

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ/факультет Навчально-науковий інститут економіки та бізнес-освіти
Кафедра Економіки та цифрового бізнесу
Спеціальність 122 «Комп'ютерні науки»
Форма навчання Денна

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Лалака Давида Сергійовича

(прізвище, ім'я, по батькові здобувача)

на тему Розробка Web-каталогу компанії-постачальника сонячних
батареї

(повна назва теми)

за матеріалами _____

(повна назва бази дослідження)

науковий керівник к.е.н., доцент Соловйова В.В.
(наук. ступінь, вчене звання) *(підпис)* *(прізвище, ініціали)*

Робота допущена до захисту в ЕК

Протокол засідання кафедри
від 09 червня 2025р. № 12

Завідувач кафедри _____
(підпис)

к.е.н., доцент
наук. ступінь, вчене звання

Радько В.М.
прізвище, ініціали

ЗАТВЕРДЖЕНО
Наказ Міністерства освіти і науки, молоді та
спорту України
29 березня 2012 року № 384

Форма № Н-9.01

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ ТА БІЗНЕС-ОСВІТИ
(повне найменування вищого навчального закладу)

Кафедра економіки та цифрового бізнесу
Освітній ступінь бакалавр
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри _____ **В.М. Радько**

“ 07 ” квітня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА ЗДОБУВАЧУ
Лалаку Давиду Сергійовичу

1. Тема роботи Розробка Web-каталогу компанії-постачальника сонячних батарей

науковий керівник роботи Соловійова Вікторія Володимирівна
затверджений наказом закладу вищої освіти від « 04 » квітня _____ 2025 р. № 224-ст (д/ф)
№ 151-ст (з/ф)

2. Строк подання здобувачем роботи 31.05.2025 р.

3. Зміст кваліфікаційної роботи бакалавра, об'єкт, предмет та мета дослідження:

Розділ 1 ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ WEB-КАТАЛОГУ КОМПАНІЇ-ПОСТАЧАЛЬНИКА СОНЯЧНИХ БАТАРЕЙ

Розділ 2 ПРОСКТУВАННЯ ТА РОЗРОБКА WEB-КАТАЛОГУ

Розділ 3 ФУНКЦІОНУВАННЯ ТА ЕЛЕМЕНТИ ІНТЕРФЕЙСУ WEB-КАТАЛОГУ

Об'єкт дослідження – процес створення інформаційної системи для відображення та керування переліком товарів у вигляді Web-каталогу.

Предмет дослідження – розробка Web-каталогу компанії-постачальника сонячних батарей.

Мета кваліфікаційної роботи бакалавра – розробка функціонального та зручного у використанні Web-каталогу для компанії-постачальника сонячних батарей, який забезпечуватиме користувачам швидкий доступ до інформації про продукцію, а також можливість звернення за консультацією через інтегровану форму зворотного зв'язку.

4. Дата видачі завдання 04.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	до 28.04.2025р.	25.04.2025
2	Підготовка розділу 2	до 16.05.2025р.	15.05.2025
3	Підготовка розділу 3	до 30.05.2025р.	29.05.2025
4	Реєстрація завершеної дипломної роботи	до 31.05.2025р.	30.05.2025
5	Отримання відгуку від наукового керівника	03-04.06.2025р.	04.06.2025
6	Отримання зовнішньої рецензії	05-06.06.2025р.	06.06.2025
7	Перевірка кваліфікаційної роботи на плагіат	02-09.06.2025р.	04.06.2025
8	Попередній захист кваліфікаційної роботи на кафедрі	03.06.2025р.	03.06.2025
9	Допуск кафедрою кваліфікаційної роботи до захисту	09.06.2025р.	09.06.2025
10	Підготовка студента до захисту в ЕК	до 17.06.2025р.	17.06.2025

Завдання підготував науковий керівник

_____ (підпис)

Соловйова В. В.

_____ (прізвище та ініціали)

Завдання одержав здобувач

_____ (підпис)

Лалак Д. С.

_____ (прізвище та ініціали)

РЕФЕРАТ

Робота містить 48 сторінок, 29 рисунків, 40 джерел, 2 додатки.

Об'єкт дослідження: процес створення інформаційної системи для відображення та керування переліком товарів у вигляді Web-каталогу.

Предмет дослідження: розробка Web-каталогу компанії-постачальника сонячних батарей.

Мета дослідження: розробка функціонального та зручного у використанні Web-каталогу для компанії-постачальника сонячних батарей, який забезпечуватиме користувачам швидкий доступ до інформації про продукцію, а також можливість звернення за консультацією через інтегровану форму зворотного зв'язку.

У результаті дослідження було вивчено особливості побудови Web-каталогу для компанії-постачальника сонячних батарей та визначено основні вимоги до такої системи. Було спроектовано інформаційну модель Web-каталогу, а також обрано відповідну архітектуру SPA-додатку, який був реалізований з використанням Vue.js, Node.js та MySQL. Створений Web-каталог має зручний інтерфейс та необхідний функціонал для наповнення та актуалізації асортименту товарів.

Область застосування: результати цієї роботи можуть бути використані в Web-ресурсах підприємств, що займаються реалізацією товарів технічного призначення, а також як основа для створення повноцінної платформи електронної комерції.

ІНФОРМАЦІЙНА СИСТЕМА, WEB-КАТАЛОГ, WEB-ІНТЕРФЕЙС,
WEB-РЕСУРС, СОНЯЧНІ БАТАРЕЇ, ЦИФРОВІЗАЦІЯ БІЗНЕСУ,
ЕЛЕКТРОННА КОМЕРЦІЯ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	7
ВСТУП	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ WEB-КАТАЛОГУ КОМПАНІЇ-ПОСТАЧАЛЬНИКА СОНЯЧНИХ БАТАРЕЙ	10
1.1 Актуальність розробки Web-ресурсів для продажу компонентів процесу сонячної генерації	10
1.2 Огляд існуючих рішень та підходів до їх реалізації	13
1.3 Вибір технологічного стеку розробки та архітектури Web- каталогу	18
Висновки до розділу 1	21
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА WEB-КАТАЛОГУ	23
2.1 Проєктування інформаційної моделі Web-каталогу	23
2.2 Реалізація архітектури Web-каталогу	29
2.3 Розробка окремих модулів Web-каталогу	31
Висновки до розділу 2	40
РОЗДІЛ 3 ФУНКЦІОНУВАННЯ ТА ЕЛЕМЕНТИ ІНТЕРФЕЙСУ WEB- КАТАЛОГУ	41.
3.1 Огляд функціоналу створеного Web-каталогу	41
3.2 Реалізація можливих сценаріїв взаємодії з Web-каталогом	45
3.3 Аналіз створеного Web-ресурсу та перспективи його подальшого розвитку	50
Висновки до розділу 3	53
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ	61

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – База Даних;

СКБД – Система Керування Базами Даних;

API – Application Programming Interface

CMS – Content Management System;

CRM – Customer Relationship Management;

CRUD – Create, Read, Update, Delete;

JWT – JSON Web Token;

REST – Representational State Transfer

SPA – Single Page Application

UI – User Interface

UX – User Experience

ВСТУП

В умовах активного розвитку ринку відновлюваних джерел енергії особливого значення набуває цифровізація бізнес-процесів компаній, що займаються постачанням компонентів сонячної генерації. Зараз спостерігається активний процес впровадження таких джерел енергії у приватному та промисловому секторах, що вимагає від виробників та постачальників необхідних комплектуючих використання сучасних підходів до подання інформації про свою продукцію. Тому ефективне представлення переліку товарів, забезпечення зручної та оперативної комунікації з клієнтами, автоматизація опрацювання запитів і управління асортиментом стають невід'ємними елементами конкурентоспроможного бізнесу. При цьому, для компаній, що постачають сонячні батареї та інші супутні компоненти сонячної генерації, наявність якісного Web-ресурсу стає необхідною умовою для виходу на нові ринки та збільшення клієнтської бази.

Слід зазначити, що в якості Web-ресурсу може виступати електронний каталог продукції, який потенційно може виконувати не тільки функцію демонстрації товарів та їх характеристик, а й стати засобом взаємодії між компанією та клієнтами. Такий каталог здатен забезпечити автоматизоване приймання замовлень та зворотний зв'язок з користувачами. Також, з урахуванням можливого подальшого розширення функціональності, він може бути інтегрований з платіжними та логістичними системами, перетворюючись на потужний засіб електронної комерції.

Об'єкт дослідження: процес створення інформаційної системи для відображення та керування переліком товарів у вигляді Web-каталогу.

Предмет дослідження: розробка Web-каталогу компанії-постачальника сонячних батарей.

Метою дослідження є розробка функціонального та зручного у використанні Web-каталогу для компанії-постачальника сонячних батарей, який забезпечуватиме користувачам швидкий доступ до інформації про продукцію, а

також можливість звернення за консультацією через інтегровану форму зворотного зв'язку.

В процесі дослідження вирішувалися наступні завдання:

- оцінити актуальність розробки Web-ресурсів для продажу компонентів процесу сонячної генерації;
- обґрунтувати вибір архітектури Web-додатку та сучасного технологічного стеку для розробки;
- розробити інформаційну модель яка відображає логіку функціонування Web-каталогу;
- реалізувати клієнтську та серверну частини Web-каталогу відповідно до обраної архітектури;
- провести аналіз створеного Web-ресурсу;
- оцінити перспективи подальшого розвитку створеного Web-ресурсу як засобу електронної комерції.

В процесі проведення дослідження використовувалися такі методики:

- аналіз предметної області;
- порівняльний аналіз;
- об'єктно-орієнтований підхід;
- моделювання з використанням UML;
- системний аналіз.

Практична значущість результатів дослідження полягає у створенні надійного, зручного та масштабованого Web-ресурсу, який можна використати як реальний інструмент електронної комерції. Реалізований Web-каталог може бути впроваджений у комерційну діяльність, сприяючи підвищенню продажів і поліпшенню взаємодії з клієнтами.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ WEB-КАТАЛОГУ КОМПАНІЇ-ПОСТАЧАЛЬНИКА СОНЯЧНИХ БАТАРЕЙ

1.1 Актуальність розробки Web-ресурсів для продажу компонентів процесу сонячної генерації

В умовах стрімкого розвитку відновлюваних джерел енергії особлива увага приділяється сонячній енергетиці як одному з найперспективніших напрямків [1-3]. Ефективне поширення компонентів сонячної генерації (сонячні панелі, інвертори, акумулятори та системи управління) неможливе без сучасного підходу до організації їх продажу. У цьому контексті розробка спеціалізованих Web-ресурсів набуває особливої актуальності, забезпечуючи зручний доступ до інформації, широке охоплення цільової аудиторії та автоматизацію бізнес-процесів [4].

Останніми роками спостерігається стійке зростання інтересу до «зеленої» енергетики як серед приватних споживачів, так і серед представників бізнесу. Цей попит стимулюють виклики, пов'язані з енергетичною безпекою, прагнення до зниження викидів вуглецю та зростання вартості традиційних енергоресурсів. Однак ефективне просування та реалізація відповідних компонентів потребують наявності якісних цифрових інструментів.

Web-ресурси, орієнтовані на продаж компонентів сонячної генерації, як правило, представляють собою сучасні онлайн-платформи, що об'єднують функції інтернет-магазину, інформаційного центру та консультаційної служби. При цьому, їх значення важко переоцінити, оскільки саме через такі цифрові канали здійснюється первинне знайомство користувача з продукцією, забезпечується можливість порівняння характеристик, отримання технічних рекомендацій та оформлення замовлення. З огляду на зростаючий попит на обладнання для сонячної генерації, зумовлений як економічними, так і екологічними факторами, наявність зручних, функціональних і надійних

інтернет-ресурсів стає стратегічно важливою умовою успішного просування таких рішень на ринку [4, 5].

Слід зазначити, що розвиток і поширення Інтернету кардинально змінили поведінку споживачів. Все більше клієнтів вважає за краще виконувати пошук інформації та здійснювати покупки онлайн, не вдаючись до послуг традиційних торгових точок. У сфері продажів сонячного обладнання це особливо актуально, оскільки купівля, як правило, пов'язана з аналізом безлічі технічних параметрів, необхідністю розрахувати необхідну потужність системи і підібрати сумісні компоненти. При цьому добре спроектований Web-ресурс здатний не тільки полегшити цей процес, а й підвищити рівень довіри клієнта до бренду, надаючи експертні поради та технічну підтримку. До основних функцій Web-ресурсів, орієнтованих на продаж компонентів сонячної генерації, можна віднести:

- надання повної інформації про технічні характеристики продукції;
- спрощення процесу підбору та замовлення обладнання;
- можливість демонстрації реальних кейсів, відгуків та рекомендацій;
- забезпечення якісної підтримки клієнтів і супровід продажів.

Однак, крім торговельних функцій, Web-платформи часто виконують також важливу освітню роль, пропонуючи користувачам доступ до статей, відеоматеріалів, інструкцій зі встановлення та експлуатації обладнання. Таким чином, вони формують лояльну аудиторію, готову інвестувати в стійкі енергетичні рішення, і стимулюють подальше зростання ринку.

Для компаній, що займаються постачанням компонентів сонячної генерації, розробка і впровадження власного Web-ресурсу відкриває нові можливості для розширення клієнтської бази. Онлайн-платформа дає змогу вийти за межі локального ринку і залучити клієнтів з інших регіонів, а також автоматизувати багато бізнес-процесів - від прийому та обробки замовлень до взаємодії з логістичними та платіжними системами. Крім того, такі ресурси забезпечують швидкий зворотний зв'язок і дають змогу оперативно реагувати на зміни попиту та поведінки користувачів. Завдяки вбудованим аналітичним інструментам власники Web-ресурсів отримують доступ до даних про

відвідуваність, вподобання покупців, ефективність маркетингових кампаній, що дає змогу оперативно коригувати стратегію розвитку [6, 7].

Таким чином, сучасний Web-ресурс може поєднувати в собі безліч важливих функцій. Зокрема він може представляти собою електронний каталог продукції з можливістю фільтрації, підбором параметрів і рекомендаціями, що значно полегшує вибір обладнання для різних потреб. Крім того він може містити вбудовані інтерактивні сервіси, як-от калькулятори розрахунку необхідних параметрів для підбору обладнання, що робить цей процес точнішим і зручнішим. При цьому, впровадження підтримки CRM-систем і платіжних сервісів автоматизує взаємодію з клієнтами, підвищуючи загальну ефективність бізнесу [8, 9].

