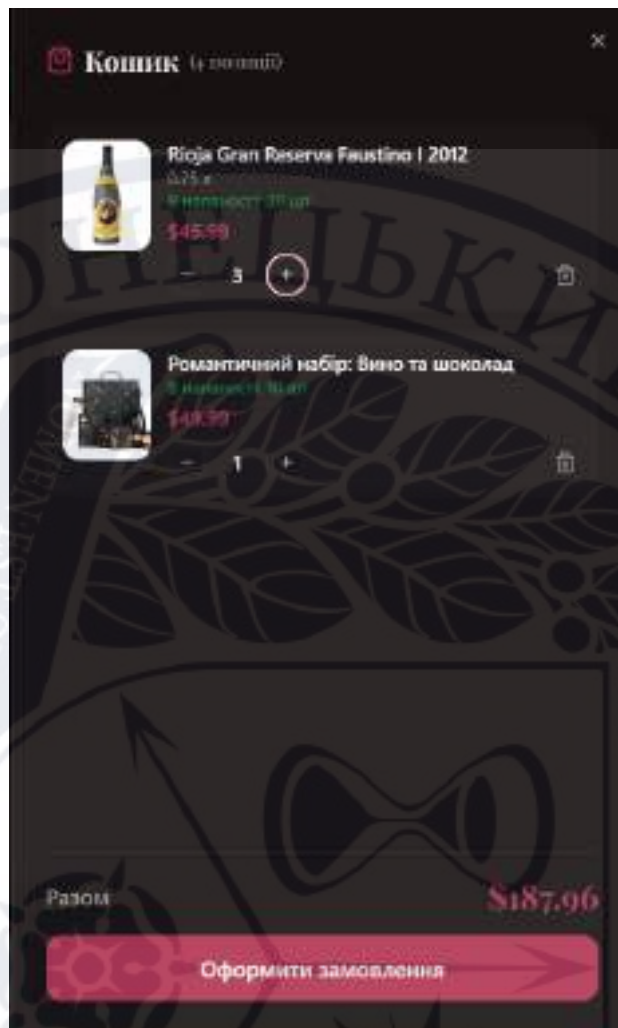


Рисунок 3.3 – Тестування функціоналу кошика

Особливу увагу було приділено тестуванню системи оформлення замовлення. Під час перевірки оцінювалася робота:

- форми введення даних;
- вибору способу доставки;
- валідації інформації;
- збереження замовлення у базі даних.

Було проведено декілька тестових сценаріїв із використання різних способів доставки:

- «Нова пошта»;
- самовивіз.

У процесі тестування система коректно відображала додаткові поля введення інформації залежно від обраного способу доставки, як показано на рис. 3.4 та 3.5.

Особисті дані

Ім'я та прізвище *
Микола Сковорода

Телефон *
0952041136

Email *
den641829@gmail.com

Доставка

☐ Кур'єр ☒ Нова Пошта ☐ Самовивіз

Місто *
Вінниця

Адреса Нової Пошти *
Вул. Академіка Янгеля 75

Номер відділення / поштового *
11

Рисунок 3.4 – Тестування оформлення замовлення «Новою поштою»

Особисті дані

Ім'я та прізвище *
Микола Сковорода

Телефон *
0952041136

Email *
den641829@gmail.com

Доставка

☐ Кур'єр ☐ Нова Пошта ☒ Самовивіз

Оберіть адресу самовивізу *

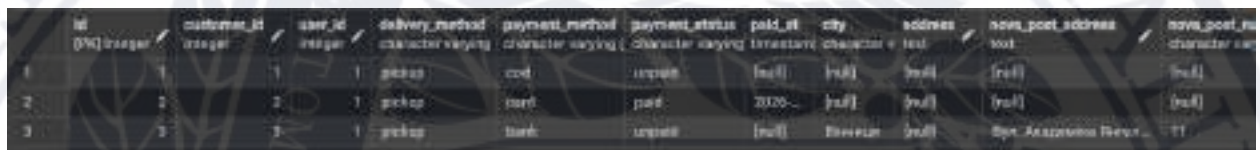
м. Вінниця, просп. Коцюбинського, 32

Рисунок 3.5 – Тестування оформлення замовлення «Самовивіз»

Після підтвердження покупки frontend частина надсилала HTTP-запит до backend сервера, а інформація про замовлення автоматично записувалася у базу даних PostgreSQL. Було перевірено:

- створення запису у таблиці orders;
- правильність SQL-запитів;
- коректність збереження інформації.

Після підтвердження покупки, було встановлено, що всі дані успішно передаються та зберігаються без помилок, як зображено на рис 3.6.



id	customer_id	user_id	delivery_method	payment_method	payment_status	paid_at	city	address	new_post_address	new_post_n
(PK) integer	integer	integer	character varying	character varying	character varying	timestamp	character	text	text	character varying
1	1	1	pickup	cod	unpaid	[null]	[null]	[null]	[null]	[null]
2	2	2	pickup	card	paid	2020-...	[null]	[null]	[null]	[null]
3	3	3	pickup	bank	unpaid	[null]	Вінниця	[null]	Бул. Академіка Печер...	77

Рисунок 3.6 – збереження замовлення у PostgreSQL

Окрему увагу було приділено тестуванню оформлення замовлення у особистий кабінет. Після оформлення замовлення система автоматично надсилала лист користувачу із підтвердженням покупки та інформацією про доставку. Було перевірено:

- правильність формування повідомлення;
- коректність відображення даних;
- швидкість надсилання листа.

Було встановлено, що система стабільно виконує ці операції без затримок та помилок, як показано на рис. 3.7.

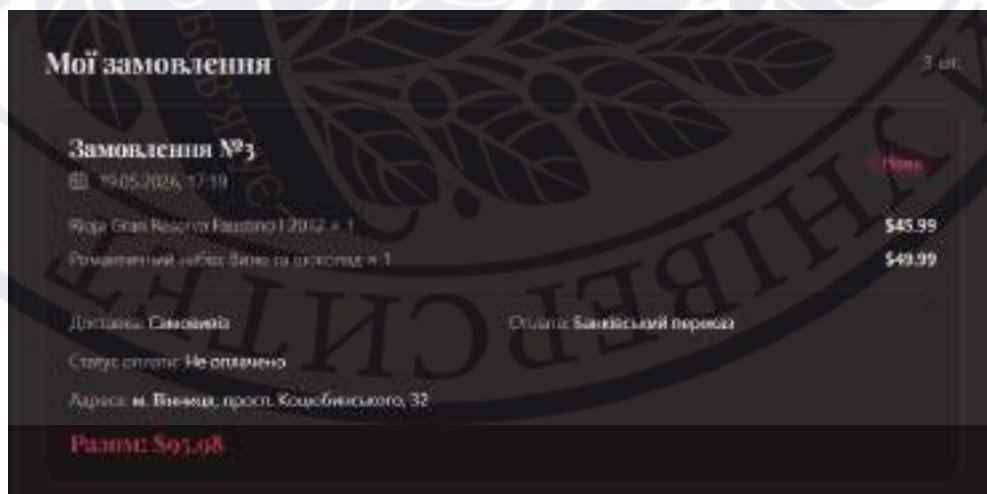


Рисунок 3.7 – Повідомлення в особистому кабінеті про оформлення замовлення

Для перевірки адаптивності вебзастосунку тестування виконувалося на різних типах пристроїв:

- персональному комп'ютері;
- планшеті;
- смартфоні.

Було встановлено, що вебсайт коректно адаптується під різні розміри екранів, а всі елементи інтерфейсу залишаються зручними для користувача, як на рис. 3.8 та 3.9.