Окрему увагу при розробці таких Web-платформ слід приділяти користувацькому інтерфейсу та його адаптивності. Оскільки доступ до сайту може здійснюватися з різних пристроїв - від стаціонарних комп'ютерів до смартфонів, - важливо забезпечити коректне відображення контенту та зручну навігацію незалежно від технічних умов. Також слід враховувати різний рівень підготовки користувачів: від фахівців у галузі енергетики до звичайних громадян, зацікавлених у встановленні сонячної панелі на власній ділянці. Успішний Web-ресурс має бути одночасно простим та інформативним, інтуїтивно зрозумілим і функціонально повним.

Використання сучасних Web-технологій сприяє створенню іміджу інноваційного та надійного бренду, що особливо важливо в умовах високої конкуренції та швидкого розвитку галузі. Компанії, які ігнорують цифровізацію, ризикують втратити зв'язок з аудиторією і поступитися позиціями більш гнучким і технологічним гравцям ринку.

Таким чином, розробка Web-ресурсів для продажу компонентів процесу сонячної генерації є актуальною складовою сучасного бізнесу у сфері відновлюваної енергетики. Такі ресурси сприяють ефективному поширенню екологічно чистих технологій, покращують взаємодію між виробниками та споживачами, а також формують культуру сталого споживання. В умовах

глобальної цифровізації всіх сфер життя саме інтернет-платформи стають ключовими каналами просування та реалізації інноваційних рішень, від яких залежить екологічне та економічне майбутнє суспільства [10].

1.2 Огляд існуючих рішень та підходів до їх реалізації

На сучасному етапі розвитку цифрових технологій існує широкий спектр програмних рішень, що дають змогу створювати ефективні Web-ресурси, орієнтовані на продаж спеціалізованого обладнання, зокрема й сонячних панелей. Підхід до створення таких ресурсів багато в чому залежить від завдань бізнесу, масштабу діяльності, цільової аудиторії та наявних технічних ресурсів.

Одним із найпопулярніших способів реалізації торгових Web-платформ є використання готових CMS-систем (Content Management System), як-от WordPress (рис. 1.1) із плагіном WooCommerce, OpenCart, Joomla з розширенням VirtueMart [11, 12].

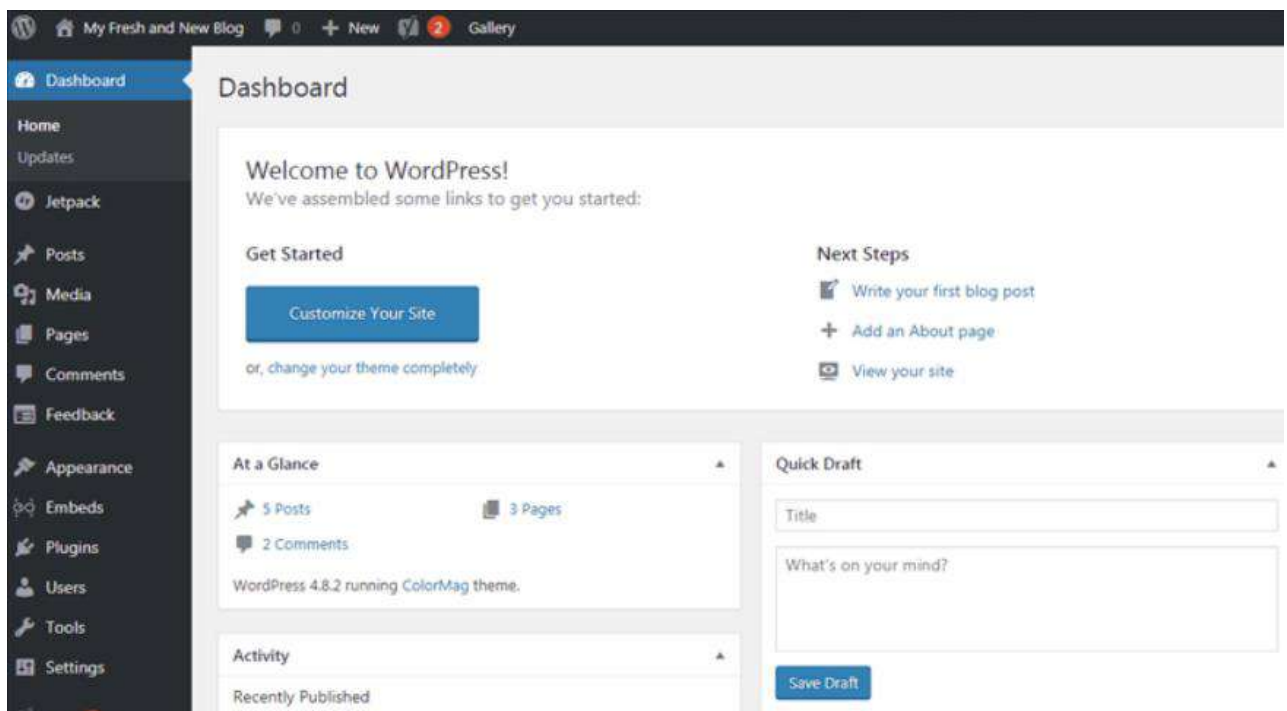


Рис. 1.1. Адміністративна панель CMS WordPress

Ці системи дають змогу швидко розгортати інтернет-магазини з мінімальними витратами, забезпечуючи широкий функціонал, необхідний для управління товарами, замовленнями та клієнтськими даними. Вони надають зручні інтерфейси для адміністраторів, дають змогу легко додавати картки товарів із детальними характеристиками, організувати фільтрацію за параметрами та підключати платіжні системи. Такий підхід підходить для малого та середнього бізнесу, який прагне швидко вийти на ринок з мінімальними технічними ризиками [13].

При цьому, для компаній з вищими вимогами до індивідуалізації та масштабування проекту, при створенні Web-ресурсів доцільно використовувати фреймворки, як-от Vue.js, Laravel, Node.js та інші. Ці фреймворки дозволяють будувати Web-ресурси враховуючи специфічні потреби замовника, включно з інтеграцією з внутрішніми системами обліку, гнучким налаштуванням бізнес-логіки та високим рівнем безпеки. Такі рішення вимагають наявності кваліфікованої команди розробників, але забезпечують більшу свободу в проектуванні користувальницького інтерфейсу, складових модулів та аналітичних інструментів. Ці інструменти часто розробляються індивідуально, з використанням JavaScript-бібліотек (наприклад, Vue.js або React) і сторонніх API, що дає змогу адаптувати сайт під технічно орієнтовану аудиторію і підвищити залученість користувачів [14-16].

Таким чином, індивідуальна розробка Web-ресурсу являє собою більш гнучкий і технологічно доцільний підхід у тих випадках, коли готові рішення обмежують можливості кастомізації, масштабування або інтеграції зі сторонніми сервісами. Такий підхід дає змогу врахувати не тільки специфічні бізнес-потреби, а й забезпечити користувачеві якіснішу взаємодію з цифровим продуктом.

Розробка фронтенду, тобто клієнтської частини Web-додатку, спрямована на створення інтуїтивно зрозумілого, сучасного інтерфейсу, через який користувач отримує доступ до товарів, фільтрів, калькуляторів розрахунку та інших інтерактивних елементів. Для цього найчастіше використовують сучасні

JavaScript-фреймворки, такі як Vue.js, React або Angular. Ці технології дають змогу будувати односторінкові додатки (SPA - Single Page Application), що забезпечують швидке завантаження, динамічне оновлення даних без перезавантаження сторінки та зручну навігацію. При цьому важливо відзначити, що одним із ключових чинників успіху Web-ресурсу є якісно реалізований інтерфейс користувача (UI), який здатен забезпечити гарний користувацький досвід (UX). Навіть за наявності широкого асортименту обладнання та потужної серверної частини, незручна навігація, перевантажені форми або заплутаний процес оформлення замовлення можуть відштовхнути потенційного клієнта і знизити конверсію сайту [17-19].

На рисунках 1.2 та 1.3 представлені Web-каталоги сонячних батарей двох популярних постачальників компонентів процесу сонячної генерації. SOLAR STATUM GROUP (solarstatumgroup.com.ua) на рис. 1.2 та Solar Market (solarmarkets.com.ua) на рис. 1.3.



Рис. 1.2. Web-каталог сонячних батарей SOLAR STATUM GROUP (solarstatumgroup.com.ua)

The image shows the Solar Markets website interface. At the top, there is a logo for Solar Markets (сонячні електростанції) and contact information: phone numbers 067-941-77-20, 063-941-77-22, and 066-941-77-22, along with a working hours schedule (Пн-Сб: з 8:30 до 18:00; Нд: з 10:00 до 14:00). A search bar is present with the text 'пошук товарів'. Below the navigation bar, the page title is 'Сонячні панелі'. The main content area displays a grid of solar panels, with a 'Сонячні панелі' category highlighted. A sidebar on the left allows filtering by price (0 to 11409 UAH), availability, and brand (AXIOMA, Canadian Solar, Inter Energy, Ja Solar, Jinko Solar, LEAPTON SOLAR, Longi Solar, Luxen). A promotional banner for 'HPBC cells' with a 22.5% discount is visible on the right.

**Рис. 1.3. Web-каталог сонячних батарей Solar Market
(solar-markets.com.ua)**

Обидва ресурси представляють функціональні Web-каталоги, проте відрізняються за структурою, стилістикою інтерфейсу і ступенем зручності для користувача. Зокрема, на сайті SOLAR STATUM GROUP (рис. 1.2) переважає суворий діловий стиль, а інтерфейс орієнтований на професійних користувачів, акцент зроблено на технічні параметри, детальні характеристики обладнання та наявність інженерної підтримки. При цьому сайт Solar Market (рис. 1.3) демонструє більш доброзичливий користувацький інтерфейс, який передбачає яскраві візуальні елементи та швидкий доступ до популярних категорій товарів, що може зробити його більш доступним для широкої аудиторії. Такі відмінності підкреслюють важливість продуманого UI/UX під час проектування Web-застосунків. Правильний баланс між інформаційною насиченістю та зручністю

сприйняття безпосередньо впливає на ефективність взаємодії користувачів з платформою. Тому при індивідуальній розробці фронтенда важливо проводити аналіз цільової аудиторії, тестувати інтерфейсні рішення й адаптувати Web-ресурс під реальні сценарії користувацької поведінки [20, 21].

На стороні бекенда здійснюється логіка обробки даних, управління базами даних, обробка замовлень, робота з авторизацією користувачів та інтеграція із зовнішніми сервісами. При індивідуальній розробці серверної частини важливо продумати архітектуру: розподіл ролей користувачів (покупці, адміністратори, менеджери), механізми безпеки (шифрування, аутентифікація та ін.), роботу з API сторонніх платіжних і логістичних систем. Велике значення має проектування та оптимізація бази даних, де можуть зберігатися характеристики товарів, дані про замовлення, відгуки та логування активності користувачів. Як правило, використовуються СКБД типу PostgreSQL, MySQL або MongoDB [22, 23].

Для зручного керування базами даних на етапі розробки та підтримки проекту часто застосовується інструмент phpMyAdmin, який надає графічний інтерфейс для роботи з MySQL-сервером. Графічний інтерфейс phpMyAdmin показаний на рисунку 1.4.

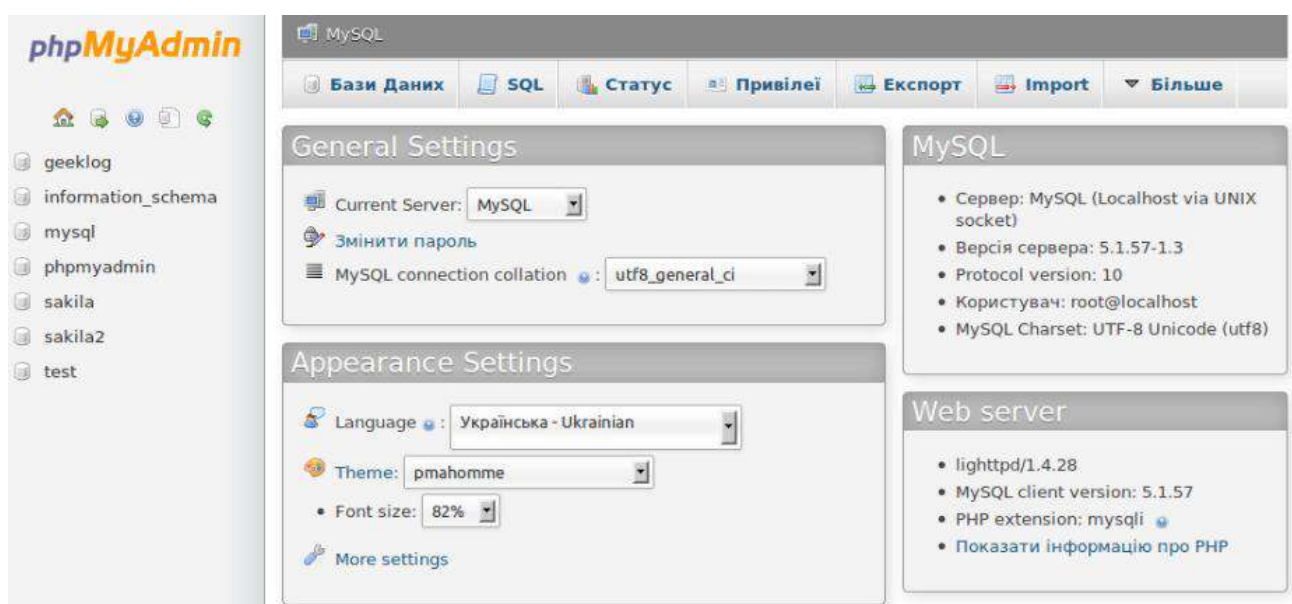


Рис. 1.4. Графічний інтерфейс phpMyAdmin

За допомогою такого графічного інтерфейсу можна швидко створювати таблиці, виконувати запити, керувати правами доступу, експортувати та імпортувати дані, що значно спрощує взаємодію розробників і адміністраторів з системою зберігання інформації.