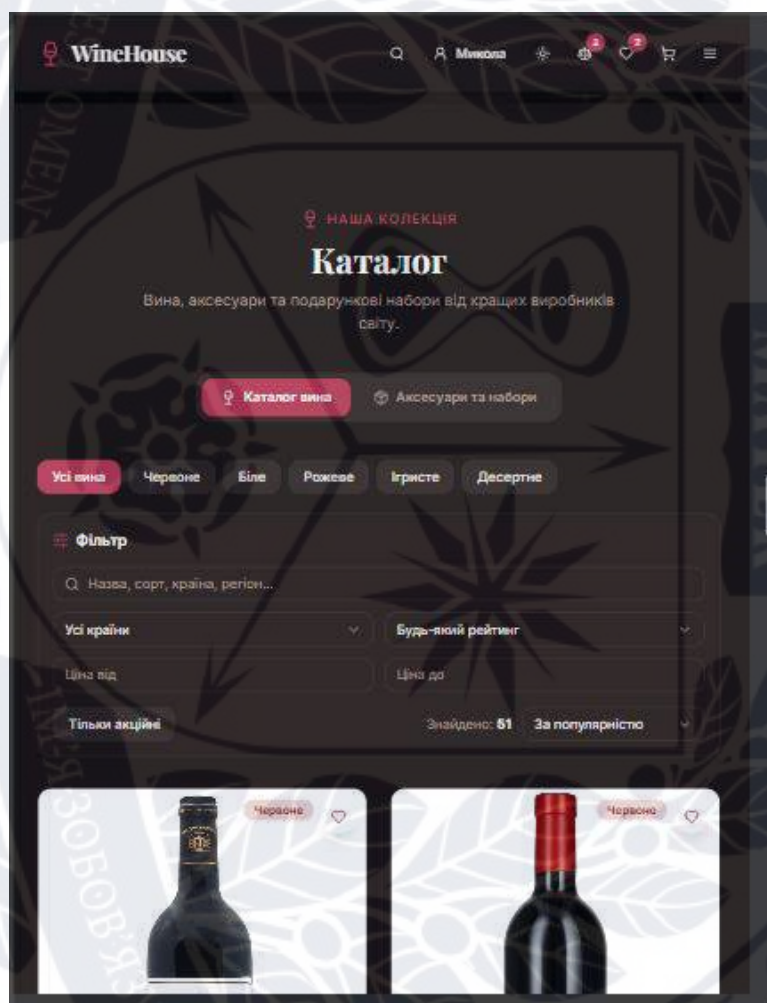


Рисунок 3.7 – Тестування на планшеті

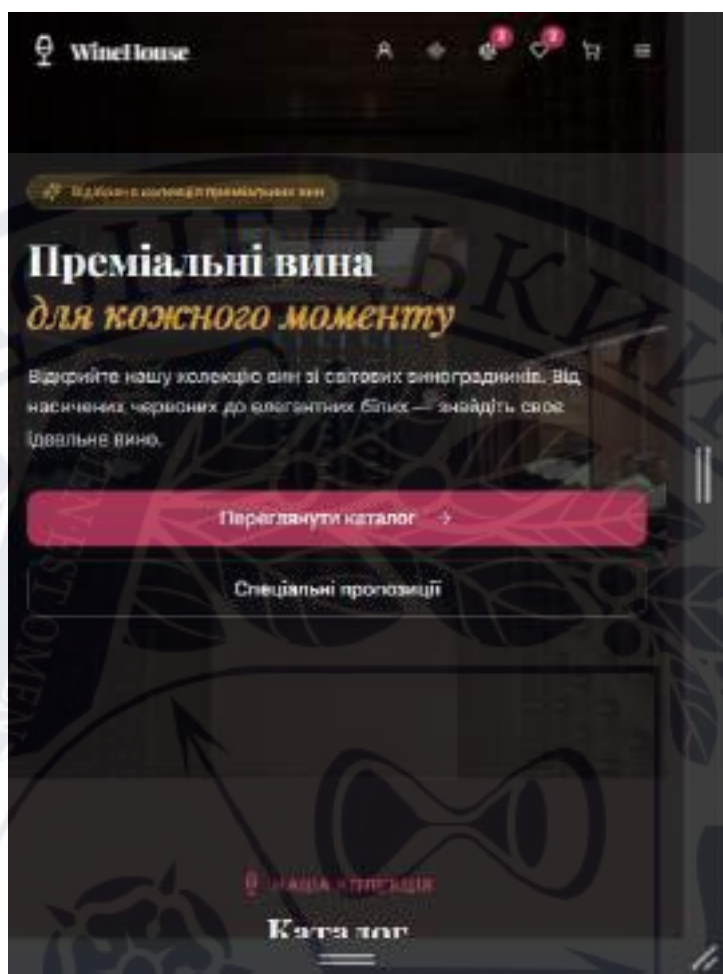


Рисунок 3.8 – Тестування на мобільному телефоні

У процесі тестування також оцінювалася швидкість роботи вебресурсу. Вебзастосунок швидко завантажує сторінки, а динамічне оновлення інформації через React дозволяє уникнути повного перезавантаження сторінок.

За результатами проведеного тестування було встановлено, що:

- усі основні функції вебзастосунку працюють коректно;
- система стабільно взаємовіє із базою даних;
- адаптивний інтерфейс правильно відображається на різних пристроях;
- процес оформлення замовлення працює без помилок.

Таким чином, проведене функціональне тестування підтвердило правильність реалізації основних модулів вебзастосунку та відповідність створеного інтернет-магазину сучасним вимогам електронної комерції.

Важливим етапом перевірки стала оцінка UX/UI дизайну вебресурсу. Під час тестування аналізувалися:

- зручність навігації;
- читабельність тексту;
- логічність структури сторінок;
- швидкість виконання дій;
- зручність оформлення замовлення.

Було встановлено, що інтерфейс вебсайту є інтуїтивно зрозумілим та не перезавантажений зайвими елементами. Основні функції вебресурсу доступні користувачу у декілька кліків, що позитивно впливає на користувацький досвід.

Для покращення UX/UI дизайну під час розробки використовувалися:

- плавні анімації;
- hover-ефекти;
- сучасна кольорова гама;
- адаптивні кнопки;
- динамічне оновлення елементів інтерфейсу.

Такі рішення дозволили зробити вебзастосунок більш сучасним та привабливим для користувача, як показано на рис. 3.9.

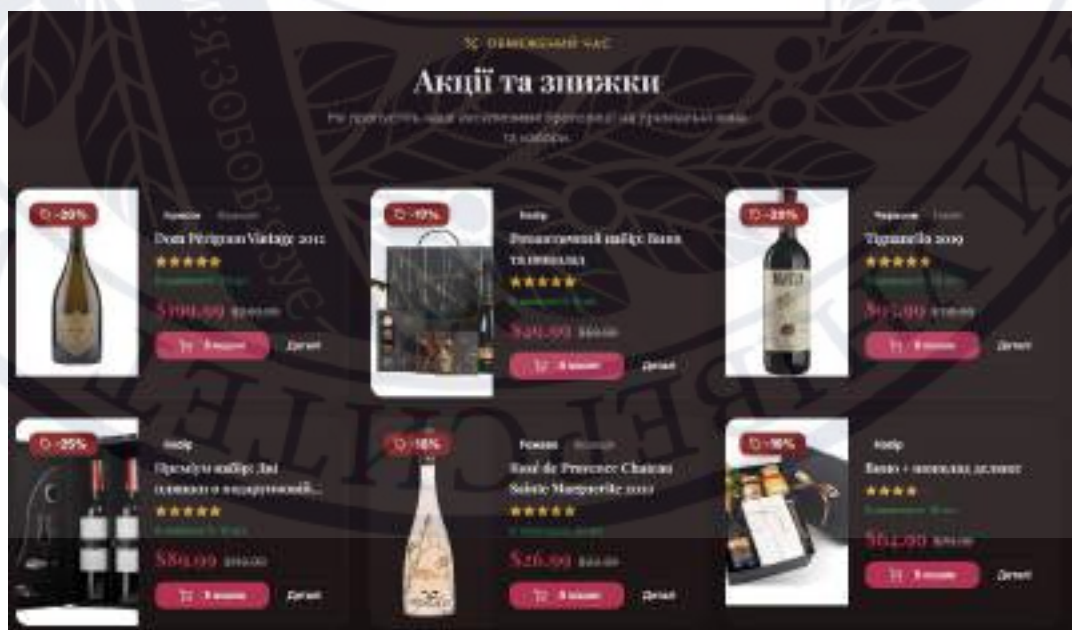


Рисунок 3.9 - UX/UI дизайн вебзастосунку

Під час тестування було перевірено швидкість взаємодії користувача із системою. Вебзастосунок швидко реагує на дії користувача:

- відкриття сторінок;
- додавання товарів до кошика;
- перехід між сторінками;
- оформлення замовлення.

Використання React дозволило реалізувати динамічне оновлення сторінок без повного перезавантаження, що позитивно впливає на продуктивність вебресурсу.

Окремо проводилося тестування сторінки оформлення замовлення на мобільних пристроях. Було встановлено, що форма коректно адаптується під вертикальний формат екрана, а всі поля введення залишаються зручними для користувача, як показано на рис. 3.10.

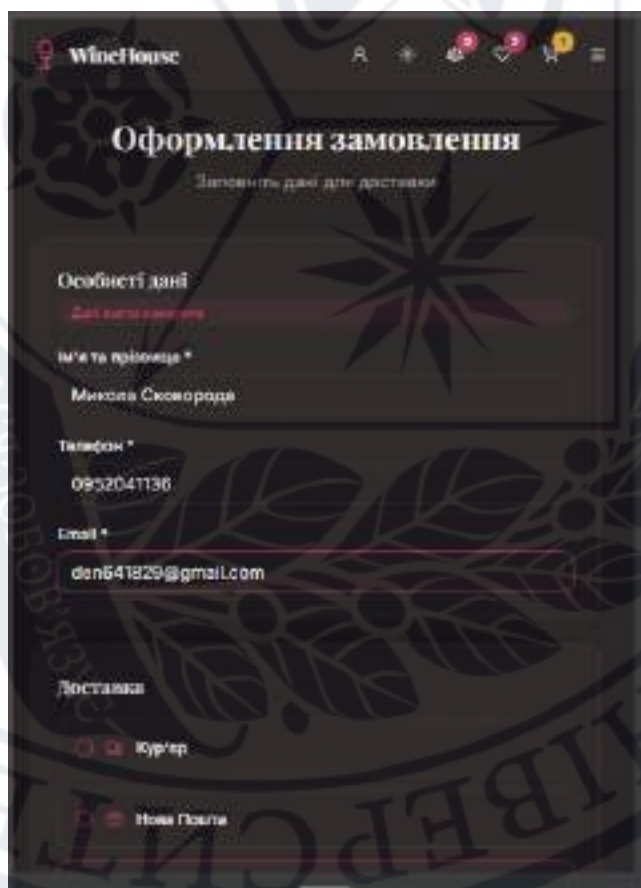


Рисунок 3.10 – Мобільна версія Checkout сторінки

Під час перевірки UX/UI дизайну також оцінювалась кольорова схема вебсайту. Використання темної кольорової гами у поєднанні з бордовими та

золотистими елементами дозволило створити стильний інтерфейс, який відповідає тематиці інтернет-магазину вина.

У результаті проведеного тестування було встановлено, що:

- вебзастосунок коректно працює на різних типах пристроїв;
- адаптивний дизайн забезпечує комфортне використання вебресурсу;
- UX/UI інтерфейс є сучасним та інтуїтивно зрозумілим;
- сторінки швидко завантажуються;
- навігація вебсайту є зручною для користувача.

Таким чином, перевірка адаптивності та UX/UI дизайну підтвердила, що реалізований вебзастосунок відповідає сучасним вимогам веброзробки та забезпечує комфортну взаємодію користувача із системою на різних типах пристроїв.