Слід зазначити, що важливою частиною індивідуальної розробки може бути також реалізація зручної адміністративної панелі, в якій адміністратор може керувати каталогом товарів, відстежувати статистику, редагувати опис, завантажувати зображення, обробляти замовлення і формувати звіти. Така панель, побудована на тому ж фреймворку, що й основний застосунок, забезпечує повну автономію компанії в управлінні ресурсом без залучення сторонніх технічних фахівців.

Таким чином, індивідуальна розробка Web-ресурсу надає можливість створити не просто електронний каталог, а повноцінну цифрову платформу, повністю адаптовану під потреби бізнесу та вимоги галузі. Особливо в контексті продажу спеціального обладнання, де велике значення має технічне консультування, робота з інженерами та подальша підтримка. В таких умовах наявність гнучкої, надійної та функціональної системи може надати суттєву конкурентну перевагу.

1.3 Вибір технологічного стеку розробки та архітектури Web-каталогу

В умовах стрімкого зростання кількості онлайн-платформ та інтенсивної цифровізації торгівлі розробка Web-каталогів стає невід'ємною частиною електронної комерції. У зв'язку з цим одним із найважливіших етапів проектування Web-ресурсу є вибір технологічного стека - сукупності програмних засобів, інструментів і мов програмування, які використовуються для реалізації клієнтської та серверної частини Web-додатку. Від цього безпосередньо залежать такі критичні параметри, як масштабованість платформи, продуктивність під час високих навантажень, стійкість до збоїв, безпека зберігання і передавання даних, а також простота підтримки та внесення

змін до системи. На практиці Web-каталоги, як правило, реалізуються за принципом поділу логіки на фронтенд і бекенд частини (рис. 1.5), які взаємодіють між собою через API [24, 25].

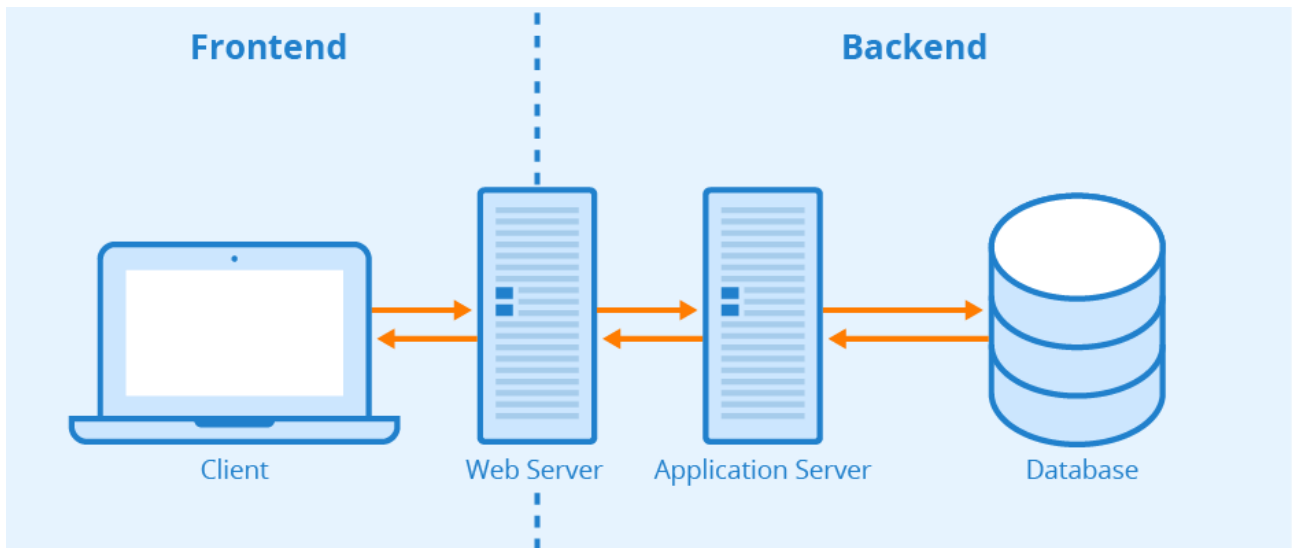


Рис. 1.5. Загальна схема взаємодії фронтенд і бекенд частини Web-додатку

Одним з ефективних і поширених сучасних підходів до вирішення цих завдань є використання зв'язки Node.js на стороні сервера (бекенд) і Vue.js на стороні клієнта (фронтенд). Така комбінація забезпечує високу продуктивність, кросплатформність і гарну масштабованість проєкту. При цьому, використання JavaScript як на фронтенді, так і на бекенді дає змогу уніфікувати підходи до написання коду. Ця зв'язка підтримується великим співтовариством, активно розвивається і надає безліч готових рішень, що робить її особливо привабливою для індивідуальних і комерційних проєктів, включно з Web-каталогами, пов'язаними з продажем технічно складної продукції, такої як компоненти сонячної генерації.

Node.js являє собою середовище виконання JavaScript на стороні сервера, яке особливо підходить для Web-додатків, у яких потрібно швидко обробляти безліч паралельних запитів - наприклад, під час роботи з товарами, замовленнями, авторизацією та зовнішніми API (платіжні системи, служби

доставки та ін.). Node.js дає змогу реалізувати легкий і продуктивний сервер, що забезпечить швидку реакцію на дії користувачів.

Крім високої продуктивності, Node.js вирізняється якісним екосистемним оточенням - завдяки менеджеру пакетів npm розробники отримують доступ до тисяч готових модулів. Це прискорює процес розробки, дає змогу гнучко масштабувати архітектуру проєкту та зменшити витрати часу на реалізацію типових функцій (аутентифікація, валідація форм тощо). Зокрема, серверна частина на Node.js може бути побудована з використанням потужного фреймворка Express.js, який має мінімалістичний і зрозумілий інтерфейс для роботи з HTTP-запитами, та надає змогу організувати чітку структуру проєкту, розділити функціональність за модулями та реалізувати API для взаємодії з клієнтською частиною. Він також дозволяє зручно працювати з базами даних і зовнішніми API, що робить оптимальним його використання для створення гнучких і надійних серверних рішень [26, 27].

Vue.js - це JavaScript-фреймворк для створення користувацьких Web-інтерфейсів. Він має інтуїтивно зрозумілий синтаксис і модульну архітектуру, що дає змогу будувати як невеликі компоненти, так і складні односторінкові додатки (SPA - Single Page Applications). Цей фреймворк доцільно використати для створення фронтенд-частини Web-каталогу, оскільки він забезпечує швидке завантаження Web-інтерфейсу, динамічне оновлення даних без повного перезавантаження сторінки, а також гнучку роботу з Web-компонентами. Зокрема, завдяки реактивності та компонентній структурі, Vue.js дає змогу ефективно реалізувати фільтри товарів, форми пошуку та інші елементи взаємодії з користувачем. При цьому, існуюча супутня розвинута екосистема дозволить швидко побудувати клієнтську частину Web-каталогу [28-31].

Поєднання Node.js і Vue.js дає змогу створити якісний користувацький Web-інтерфейс та спрощує подальше масштабування проєкту, знижуючи складність технічної підтримки. Такий стек вже використовується в існуючих комерційних проєктах, для яких важлива швидкість виходу на ринок, а також безпека і зручність користувацької взаємодії [32].

Одним із ключових завдань при розробці Web-каталогу є необхідність забезпечення надійного й організованого зберігання інформації. Зокрема, в процесі роботи Web-каталогу можуть оброблятися та зберігатися різні типи даних, наприклад дані про товари та їхні характеристики, облікові записи користувачів та ін.. Для ефективної роботи з цими даними потрібна надійна та продуктивна система управління базою даних, яка буде здатна обробляти безліч паралельних запитів, забезпечувати цілісність даних і мати зручні засоби доступу до них. Найчастіше в таких проєктах використовують СКБД MySQL. Вона є надійним рішенням з широкими можливостями для масштабування. MySQL забезпечує високу швидкість обробки типових операцій і є досить простою в адмініструванні [33-35].

На етапі розробки, тестування і прототипування часто використовується інструмент phpMyAdmin, який надає зручний веб-інтерфейс для управління базами даних MySQL. З його допомогою можна швидко створювати і модифікувати таблиці, керувати зв'язками між сутностями, виконувати SQL-запити, переглядати структуру даних і виконувати резервне копіювання та перенесення даних. Використання phpMyAdmin також полегшує виявлення помилок у структурі бази даних і значно прискорює процес відлагодження створюваного Web-каталогу [36].

Таким чином, вибір технологічного стека є критично важливим етапом при розробці Web-каталогу, від якого безпосередньо залежить надійність, масштабованість і ефективність усієї системи. Комбінація Node.js на стороні сервера і Vue.js на стороні клієнта забезпечує сучасну, гнучку та продуктивну архітектуру, що дає змогу реалізувати як функціональний каталог товарів, так і можливість обробки замовлень. Доповнює стек використання MySQL, яка виконує надійне зберігання структурованих даних, забезпечуючи високу продуктивність та можливість використання зручних інструментів для адміністрування БД. Загалом, обраний стек дає змогу створити якісний Web-ресурс, що буде відповідати сучасним принципам електронної комерції та потребам ринку, що динамічно розвивається.

Висновки до розділу 1

Розробка якісного Web-каталогу постачальника сонячних батарей охоплює як технічні аспекти, так і особливості користувацької взаємодії. При цьому, в сучасних умовах, такі Web-ресурси є не тільки інструментом електронної комерції, а й важливим засобом інформаційної підтримки клієнтів. Тому важливу роль відіграє якісний UX/UI-дизайн, від якого залежить зручність інтерфейсу для користувачів.

Вибір архітектури та технологічного стека має ключове значення при створенні Web-каталогу. При цьому стек Node.js і Vue.js дає змогу створювати продуктивні та масштабовані Web-додатки з сучасною структурою та адаптивним дизайном. Також доцільне використання MySQL, яка здатна забезпечити надійне зберігання даних та швидкий доступ до них. У сукупності ці компоненти дають змогу створити ефективний, надійний і зручний Web-ресурс, що буде відповідати очікуванням користувачів і вимогам ринку компонентів процесу сонячної генерації.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА WEB-КАТАЛОГУ

2.1 Проєктування інформаційної моделі Web-каталогу

Розробка Web-каталогу сонячних батарей передбачає створення чітко організованої структури зберігання даних. Одним із ключових етапів у цьому процесі є проєктування інформаційної моделі та бази даних, що забезпечить надійну й ефективну роботу системи, підтримку бізнес-логіки та зручну взаємодію з користувачами [37].

Інформаційна модель являє собою логічну структуру, що описує основні сутності предметної області та зв'язки між ними. У контексті Web-каталогу такими сутностями можуть бути: товари, користувачі, замовлення, відгуки тощо. При моделюванні визначається, які дані мають зберігатися, які операції будуть над ними виконуватися і як вони будуть пов'язані між собою. На основі інформаційної моделі за допомогою СКБД створюється структура бази даних. У цій структурі кожна сутність відображається у вигляді таблиці, а зв'язки реалізуються через зовнішні ключі.

На етапі проєктування інформаційної моделі важливу роль відіграє використання мови UML (Unified Modeling Language), яка дає змогу візуалізувати архітектуру системи до початку розробки бази даних. За допомогою діаграм класів UML можна описати основні сутності системи, їхні атрибути та взаємозв'язки, що спрощує розуміння структури даних як для технічних фахівців, так і для замовників. Такі діаграми допомагають виявити потенційні логічні невідповідності на ранніх етапах, сприяють узгодженості між фронтендом і бекендом, а також слугують основою для подальшої розробки реляційної моделі. У проєктах також можуть використовуватися діаграми прецедентів, що демонструють типові сценарії взаємодії користувачів із системою (наприклад, перегляд каталогу, оформлення замовлення, реєстрація). Інтеграція UML-моделювання в процес проєктування підвищує прозорість

архітектури Web-каталогу і сприяє більш системному підходу до розробки [38-40].

На першому етапі розробки інформаційної моделі Web-каталогу створимо діаграму класів UML, яка буде відображати основні виділені сутності та взаємодію між ними. Створена діаграма класів Web-каталогу представлена на рисунку 2.1.

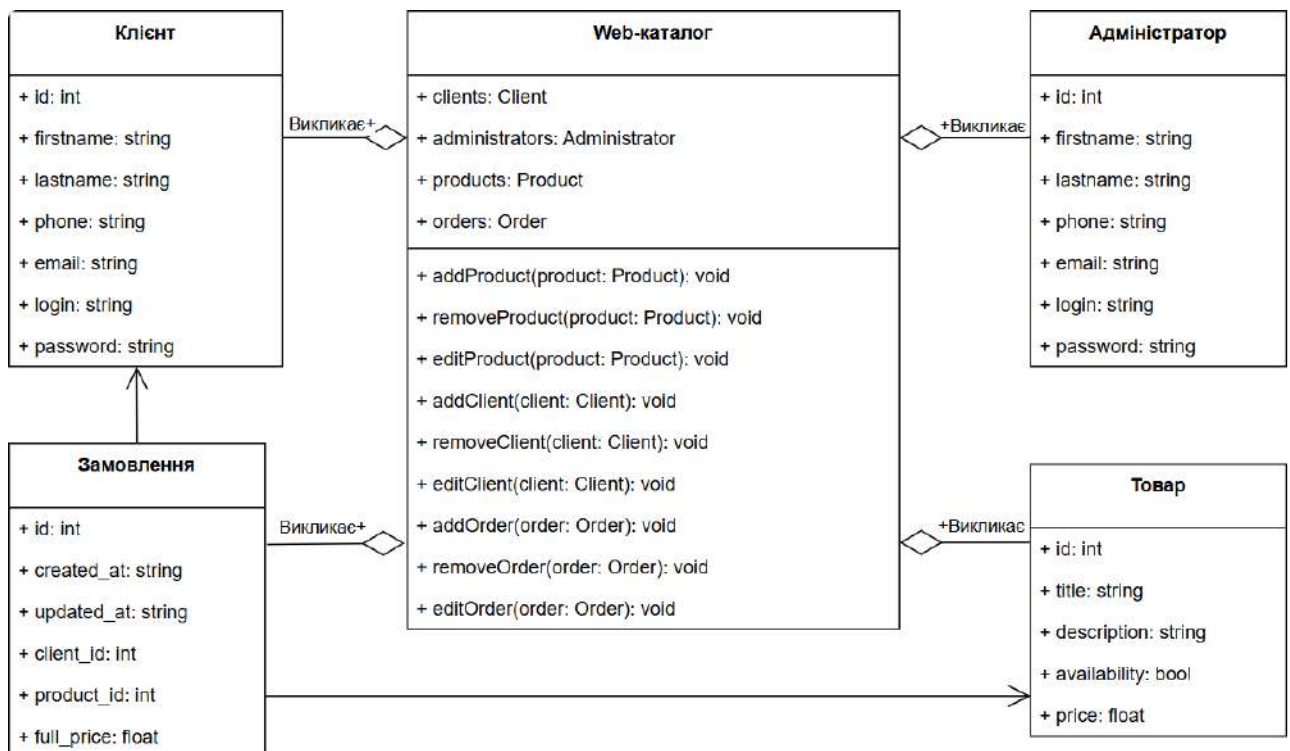


Рис. 2.1. Діаграма класів Web-каталогу сонячних батарей

Представлена діаграма містить п'ять класів, серед яких головний це безпосередньо сам Web-каталог, який містить основну логіку маніпулювання об'єктами інших чотирьох класів: Клієнт, Товар, Замовлення та Адміністратор. Створена діаграма є початковою та може бути в подальшому адаптована до реальних умов розробки, але достатньо ілюструє загальну логіку взаємодії між виділеними сутностями.