3.2. Аналіз результатів тестування вебзастосунку

Після проведення тестування вебзастосунку було виконано аналіз отриманих результатів, що дозволило оцінити якість реалізації системи, ефективність роботи основних функціональних модулів та відповідність вебресурсу сучасним вимогам до інтернет-магазинів. Аналіз результатів тестування є важливим етапом розробки, оскільки саме він дозволяє визначити сильні сторони вебзастосунку, виявити можливі недоліки та оцінити загальну стабільність роботи системи.

У результаті проведеного функціонального тестування було встановлено, що всі основні модулі вебзастосунку працюють коректно та забезпечують стабільну взаємодію користувача із системою. Каталог товарів успішно отримує інформацію із бази даних PostgreSQL та коректно відображає продукцію на сторінках вебресурсу.

Пошукова система також продемонструвала стабільну роботу. Пошук товарів виконувався швидко та без помилок, а результати автоматично

оновлювалися без перезавантаження сторінки. Це позитивно впливає на користувацький досвід та забезпечує швидкий доступ до необхідної інформації.

У процесі тестування кошика було встановлено, що система коректно:

- додає товари;
- оновлює кількість продукції;
- обчислює загальну суму замовлення;
- видаляє товари із кошика.

Кошик стабільно працював як на персональних комп'ютерах, так і на мобільних пристроях. Динамічне оновлення інформації через React дозволило забезпечити швидку взаємодію користувача із системою.

Особливу увагу було приділено аналізу роботи системи оформлення замовлення. У результаті тестування було встановлено, що:

- форма коректно перевіряє введені дані;
- система правильно обробляє різні способи доставки;
- інформація успішно передається до backend частини;
- замовлення зберігаються у базі даних.

У процесі аналізу результатів було встановлено, що база даних PostgreSQL забезпечує:

- стабільне зберігання інформації;
- швидке виконання SQL-запитів;
- коректну роботу зв'язків між таблицями;
- ефективну обробку замовлень.

Під час тестування адаптивності вебресурсу було перевірено роботу інтерфейсу на різних типах пристроїв. У результаті аналізу встановлено, що вебсайт коректно адаптується під:

- смартфони;
- планшети;
- ноутбуки;
- персональні комп'ютери.

Меню навігації, каталог товарів та сторінка оформлення замовлення автоматично змінюють структуру відповідно до розміру екрана. Позитивні результати було отримано й під час перевірки UXІ дизайну вебзастосунку. Інтерфейс вебресурсу є:

- інтуїтивно зрозумілим;
- зручним для користувача;
- візуально привабливим;
- адаптованим під сучасні вимоги вебдизайну.

Використання мінімалістичного стилю, темної кольорової гами та плавних анімацій позитивно вплинуло на загальне сприйняття вебсайту користувачем.

Під час аналізу продуктивності вебресурсу було встановлено, що сторінки вебзастосунку швидко завантажуються та стабільно працюють навіть при багаторазовому оновленні інформації. Використання React та компонентного підходу дозволяє зменшити навантаження на браузер та покращити швидкість роботи інтерфейсу.

У результаті проведеного аналізу було визначено основні переваги реалізованого вебзастосунку:

- сучасний UX/UI дизайн;
- адаптивність інтерфейсу;
- стабільна робота frontend та backend частини;
- інтеграція з PostgreSQL;
- зручний функціонал оформлення замовлення;
- швидка взаємодія користувача із системою.

Таким чином, проведений аналіз результатів тестування підтвердив, що створений вебзастосунок інтерне-магазину вина відповідає сучасним вимогам веброзробки та забезпечує стабільну, швидку та зручну роботу системи. Реалізований вебресурс може бути використаний як повноцінний інтернет-магазин із підтримкою електронної комерції та адаптивного UX/UI дизайну.