Також при створенні інформаційної моделі доцільно побудувати діаграму прецедентів, що відображає основні сценарії взаємодії користувача з системою. Така діаграма дає змогу визначити акторів, які беруть участь у взаємодії, та дії,

які вони можуть виконувати, - авторизацію, перегляд каталогу, додавання та редагування товару, оформлення замовлення тощо. Діаграма прецедентів допомагає не тільки формалізувати вимоги до функціональності Web-каталогу, а й слугує відправною точкою для подальшого проєктування Web-інтерфейсу і реалізації логіки на стороні сервера і клієнта.

Розроблена діаграма прецедентів Web-каталогу сонячних батарей представлена на рисунку 2.2.

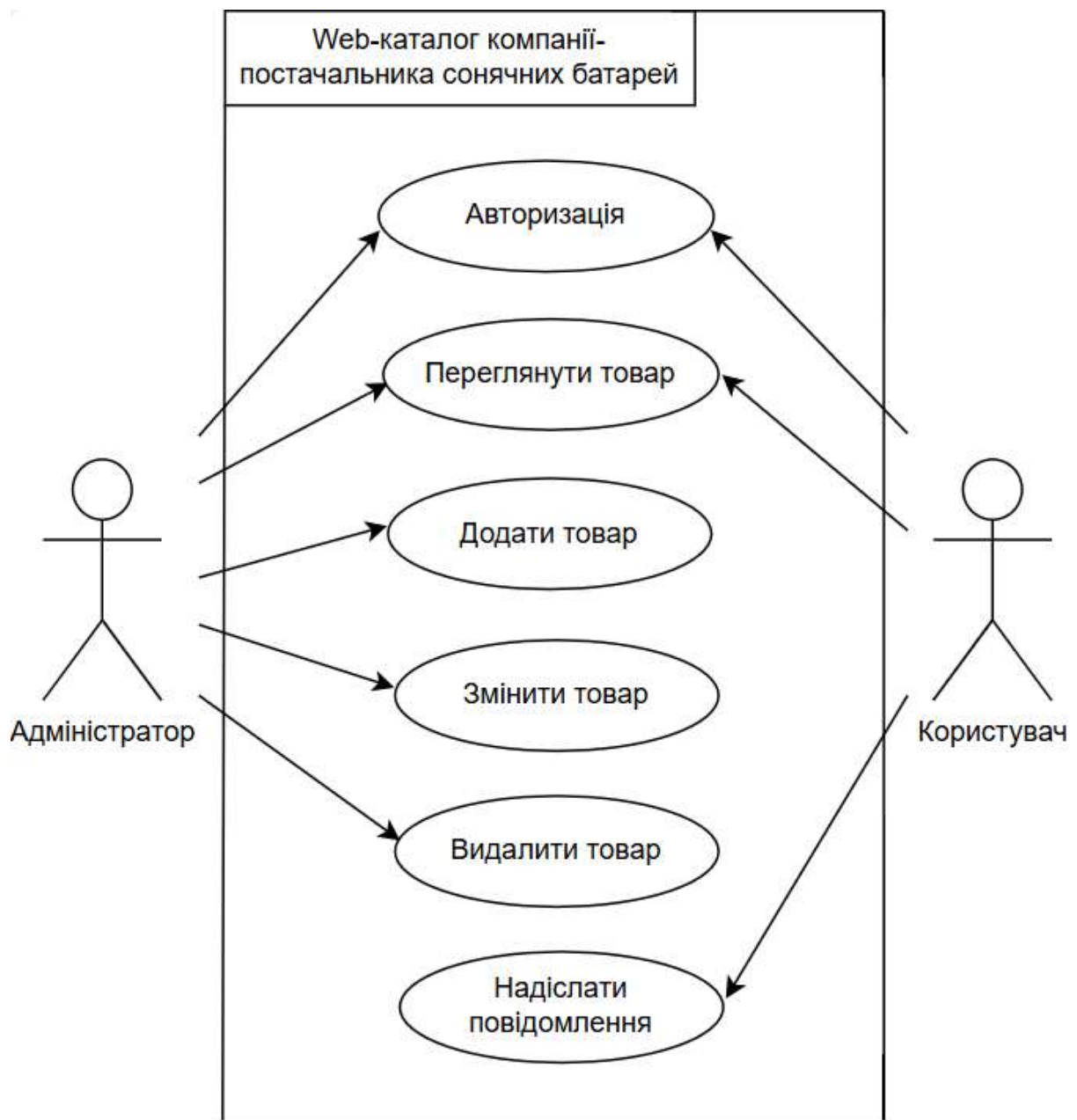


Рис. 2.2. Діаграма прецедентів Web-каталогу сонячних батарей

Представлена діаграма прецедентів передбачає наявність двох ролей: Адміністратор та Користувач. При цьому передбачається, що Користувач має можливість лише переглядати товари та надіслати повідомлення про замовлення до служби підтримки через форму зворотного зв'язку. В той же час Адміністратор після авторизації має можливість виконувати ширший спектр дій серед яких додавання, змінювання та видалення товару.

Наявність такого функціоналу цілком відповідає концепції CRUD (Create, Read, Update, Delete), яка лежить в основі більшості інформаційних систем. Застосування CRUD-підходу сприяє спрощенню проектування та реалізації серверної логіки, а також дає змогу структурувати архітектуру бази даних та інтерфейсів взаємодії з нею. Крім того, подібна логіка легко масштабується: за необхідності можна розширити модель, додавши, наприклад, роль менеджера, відповідального за обробку замовлень. Таким чином, створена діаграма прецедентів не тільки ілюструє поточну функціональність системи, а й закладає основу для її подальшого розвитку.

На наступному етапі розробки інформаційної моделі доцільно розглянути діаграму послідовності прецеденту додавання нового товару, що наочно демонструє покрокову взаємодію між учасниками та об'єктами системи під час виконання цієї операції. Така діаграма дає змогу візуалізувати порядок виклику методів, передачу даних і реакцію компонентів системи на дії користувача. У цьому випадку основними учасниками діаграми є Адміністратор, Web-інтерфейс (фронтенд), серверна логіка (бекенд) і база даних. Розроблена діаграма послідовності прецеденту додавання нового товару представлена на рисунку 2.3.

Процес починається з того, що Адміністратор, після авторизації, через користувацький інтерфейс ініціює додавання товару, заповнюючи відповідну форму з полями, такими як назва, характеристики, ціна і зображення. Після натискання кнопки підтвердження дані відправляються на сервер, де відбувається їхня перевірка. У разі успішної перевірки сервер формує SQL-запит і зберігає дані в базу. Після цього клієнтська частина отримує підтвердження про

успішну операцію і може оновити відображення каталогу та вивести повідомлення про успішне додавання.

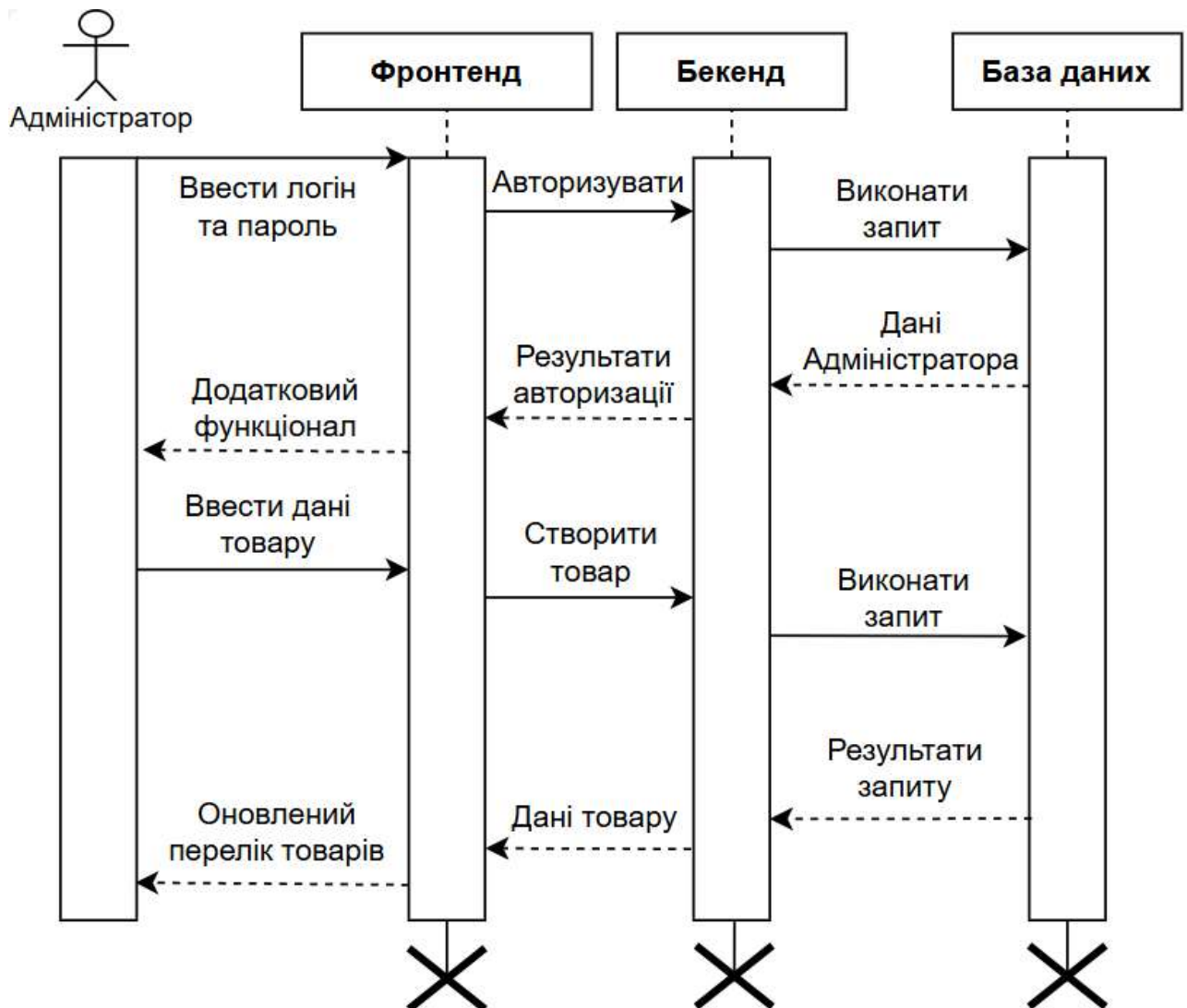


Рис. 2.3. Діаграма послідовності прецеденту додавання нового товару

Рівень деталізації, який надає діаграма послідовності прецеденту особливо важливий для технічної команди, оскільки дає змогу виявити потенційні помилки або слабкі місця Web-застосунку ще до початку написання коду. Діаграма послідовності слугує також чудовим фрагментом документації, що полегшує тестування та подальший супровід проєкту.

Далі доцільно створити концептуальну ER-діаграму, яка слугує основою для подальшого проектування структури бази даних. Така діаграма дає змогу

наочно представити основні сутності системи, їхні атрибути та зв'язки між ними. Вона допомагає визначити, як інформація зберігатиметься в базі даних, встановити типи зв'язків і вибрати правильні ключі та індекси для забезпечення цілісності даних і продуктивності їх використання.

Розробка такої ER-діаграми корисна безпосередньо перед реалізацією структури бази даних у MySQL, оскільки вона дає можливість заздалегідь продумати всі залежності й уникнути дублювання даних або проблем із логікою роботи програми.

У контексті Web-каталогу сонячних батарей, відповідно до діаграми класів (див. рис. 2.1), ключовими сутностями будуть: Товар, Користувач, Замовлення, Адміністратор. Створена концептуальна ER-діаграма в спрощеному вигляді представлена на рисунку 2.4.