ВИСНОВКИ

У результаті виконання роботи було розроблено адаптивний вебсайт інтернет-магазину вина з оптимізованим UX/UI дизайном, який забезпечує зручну взаємодію користувача із системою та підтримує основний функціонал електронної комерції. Під час виконання роботи було проведено аналіз сучасних вебтехнологій, досліджено принципи створення адаптивних вебзастосунків та реалізовано повноцінний вебресурс із використанням сучасних frontend і backend технологій.

У процесі розробки було створено frontend частину вебзастосунку із використанням HTML, CSS, JavaScript та React. Реалізовано головну сторінку вебсайту, каталог товарів, систему пошуку продукції, сторінку акційних товарів, кошик покупок та сторінку оформлення замовлення. Для забезпечення зручності користувача було створено сучасний адаптивний інтерфейс із оптимізованою навігацією, плавними анімаціями та responsive-дизайном.

Для серверної частини вебзастосунку використовувалися Node.js а Express.js.

Було реалізовано REST API для взаємодії frontend частини із базою даних PostgreSQL. У процесі роботи також було створено структуру бази даних для зберігання інформації про товари, замовлення, кошик та користувачів. Реалізована база даних забезпечує стабільне зберігання інформації та ефективну обробку SQL-запитів.

Однією з важливих складових роботи стала реалізація адаптивності вебресурсу. Вебсайт коректно працює на персональних комп'ютерах, ноутбуках, планшетах та смартфонах. Для створення адаптивного інтерфейсу використовувалися Flexbox, CSS Grid та media queries, що дозволило забезпечити автоматичне налаштування структури сторінок залежно від розміру екрана користувача.

Під час розробки особлива увага приділялася UX/UI дизайну вебзастосунку. Було реалізовано мінімалістичний сучасний інтерфейс, який

забезпечує інтуїтивно зрозумілу навігацію та комфортне використання вебресурсу. Для покращення користувацького досвіду використовувалися hover-ефекти, плавні переходи, адаптивне меню навігації та динамічне оновлення інформації без перезавантаження сторінок.

У процесі виконання дипломної роботи також було реалізовано систему оформлення замовлення із підтримкою декількох способів доставки. Користувач має можливість додавати товари до кошика, змінювати кількість продукції, оформлювати покупку та отримувати email-повідомлення після підтвердження замовлення. Інформація про замовлення автоматично передається до бази даних PostgreSQL та обробляється серверною частиною вебзастосунку.

Після завершення розробки було проведено тестування вебресурсу. Під час тестування перевірялася робота каталогу товарів, кошика, системи оформлення замовлення, REST API, адаптивності інтерфейсу та швидкості роботи вебзастосунку. У результаті тестування було встановлено, що вебсайт стабільно працює на різних типах пристроїв та забезпечує коректну взаємодію користувача із системою.

Практичним результатом дипломної роботи є створення сучасного адаптивного вебзастосунку інтернет-магазину вина, який відповідає сучасним вимогам веброзробки та може використовуватися як повноцінний вебресурс електронної комерції. Реалізований вебсайт демонструє ефективність використання сучасних технологій frontend і backend розробки, а також практичне застосування принципів UX/UI дизайну під час створення адаптивних вебзастосунків.

У перспективі подальшого розвитку вебресурсу можливе розширення функціоналу шляхом реалізації системн авторизації користувачів, адміністративної панелі, інтеграції онлайн-оплати, системи відгуків та рейтингу товарів, а також підключення API служб доставки та персоналізованих рекомендацій продукції.