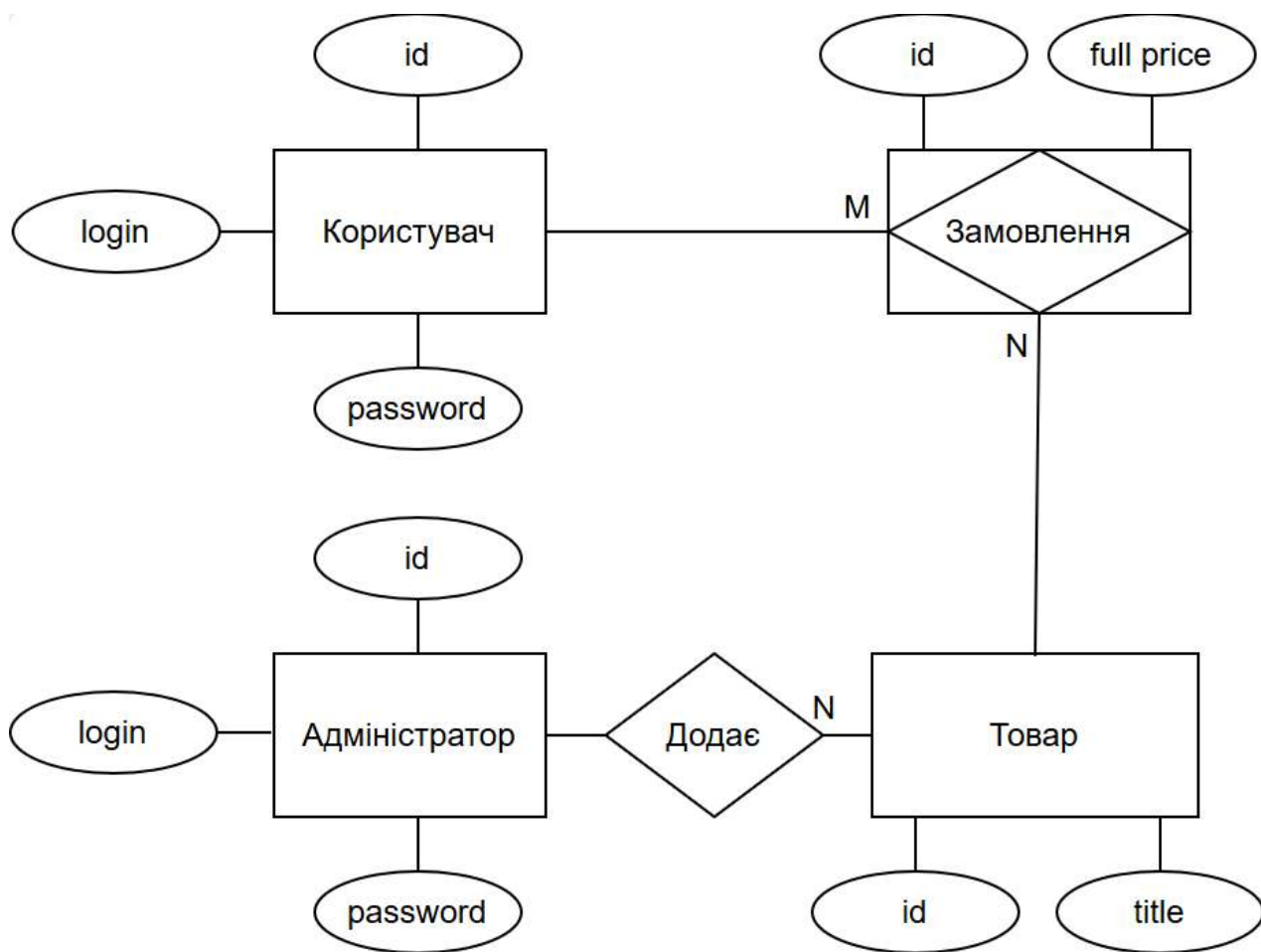


Рис. 2.4. ER-діаграма Web-каталогу сонячних батарей

Таким чином, у результаті розробки UML-діаграм вдалося формалізувати структуру та визначити ключові взаємозв'язки між основними сутностями системи. Отримана інформаційна модель досить повно відображає логіку функціонування Web-каталогу і слугує надійним підґрунтям для проектування бази даних та подальшої реалізації модулів Web-додатку.

2.2 Реалізація архітектури Web-каталогу

Для ефективної реалізації як серверної, так і клієнтської частини Web-каталогу доцільно почати з побудови діаграми компонентів. Така діаграма дає змогу визначити основні логічні модулі системи, їхні взаємозв'язки та межі відповідальності.

Компонентна модель особливо корисна на етапі проектування, оскільки допомагає розділити застосунок на окремі ізольовані частини. Зокрема, на стороні клієнта діаграма компонентів може включати інтерфейсні модулі (каталог товарів, форма авторизації та ін.), що взаємодіють з сервером API. На стороні сервера, можна виокремити компоненти обробки запитів, сервіси роботи з даними, модуль аутентифікації тощо. Крім того використання діаграми компонентів на ранньому етапі допомагає полегшити налагодження та спростити тестування і масштабування проєкту.

На рисунку 2.5 представлена розроблена діаграма компонентів Web-каталогу сонячних батарей. Діаграма демонструє модульну структуру системи та чіткий розподіл функцій між її основними частинами - клієнтською (Frontend) і серверною (Backend). Загалом, ця діаграма відображає архітектуру інформаційної системи та показує основні функціональні блоки, з яких вона складається, а також зв'язки між ними.

Таким чином, архітектура створюваної інформаційної системи передбачає розділення на дві основні частини: Frontend (клієнтська частина) і Backend (серверна частина), кожна з яких реалізує свій набір функцій і взаємодіє з базою даних.

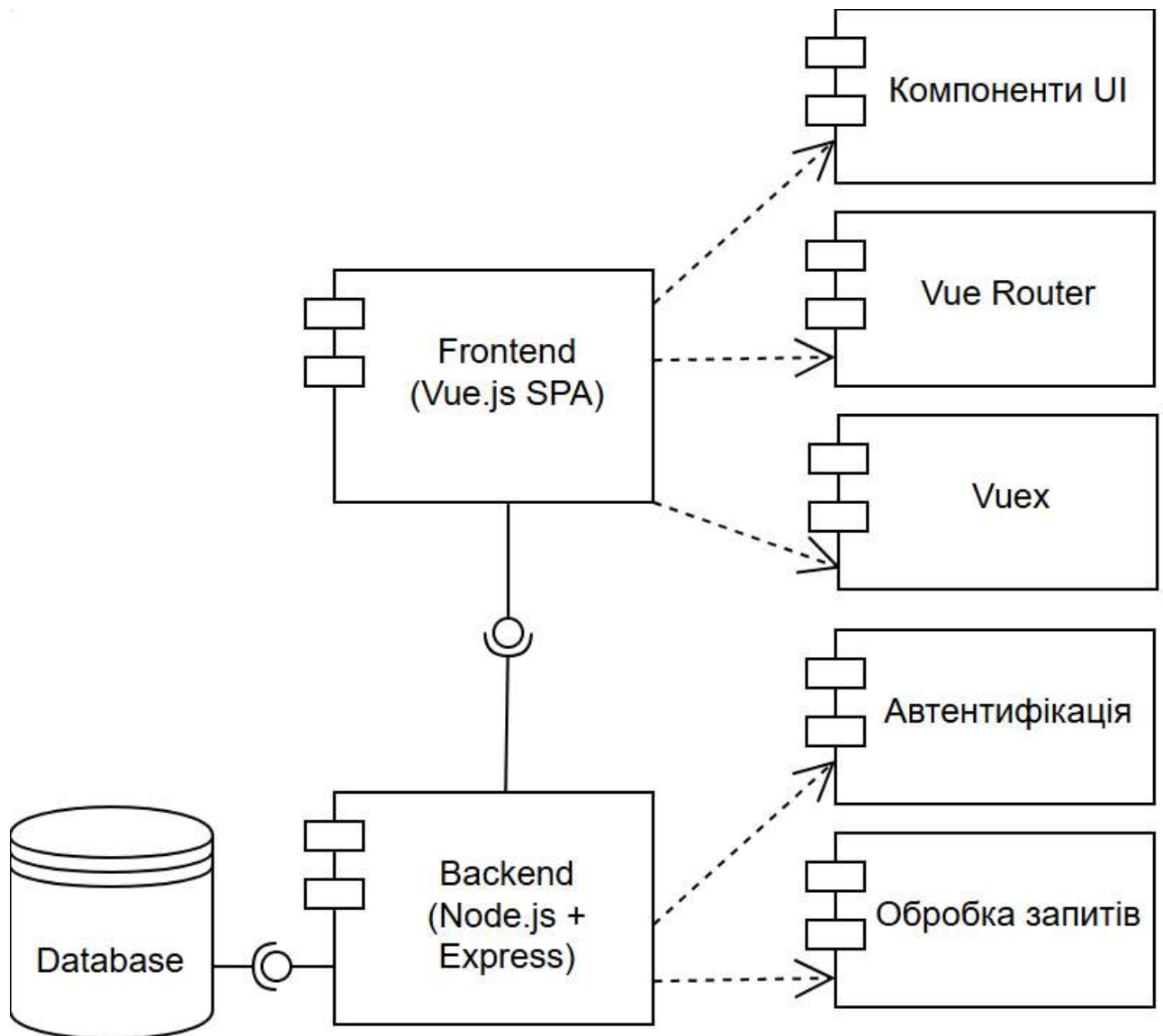


Рис. 2.5. Діаграма компонентів Web-каталогу сонячних батарей

Передбачається, що клієнтська частина реалізована як SPA-додаок з використанням фреймворку Vue.js. До складу цієї частини входять наступні модулі:

UI-компоненти - елементи візуального інтерфейсу, що забезпечують взаємодію з користувачем;

Vue Router - відповідає за маршрутизацію всередині SPA-додатку без перезавантаження сторінки;

Vuex - централізоване сховище даних, що використовується для управління станом SPA-додатку.

Серверна частина розроблена на Node.js з використанням Express, що забезпечує високу продуктивність і гнучкість при використанні REST API. Вона включає наступні два модулі:

Аутентифікація - забезпечує перевірку облікових даних користувачів та розмежування прав доступу;

Обробка запитів - приймає й обробляє HTTP-запити від клієнта, передає їх у відповідні модулі логіки та повертає відповіді.

Додатково система містить окремий компонент Database, який являє собою СУБД MySQL і використовується для зберігання даних необхідних для функціонування Web-каталогу. Серверна частина безпосередньо взаємодіє з базою даних через відповідні запити.

Представлена архітектура компонентів забезпечує чіткий розподіл функцій між клієнтською та серверною частинами, що сприяє спрощенню розробки, масштабованості та подальшому супроводу системи. Використання сучасних фреймворків і підходів, як-от SPA, REST API, дає змогу створити повноцінний і зручний у використанні Web-каталог. При цьому, така компонентна структура слугує надійною основою для подальшої реалізації, тестування та розвитку створюваного програмного забезпечення.

2.3 Розробка окремих модулів Web-каталогу

Розробка Web-каталогу є комплексним завданням, що включає створення як клієнтської, так і серверної частини. Логічним і технічно обґрунтованим кроком є початок розробки з бекенд-частини, оскільки саме вона забезпечує взаємодію з базою даних і реалізацію бізнес-логіки застосунку. Надійно функціонуючий серверний модуль створює основу для подальшої інтеграції користувацького інтерфейсу та забезпечує стабільність роботи всієї системи.

Розглянемо основні частини коду серверної частини почавши з підключення до бази даних. Для взаємодії з базою даних MySQL будемо використовувати бібліотеку mysql2. Це дасть змогу серверу виконувати SQL-

запити до бази даних, такі як додавання, оновлення та видалення даних. Фрагмент коду бекенд-частини Web-каталогу, який відповідає за підключення до бази даних представлений на рисунку 2.6.

```
const db = mysql.createConnection({
  host: 'localhost',
  user: 'user',
  password: 'pass',
  database: 'catalog',
  port: 3306
});

db.connect(err => {
  if (err) {
    console.error('Помилка підключення до MySQL:', err);
  } else {
    console.log('Підключено до MySQL');
  }
});
```

Рис. 2.6. Фрагмент коду бекенд-частини Web-каталогу, який відповідає за підключення до бази даних

Також, за логікою, у створюваному Web-каталозі необхідно реалізувати аутентифікацію користувачів, оскільки, виходячи зі створеної інформаційної моделі, передбачається мінімум дві ролі: Адміністратор та Користувач. Фрагмент коду серверної частини Web-каталогу, який відповідає за аутентифікацію користувачів представлений на рисунку 2.7.

```
app.post('/login', (req, res) => {
  const { username, password } = req.body;

  db.query('SELECT * FROM users WHERE username = ?', [username], (err, results) => {
    if (err) return res.status(500).send(err);
    const user = results[0];
    if (!user) return res.status(400).json({ message: 'Користувач не знайдений' });
    if (password !== user.password) {
      return res.status(400).json({ message: 'Невірний пароль' });
    }
    const token = jwt.sign({ userId: user.id, role: user.role }, SECRET_KEY, { expiresIn: '1h' });
    res.json({ token });
  });
});
```

Рис. 2.7. Фрагмент коду, який відповідає за аутентифікацію користувачів

Під час аутентифікації використовується HTTP-запит за маршрутом /login, де сервер приймає ім'я користувача та пароль, перевіряє їх у базі даних і, в разі успіху, повертає клієнту токен доступу. Цей токен створюється за допомогою бібліотеки jsonwebtoken і містить інформацію про користувача та його роль.

Далі розглянемо реалізацію основних дій з товарами; додавання, редагування та видалення. За реалізацію цих дій відповідає фрагмент коду представлений на рисунку 2.8.

```
app.post('/products', authenticateToken, isAdmin, (req, res) => {
  const { name, price, description, image_url } = req.body;
  db.query(
    'INSERT INTO products (name, price, description, image_url) VALUES (?, ?, ?, ?)',
    [name, price, description, image_url],
    err => {
      if (err) return res.status(500).send(err);
      res.json({ message: 'Товар додано' });
    }
  );
});

app.put('/products/:id', authenticateToken, isAdmin, (req, res) => {
  const { name, price, description, image_url } = req.body;
  const { id } = req.params;
  db.query(
    'UPDATE products SET name = ?, price = ?, description = ?, image_url = ? WHERE id = ?',
    [name, price, description, image_url, id],
    err => {
      if (err) return res.status(500).send(err);
      res.json({ message: 'Товар оновлено' });
    }
  );
});

app.delete('/products/:id', authenticateToken, isAdmin, (req, res) => {
  const { id } = req.params;
  db.query('DELETE FROM products WHERE id = ?', [id], err => {
    if (err) return res.status(500).send(err);
    res.json({ message: 'Товар видалено' });
  });
});
```

Рис. 2.8. Фрагмент коду бекенд-частини Web-каталогу, який відповідає за додавання, редагування та видалення товару

Основні операції з товарами виконуються через HTTP-запити до відповідних маршрутів серверної частини і вимагають авторизації з правами Адміністратора. Зокрема, додавання товару здійснюється за допомогою методу POST за маршрутом /products. (сервер отримує дані нового товару з тіла запиту, формує SQL-запит і зберігає запис у таблиці products), редагування товару реалізовано через метод PUT з передачею ідентифікатора товару в параметрах маршруту /products/:id (сервер оновлює відповідний запис у базі даних на основі нових даних, переданих у тілі запиту), а видалення товару відбувається через метод DELETE за маршрутом /products/:id (сервер виконує SQL-запит на видалення запису з зазначеним ідентифікатором із таблиці products).

Таким чином, цей фрагмент коду реалізує повний набір CRUD-операцій над товарами (за винятком отримання списку товарів, реалізацію якого розглянемо окремо далі).

Також, слід зазначити, що типовий Web-каталог, як правило передбачає представлення товарів у вигляді карток з зображеннями, тому необхідно реалізувати можливість додавання зображення при додаванні товару. Для цього в серверній частині використовується бібліотека multer, яка обробляє завантаження файлів через HTTP-запити. Файли зображень зберігаються в локальну директорію uploads, а сервер автоматично створює цю папку під час першого запуску, якщо вона відсутня. Маршрут POST /upload приймає зображення від авторизованого користувача. Після успішного завантаження сервер повертає клієнту URL, за яким завантажене зображення буде доступне. Цей URL надалі передається разом з рештою даних товару в запиті на додавання або редагування товару. Загальний код, який реалізує додавання зображення представлений на рисунку 2.9.

Таким чином, реалізована можливість завантаження зображень забезпечує візуальний супровід карток товарів і робить каталог більш зручним та інформативним для користувачів. Після реалізації механізму завантаження зображень логічним продовженням є організація виведення списку товарів, зокрема з відображенням відповідних зображень.

```

const uploadsDir = path.join(__dirname, 'uploads');
if (!fs.existsSync(uploadsDir)) {
  fs.mkdirSync(uploadsDir);
}
const storage = multer.diskStorage({
  destination: uploadsDir,
  filename: (req, file, cb) => {
    const ext = path.extname(file.originalname);
    const uniqueName = Date.now() + ext;
    cb(null, uniqueName);
  }
});
const upload = multer({ storage });

app.post('/upload', authenticateToken, upload.single('image'), (req, res) => {
  const imageUrl = `http://localhost:${PORT}/uploads/${req.file.filename}`;
  res.json({ imageUrl });
});

```

Рис. 2.9. Фрагмент коду бекенд-частини Web-каталогу, який відповідає за завантаження зображення товару

Розглянемо реалізацію отримання списку товарів. Для цього в серверній частині Web-каталогу реалізовано маршрут GET /products, який доступний усім користувачам, включно з неавторизованими. Це дає змогу будь-якому відвідувачу сайту переглядати товари, представлені в каталозі. На рисунку 2.10 представлений фрагмент коду, який відповідає за завантаження списку товарів.

```

app.get('/products', (req, res) => {
  db.query('SELECT * FROM products', (err, results) => {
    if (err) return res.status(500).send(err);
    res.json(results);
  });
});

```

Рис. 2.10. Фрагмент коду бекенд-частини Web-каталогу, який відповідає за завантаження списку товарів

При отриманні запиту на вказаний маршрут, сервер звертається до бази даних і виконує SQL-запит на вибірку всіх записів з таблиці products. У разі успішного виконання запиту, сервер повертає клієнту масив об'єктів з

інформацією про кожен товар - назвою, ціною, описом і посиланням на зображення. Така реалізація забезпечує швидкий і простий доступ до даних каталогу, а також створює основу для відображення товарів на клієнтській стороні, наприклад, у вигляді карток.

Також, у відповідності з інформаційною моделлю Web-додатку, для зареєстрованих користувачів необхідно реалізувати надсилання форми зворотного зв'язку (повідомлення). На стороні сервера ця функціональність реалізується за допомогою маршруту POST /feedback, який приймає дані форми: ім'я користувача, адресу електронної пошти, номер телефону і текст повідомлення.

Після отримання даних сервер виконує SQL-запит на додавання нового запису в таблицю feedback, де додатково зберігається ідентифікатор користувача, який надіслав повідомлення. Це дає змогу надалі пов'язувати кожне звернення з конкретним користувачем і, за потреби, реалізувати перегляд історії звернень адміністраторами. Фрагмент коду, який забезпечує надсилання форми зворотного зв'язку представлений на рисунку 2.11.

```
app.post('/feedback', authenticateToken, (req, res) => {
  const { name, email, phone, message } = req.body

  db.query(
    'INSERT INTO feedback (name, email, phone, message, user_id) VALUES (?, ?, ?, ?, ?)',
    [name, email, phone, message, req.user.userId],
    (err) => {
      if (err) return res.status(500).send(err)
      res.json({ message: 'Форма зворотного зв'язку збережена' })
    }
  )
})
```

Рис. 2.11. Фрагмент коду бекенд-частини Web-каталогу, який забезпечує надсилання форми зворотного зв'язку

Таким чином, на поточному етапі розробки реалізовано основну функціональність серверної частини Web-каталогу. Отриманий функціонал є основою для подальшої інтеграції з клієнтською частиною та розширення

можливостей Web-додатку. Повний код серверної (backend) частини Web-каталогу наведений в додатку А.

Далі доцільно розглянути реалізацію деяких основних модулів клієнтської (frontend) частини створюваного Web-каталогу. При цьому доцільно розпочати з форми авторизації користувача.

Компонент форми авторизації забезпечує введення логіна і пароля, надсилання даних на сервер і, в разі успішної авторизації, збереження JWT-токена в локальному сховищі. Після цього здійснюється оновлення стану аутентифікації в основному компоненті застосунку. Це дає змогу гнучко керувати доступом до різних розділів каталогу, зокрема до адміністративної частини. Фрагмент коду з розміткою форми авторизації представлений на рисунку 2.12.

```
<template>
  <form @submit.prevent="login" class="login-form">
    <input v-model="loginForm.username" placeholder="Логін" required />
    <input v-model="loginForm.password" type="password" placeholder="Пароль" required />
    <button type="submit">Увійти</button>
  </form>
</template>
```

Рис. 2.12. Фрагмент коду з розміткою форми авторизації

Також, необхідно реалізувати компонент, який буде виконувати відображення товарів. Цей компонент отримує масив об'єктів і візуалізує кожен з них у вигляді окремої картки, що містить найменування, зображення, опис і ціну товару. Фрагмент коду з розміткою відображення карток товарів представлений на рисунку 2.13.

В наведеному компоненті реалізована логіка відображення для Адміністратора додаткових кнопок: «Редагувати» і «Видалити» на кожній картці товару. Ці елементи надають можливість в інтерактивному режимі виконувати відповідні дії з товаром - ініціювати його редагування або видалити зі списку. Таким чином, адміністратор може оперативно керувати вмістом каталогу прямо

з інтерфейсу сайту без необхідності переходу на окремі сторінки або виконання додаткових запитів вручну. Це підвищує зручність і ефективність адміністрування Web-ресурсу.

```

<template>
  <div class="product-container">
    <div v-for="product in products" :key="product.id" class="product-card">
      <h3 class="product-name">{{ product.name }}</h3>
      
      <p class="product-description">{{ product.description }}</p>
      <h3 class="price">Ціна {{ product.price }} грн</h3>

      <div v-if="isAdmin" class="admin-buttons">
        <button class="admin-btn edit-btn" @click="$emit('edit', product)">
          ✎ Редагувати</button>
        <button class="admin-btn delete-btn" @click="$emit('delete', product.id)">
          🗑️ Видалити</button>
      </div>
    </div>
  </div>
</template>

```

Рис. 2.13. Фрагмент коду з розміткою відображення карток товарів

Для того, щоб Адміністратор міг оперативно керувати вмістом каталогу прямо з інтерфейсу сайту, необхідно створити відповідну форму, яка буде доступна тільки йому. Ця форма повинна містити набір полів, необхідних для введення інформації про товар: назва, опис та ціна. Для завантаження зображення доцільно використати елемент керування типу «file», який дає змогу прикріпити файл з локального пристрою. При цьому має бути реалізовано відправлення зображення на сервер із подальшим отриманням URL зображення, який зберігається в базі даних.

Якщо адміністратору необхідно відредагувати наявний товар, поля форми мають автоматично заповнюватися поточними даними цього товару, що дає

змогу швидко внести зміни. Після завершення редагування або додавання товару дані відправляються на сервер через HTTP-запит (POST або PUT залежно від дії). Також, необхідно передбачити можливість скасування редагування за допомогою кнопки «Скасувати», що дає змогу повернутися до початкового стану форми. В рамках цієї роботи було реалізовано компонент форми додавання та редагування товару відповідно до наведеного опису, Код цієї форми є частиною основного Vue-компоненту.

На завершальному етапі розробки клієнтської (frontend) частини Web-каталогу треба створити компонент форми зворотного зв'язку для зареєстрованих користувачів. Ця форма дозволяє залишити повідомлення, вказавши ім'я, email, номер телефону та текст звернення. Усі поля є обов'язковими для заповнення, що забезпечує отримання повної контактної інформації від користувача. Фрагмент коду з розміткою форми зворотного зв'язку для зареєстрованих користувачів представлений на рисунку 2.14.

```
<template>
  <form @submit.prevent="submitFeedback" class="form_feedback">
    <div class="textarea-block">
      <textarea v-model="feedback.message" placeholder="Повідомлення" required>
      </textarea>
    </div>
    <div class="inputs-block">
      <input v-model="feedback.name" placeholder="Ваше ім'я" required />
      <input v-model="feedback.email" type="email" placeholder="Email" required />
      <input v-model="feedback.phone" type="tel" placeholder="Телефон" required />
      <button type="submit">Надіслати</button>
    </div>
  </form>
</template>
```

Рис. 2.14. Фрагмент коду з розміткою форми зворотного зв'язку для зареєстрованих користувачів

Форма використовує токен авторизації, що зберігається в localStorage, для перевірки прав доступу та надсилання даних на сервер. Після успішної відправки користувачеві відображається повідомлення, а поля форми автоматично

очищаються. Таким чином забезпечується зручний механізм зв'язку користувачів з адміністрацією сайту.

Використання фреймворку Vue.js передбачає, що усі розроблені компоненти будуть використовуватися в основному компоненті Web-додатку, що забезпечує централізоване управління станом, маршрутизацією та відображенням вмісту залежно від ролей користувачів. Така структура полегшує масштабування проєкту, підвищує читабельність коду і сприяє модульній організації інтерфейсу, де кожен компонент відповідає за певну частину функціональності Web-каталогу. Повний код основного Vue-компоненту створеного Web-інтерфейсу наведений у додатку Б.

Висновки до розділу 2

Отримана інформаційна модель достатньо повно відображає структуру і логіку функціонування Web-каталогу та стала міцним підґрунтям для реалізації ключових модулів клієнтської та серверної частин Web-додатку. На основі цієї моделі вдалося вибудувати цілісну архітектуру системи, яка передбачає поділ на серверну (backend) та клієнтську (frontend) частини. Такий підхід сприяє зручності розробки, супроводу та масштабування проєкту.

Використання сучасних підходів, таких як SPA та REST API, дає змогу реалізувати якісний Web-застосунок, у якому взаємодія між компонентами здійснюється через зрозумілі та стандартизовані інтерфейси. При цьому застосування фреймворку Vue.js дало змогу реалізувати Web-інтерфейс у вигляді сукупності зв'язаних компонентів, кожен із яких реалізує відокремлений чітко визначений функціонал.

Таким чином, обрана архітектура та реалізовані на її основі програмні компоненти, формують сучасний якісний Web-ресурс, який здатен ефективно виконувати функції електронного каталогу та в подальшому може бути адаптований під необхідні умови.

РОЗДІЛ 3

ФУНКЦІОНУВАННЯ ТА ЕЛЕМЕНТИ ІНТЕРФЕЙСУ WEB-КАТАЛОГУ

3.1 Огляд функціоналу створеного Web-каталогу

Створений Web-каталог являє собою сучасний SPA, в якому реалізовано функції перегляду, додавання, редагування та видалення товарів. При цьому Web-інтерфейс користувача реалізовано з урахуванням двох можливих ролей: Адміністратор і Користувач. Залежно від ролі інтерфейс динамічно змінюється, при цьому, до авторизації всі відвідувачі бачать тільки публічну частину каталогу - список товарів з їхніми зображеннями, назвами, цінами і описами (але без можливості управління контентом або надсилання повідомлень) та форму авторизації. На рисунку 3.1 показаний загальний вигляд Web-каталогу до виконання авторизації.



Рис. 3.1. Загальний вигляд Web-каталогу до виконання авторизації

Після входу в систему за допомогою токена визначається, яку роль виконує користувач, і на основі цього відкриваються відповідні можливості. Зокрема, якщо користувач - Адміністратор, то він отримує доступ до форми, за допомогою якої можна додавати нові товари в каталог та редагувати наявні. У цій формі передбачені поля для назви, ціни та опису товару, а також можливість завантаження зображення товару. Адміністратор також може редагувати наявні товари або видаляти їх для цього у картках товарів передбачені відповідні інтерактивні елементи, які доступні тільки Адміністратору. Усі дії з управління товарами супроводжуються оновленням списку без необхідності перезавантаження сторінки. На рисунку 3.2 показаний загальний вигляд Web-каталогу після входу Адміністратора.