Таким чином, поставлена мета дипломної роботи була досягнута, а всі основні завдання успішно виконані.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Карпенко М. Ю. Електронна комерція : конспект лекцій для студентів усіх форм навчання першого (бакалаврського) рівня вищої освіти спеціальності 073 – Менеджмент / М. Ю. Карпенко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2021. – 5 с.
2. Топ-20+ інструментів для власника інтернет-магазину URL: <https://solomono.net/uk/top-20-instrumentiv-dlja-vlasnika-internet-magazinu-a-674.html> (дата звернення: 25.04.2026)
3. Франків Р. Б. Дітковський Д. А. Вплив економічних чинників на роботу UX/UI дизайнерів над цифровими сервісами. Культура і сучасність : альманах. 2023. № 2. 65 с.
4. Гребенюк І. В. Адаптивний та інклюзивний дизайн 2025 – 24 с.
5. Пустюльга С. І., Самчук В. П. технології вебдизайну. Луцьк, 2022. С. 267 - 270.
6. Болотіна, В. В., Чижмотря, О. Г., & Чижмотря, О. В. (2019). Перспективи використання технології Flexbox при адаптивній верстці вебсторінок. Технічна інженерія, (2(84), 70–76.
7. Тютюнник А.Н., адаптивний дизайн та оптимізація лендінгу. Умань, 2023 С. 298.
8. Буката Ю.В., сучасні технології для побудови веб-сайтів для перегляду на різних пристроях. Херсон 2020. С. 12-13.
9. Kalbach J. Designing web navigation: optimizing the user experience. Sebastopol, CA : O'Reilly Media, Inc., 2007. 456 с.
10. Lauer Ph., Zintel M. React - Einführung in ein Framework. 2020. 66 p. URL: https://www.researchgate.net/publication/357684856_React_Einfuehrung_in_ein_Framework (Last accessed: 27.04.2026).
11. Котенко Н. О., Жирова Т. О., Десятко А. М., Хорольська К. В. UX-дизайн інформаційної системи підприємства торгівлі. Київ, 2020 С. 67 – 74.

12. Веб-дизайн та UX/UI: як створити зручний інтерфейс? WhileWeb. URL: <https://whileweb.com/uk/blog/veb-dizajn-ta-uxui-yak-stvoriti-zruchnij-interfejs/> (дата звернення: 28.04.2026).
13. Навігація сайту: як оптимізувати, щоб клієнт «не загубився» на шляху до покупки. URL: <https://web-promo.ua/ua/blog/navihatsiya-saytu-yak-optimizuvaty-shchob-kliiyent-ne-zahubyvsya-na-shlyakhu-do-pokupky/> (дата звернення 28.04.2026)
14. Більць Д. Ю., Матюшин Л. С. вплив інтерактивних елементів веб-інтерфейсу на сприйняття інформації користувачем. Харків, 2025 С. 184 – 185.
15. Денесюк В. О. дослідження та розробка web-ресурсу персонального креативного простору з унікальним ui-дизайном. Київ, 2025 С. 1316.
16. Готинчан Т. І. Основи веброзробки: HTML і CSS. Чернівці, 2023 С. 72-84.
17. Рафальський В. опанування основ інтернет та вебтехнологій в курсі інформатики. 2023 С. 108-112.
18. Вступ до JavaScript URL: <https://uk.javascript.info/intro> (дата звернення 30.04.2026)
19. React JavaScript-бібліотека для створення користувацьких інтерфейсів URL: <https://uk.legacy.reactjs.org/> (дата звернення 01.05.2026)
20. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/en> (дата звернення 03.05.2026)
21. Веб-фреймворк Express (Node.js/JavaScript) URL: https://developer.mozilla.org/ru/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs (дата звернення 03.05.2026)
22. PostgreSQL: Documentation URL: <https://www.postgresql.org/docs/> (дата звернення 03.05.2026)
23. Опорний конспект лекцій «Проектування баз даних». Тернопіль, 2021 С. 10.
24. Йолкіна А. С. Особливості використання мови sql для обробки даних. С. 19-23.

25. pgAdmin – documentation URL: <https://www.pgadmin.org/docs/> (дата звернення 05.05.2026)
26. WINETIME. WINETIME. URL: <https://winetime.com.ua/ua> (дата звернення: 07.05.2026).
27. Vivino. Vivino. URL: <https://www.vivino.com/en/> (дата звернення 07.05.2026)
28. Власенко Б. Є. захист інформації у web-застосунку для обміну туристичним досвідом на основі бази відгуків про місця інтересу. Київ, 2025 С. 269-270.