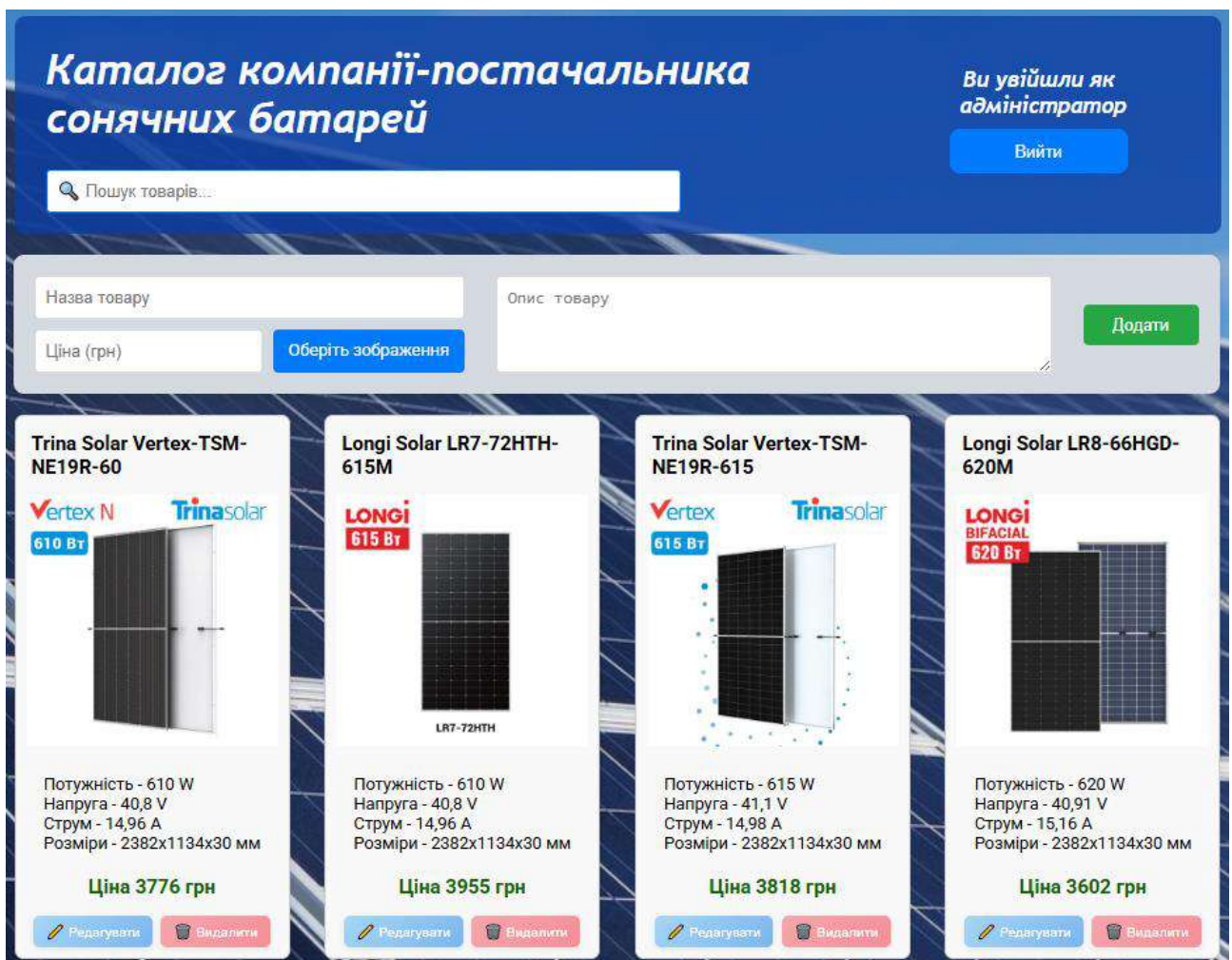


Рис. 3.2. Загальний вигляд Web-каталогу після входу Адміністратора

Зареєстровані користувачі, які пройшли авторизацію, але не мають прав Адміністратора, можуть переглядати всі товари, розміщені на сайті без можливості додавання або редагування, а також залишати повідомлення через форму зворотного зв'язку, що стає доступною тільки після входу в систему. Це дає змогу надати можливість надсилання повідомлень тільки реальним, перевіреним користувачам, виключивши анонімні звернення.

Форма зворотного зв'язку містить обов'язкові поля для введення імені користувача, його електронної пошти та номера телефону, а також текстового повідомлення, в якому користувач може вказати товари, що його цікавлять, уточнити характеристики або зробити запит на оформлення замовлення. Після відправлення дані форми автоматично обробляються і зберігаються у відповідній таблиці бази даних. На рисунку 3.3 показана форма зворотного зв'язку для зареєстрованих користувачів, а також нижня частина сайту (footer), яка містить контактну інформацію компанії-постачальника сонячних батарей.

The image displays a user interface for a solar battery supplier. At the top, there are four product cards, each featuring a solar panel image and technical specifications:

- Card 1:** Потужність - 430 W, Напруга - 33,26 V, Струм - 12,93 A, Розміри - 1722x1134x30 мм. Ціна 2739 грн.
- Card 2:** Потужність - 435 W, Напруга - 33,04 V, Струм - 13,17 A, Розміри - 1722x1134x30 мм. Ціна 2739 грн.
- Card 3:** Потужність - 440 W, Напруга - 44 V, Струм - 10,01 A, Розміри - 1762x1134x30 мм. Ціна 2739 грн.
- Card 4:** Потужність - 580 W, Напруга - 44,06 V, Струм - 13,17 A, Розміри - 2278x1134x30 мм. Ціна 3361 грн.

Below the product cards is a feedback form with the following fields:

- Повідомлення (Text area)
- Ваше ім'я (Text input)
- Email (Text input)
- Телефон (Text input)
- Надіслати (Green button)

The footer section contains the following information:

- ТОВ "Постачальник сонячних батарей"
- Телефон: +38 (095) 123-45-67
- Email: info@solarpower.ua
- © 2025 Постачальник сонячних батарей. Усі права захищено.

Рис. 3.3. Форма зворотного зв'язку для зареєстрованих користувачів та нижня частина сайту (footer)

Адаптивність інтерфейсу створеного Web-каталогу забезпечена завдяки використанню технології Flexbox, яка дає змогу гнучко керувати розташуванням і розмірами елементів на сторінці залежно від розмірів екрана пристрою. Завдяки цьому інтерфейс коректно відображається як на великих екранах (наприклад, на моніторах настільних комп'ютерів), так і на пристроях з меншою роздільною здатністю. Картки товарів та інші компоненти інтерфейсу автоматично перебудовуються і масштабуються, забезпечуючи зручність використання створеного Web-інтерфейсу в різних умовах. На рисунку 3.4 представлений фрагмент Web-інтерфейсу каталогу, відображений на планшеті з умовною роздільною здатністю екрану 860x950 пікселів.

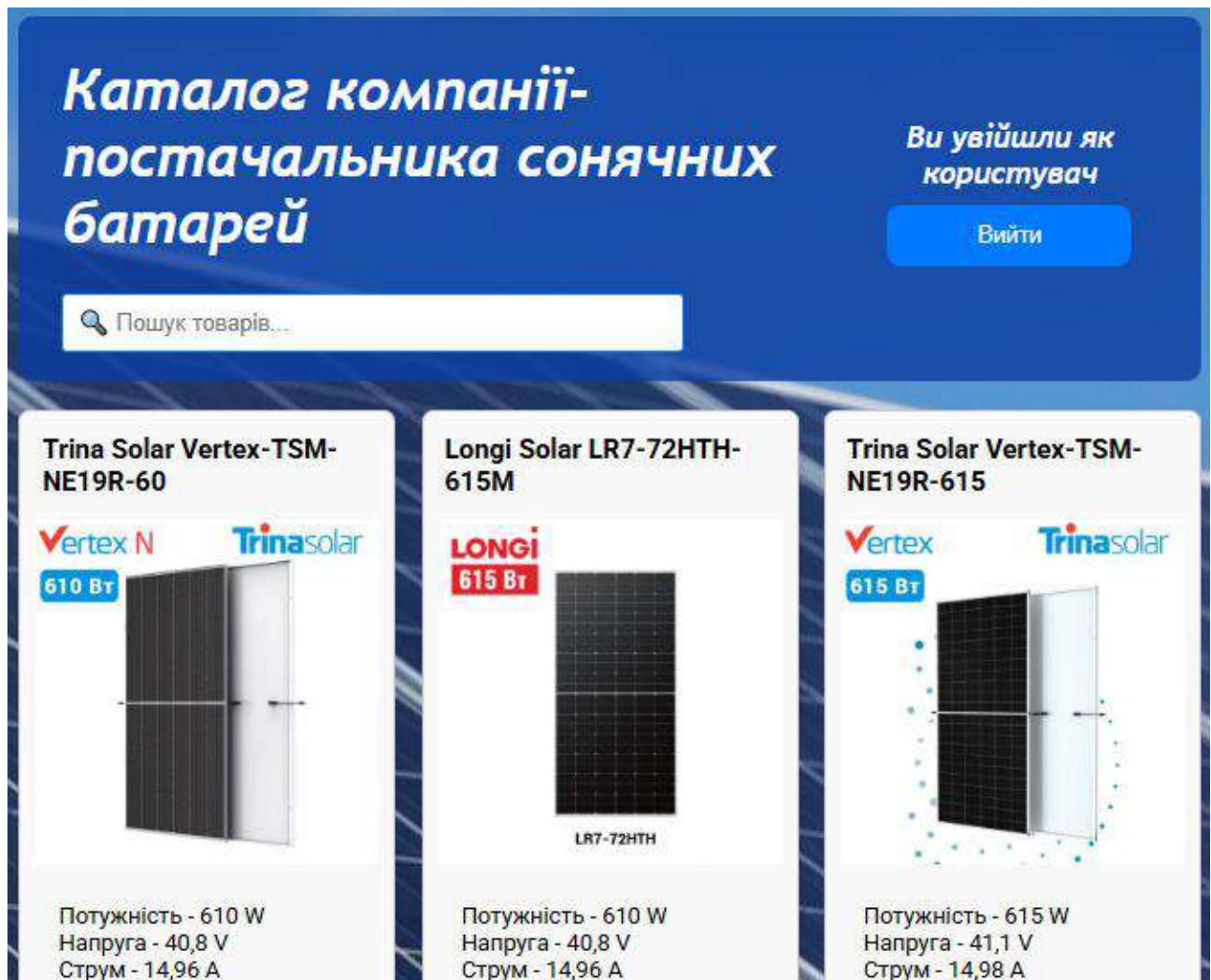


Рис. 3.4. Фрагмент Web-інтерфейсу каталогу, відображений на планшеті з умовною роздільною здатністю екрану 860x950 пікселів

На представленому фрагменті видно, що у порівнянні з початковою версією (див. рис. 3.1) автоматично зменшилась кількість товарів у кожному рядку (до трьох), а також автоматично зменшилася відстань між елементами верхньої частини (header) сторінки.

Таким чином створений Web-каталог являє собою повнофункціональний SPA-додаток, та (залежно від ролі користувача) забезпечує відкритий перегляд, додавання/редагування товарів та доступ до форми зворотного зв'язку, що робить його зручним інструментом як для клієнтів, так і для Адміністратора.

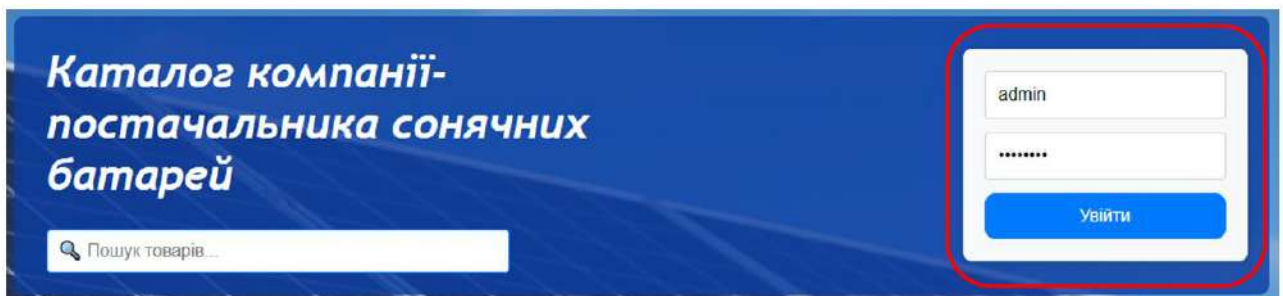
3.2 Реалізація можливих сценаріїв взаємодії з Web-каталогом

Взаємодія з Web-каталогом охоплює широкий спектр дій, спрямованих як на перегляд переліку товарів, так і на редагування та розширення його вмісту. При цьому взаємодія з Web-каталогом може передбачати різні сценарії, які можуть починатися з простого ознайомлення з асортиментом (з використанням системи пошуку щоб швидше знайти потрібну інформацію) та включати в себе взаємодію з іншими інтерактивними елементами Web-інтерфейсу, залежно від необхідного кінцевого результату. Наприклад, Користувач, після перегляду позицій Web-каталогу, може заповнити форму зворотного зв'язку, якщо в нього виникли запитання, пропозиції або потреба в уточненні деталей.

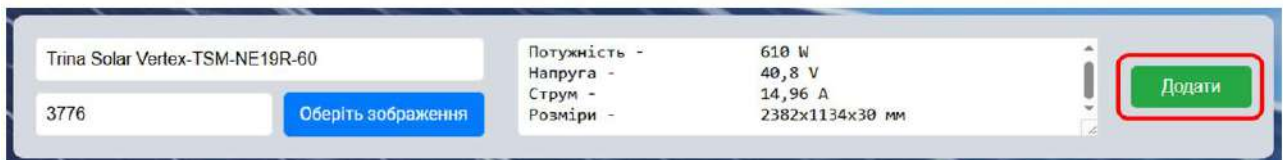
Деякі сценарії передбачають глибшу роботу зі вмістом каталогу - додавання нових товарів, редагування наявних даних і управління структурою. Виконання цих сценаріїв, безпосередньо впливає на вміст Web-каталогу, тому їх реалізація має бути надійною та інтуїтивно зрозумілою.

При цьому, слід зазначити, що такі дії виконуються в інтерфейсі, що забезпечує доступ до розширеного функціоналу, який повинен бути активним тільки після проходження процедури авторизації Адміністратором. Такий підхід дає змогу розмежувати рівні доступу між ролями та запобігти несанкціонованим змінам даних, що особливо важливо під час роботи з комерційним контентом.

Спочатку розглянемо послідовність дій, які необхідно виконати для додавання нового товару. Розроблений Web-інтерфейс передбачає, що для додавання нового товару, Адміністратор Web-ресурсу спочатку повинен виконати авторизацію шляхом заповнення відповідної форми (рис. 3.5, а). Після цього для Адміністратора стане доступною форма додавання нового товару (рис. 3.5, б), в яку необхідно внести назву, ціну, опис товару, а також обрати зображення товару (натиснувши відповідну кнопку). Після внесення необхідних даних, Адміністратор повинен натиснути кнопку додати, після чого новий товар (сонячна батарея) автоматично з'явиться в переліку товарів без необхідності перезавантаження Web-сторінки.



а)



б)

Рис. 3.5. Послідовність процедури додавання нового товару у Web-каталог

Також створений Web-каталог надає Адміністратору можливість редагування товарів у каталозі, включно зі зміною назви, опису, ціни, та зображення. Редагування здійснюється через доступний та інтуїтивно зрозумілий інтерфейс, який стає доступним тільки після авторизації. Це дає змогу оперативно оновлювати інформацію про продукцію, підтримувати її актуальність і вносити коригування в разі зміни асортименту.

Для виконання редагування товару, Адміністратор (після авторизації) може спершу скористуватися функцією пошуку товарів (рис. 3.6), щоб швидше знайти позицію, яка потребує коригування.

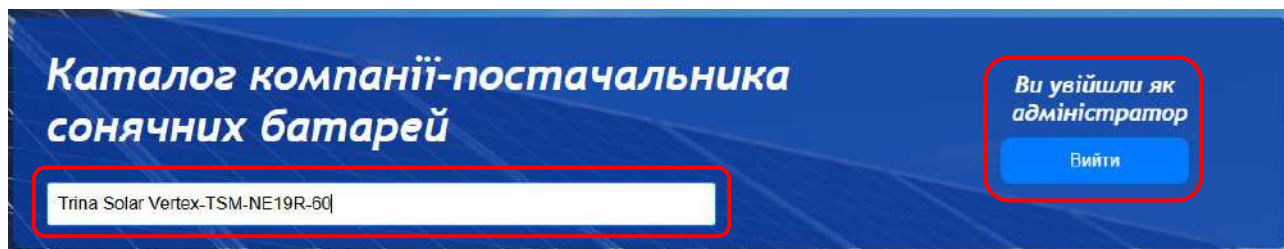


Рис. 3.6. Пошук товару Адміністратором за назвою

Далі, після знаходження потрібної позиції у каталозі Адміністратору необхідно натиснути кнопку «Редагувати», яка знаходиться у нижній частині кожної картки товару (рис. 3.7). При цьому слід відзначити, що цей інтерактивний елемент доступний тільки адміністратору.

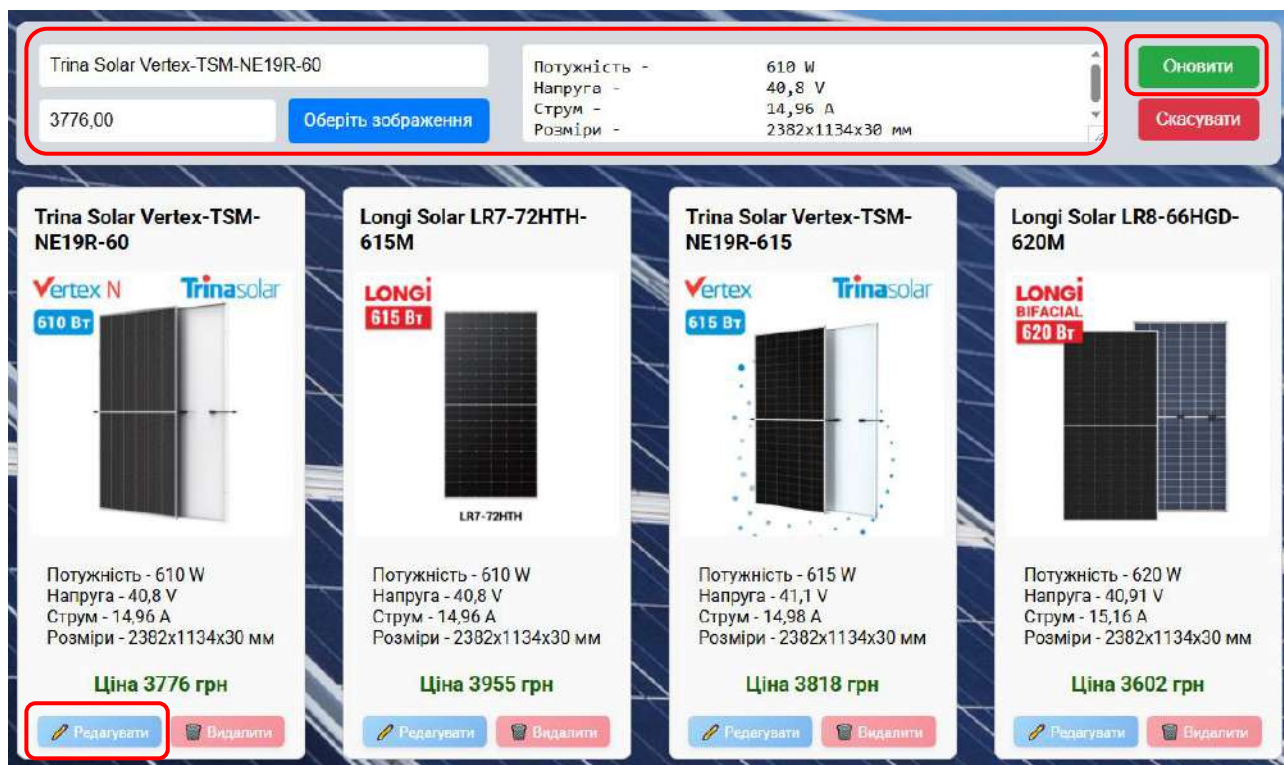


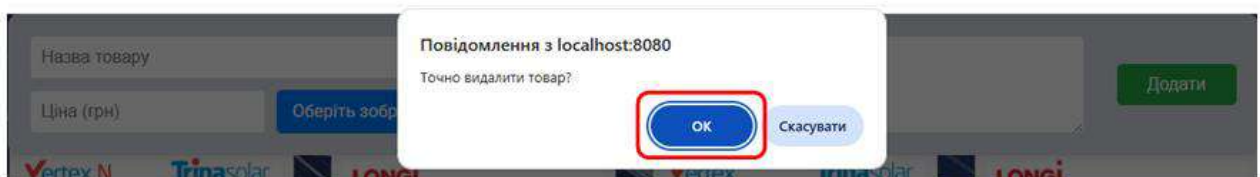
Рис. 3.7. Послідовність процедури редагування товару у Web-каталозі Адміністратором

Після натискання кнопки «Редагувати», дані обраного товару автоматично заповняють відповідні поля у формі, яка використовується для редагування та додавання товарів. Після цього Адміністратор може внести необхідні корективи та натиснути кнопку «Оновити», щоб зберегти зміни. При цьому, є також можливість скасувати зміни (наприклад, якщо позиція каталогу обрана помилково), для цього Адміністратор може натиснути кнопку «Скасувати», яка також знаходиться у формі редагування товару.

Також, у Адміністратора є можливість видалення будь-якого товару, у разі виникнення такої необхідності. Для цього, після авторизації та пошуку необхідної позиції Адміністратор повинен натиснути кнопку «Видалити», яка також знаходиться у нижній частині картки товару поряд з кнопкою «Редагувати» (рис. 3.8, а). Після цього треба підтвердити дію у спливаючому вікні (рис. 3.8, б), це необхідно для запобігання випадковому видаленню товару.



а)



б)

Рис. 3.8. Послідовність процедури видалення товару Адміністратором

Таким чином, створений Web-інтерфейс дійсно надає Адміністратору весь основний функціонал для ефективного управління вмістом каталогу. Адміністратор може в зручній формі додавати нові товари, вказувати їхні характеристики, завантажувати зображення, а за необхідності - редагувати або видаляти наявні позиції. При цьому усі дії виконуються без перезавантаження сторінки, що прискорює процес адміністрування і робить його більш інтуїтивно зрозумілим. Такий підхід забезпечує повнофункціональне управління даними в режимі реального часу і сприяє оперативному оновленню інформації на сайті.

В той же час, для зареєстрованих користувачів, створений Web-інтерфейс надає можливість відправлення повідомлення до адміністрації Web-каталогу шляхом заповнення відповідної форми зворотного зв'язку. Для цього зареєстрований користувач повинен заповнити наведені поля форми зворотного зв'язку: ім'я, email, телефон та текст повідомлення (рис. 3.9).

Добрий день хочу замовити наступні позиції:

1. Tongwei-TWMND-54HE430W	5 шт.
2. Longi Solar LR5-54HTH 435M	7 шт.
3. Trina Solar Vertex-TSM-NE9R.28	3 шт.
4. Longi Solar LR5-72HTH-580M	5 шт.
5. Tongwei-TWMND-78HD625W	3 шт.

Прошу мені перетелефонувати для обговорення деталей.

Сергій

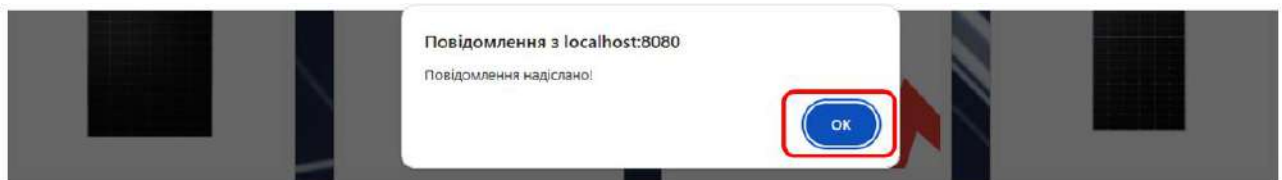
sergij@post.com.ua

(095)345-12-13

Надіслати

Рис. 3.9. Послідовність процедури надсилання повідомлення зареєстрованим користувачем до адміністрації Web-каталогу

Після заповнення всіх полів форми зворотного зв'язку користувач має натиснути кнопку «Відправити» (що розміщена в правому нижньому кутку цієї форми) і підтвердити дію, натиснувши «ОК» у спливаючому вікні (рис. 3.10, а). Після цього дані буде надіслано на сервер, де відбувається їх опрацювання та збереження у відповідній таблиці (feedback) бази даних (рис. 3.10, б). Користувач не бачить технічних деталей процесу, але в разі успішного відправлення отримує візуальне підтвердження (очищення форми), що забезпечує простий і зрозумілий спосіб зв'язку з адміністрацією сайту.



а)

id	name	email	phone	message	user_id	created_at
2	Сергій	sergij@post.com.ua	(095)345-12-13	Добрий день хочу замовити наступні позиції: 1. Топ...	2	2025-05-18

б)

Рис. 3.10. Спливаюче вікно підтвердження відправки заповненої форми зворотного зв'язку (а) та відповідний запис у таблиці бази даних (б) після підтвердження відправки

Таким чином, Web-каталог забезпечує зручний і зрозумілий Web-інтерфейс для користувачів з різними ролями і, як наслідок, рівнями доступу. Зокрема, для всіх відвідувачів доступний перегляд товарів, а авторизовані користувачі без прав адміністратора додатково можуть надсилати повідомлення до адміністрації через форму зворотного зв'язку. В той же час, Адміністратор отримує зручний функціонал для наповнення, редагування та оновлення Web-каталогу. При цьому, незалежно від ролі користувача, створений Web-інтерфейс забезпечує комфортну роботу з каталогом.

3.3 Аналіз створеного Web-ресурсу та перспективи його подальшого розвитку

Створений Web-ресурс має сучасну архітектуру, гнучку структуру і надає користувачам зручні та інтуїтивно зрозумілі механізми взаємодії з каталогом товарів. Web-каталог реалізовано як SPA-додаток, що забезпечує швидке виконання необхідних дій, без необхідності перезавантаження сторінки. Така реалізація значно скорочує час на виконання операцій, покращуючи

користувацький досвід, створюючи у користувача враження роботи з повноцінним десктопним додатком. Це особливо важливо під час інтенсивної роботи з вмістом Web-каталогу - наприклад, під час редагування, додавання або видалення великої кількості товарів.

Інтерфейс Web-каталогу розроблено з використанням технології Flexbox, що забезпечує базову адаптивність, даючи змогу коректно відображати вміст на великій кількості пристроїв. Це робить Web-ресурс більш доступним для різних користувачів, оскільки базова підтримка різних форм-факторів пристроїв забезпечує зручність взаємодії незалежно від платформи, що особливо актуально в умовах постійного зростання частки мобільного трафіку. При цьому, візуальні компоненти розробленого Web-інтерфейсу є логічно і рівномірно розподіленими з дотриманням принципів юзабіліті, завдяки чому користувач може легко орієнтуватися на сторінці і швидко знаходити потрібну інформацію або необхідні інтерактивні елементи.

В той же час, функціональні можливості створеного Web-ресурсу передбачають розмежування прав доступу на основі ролі користувача, яка визначається після авторизації через JWT-токен. Це дає змогу реалізувати декілька сценаріїв взаємодії з інформаційною системою, залежно від ролі користувача. Зокрема, для Адміністратора доступна спеціальна форма додавання і редагування товарів, що включає можливість прикріплення зображень, вказівки ціни та опису товару. При цьому зареєстрований користувач, який не володіє адміністративними правами, може переглядати каталог товарів і надсилати повідомлення через форму зворотного зв'язку. Цей елемент, в свою чергу, виконує роль базового засобу комунікації між користувачами та адміністрацією сайту, що забезпечує не тільки отримання зворотного зв'язку, а й сприяє потенційному розвитку клієнтської бази. Отримана інформація може використовуватися для вдосконалення асортименту, підвищення якості обслуговування та оперативного реагування на потреби аудиторії. При цьому, доступність форми зворотного зв'язку тільки авторизованим користувачам, дає

змогу підтримувати певний рівень безпеки та достовірності одержуваної інформації.

В подальшому функціональні можливості створеного Web-каталогу можуть бути суттєво розширені. Серед пріоритетних напрямів розвитку можна виділити реалізацію системи фільтрів та функціоналу кошика для покупок з можливістю оформлення замовлення та інтеграцією з зовнішніми платіжними сервісами та службами доставки. Це дасть змогу перетворити створений Web-ресурс з довідкової системи на повноцінну платформу електронної комерції.

Також, доцільно розглянути можливість створення особистого кабінету користувача з історією переглядів, відстеженням замовлень та системою сповіщення про нові товари, акції або зміни в асортименті товарів, що може підвищити залученість користувачів і вірогідність повторного відвідування Web-ресурсу. Пори цьому слід врахувати можливість під'єднання систем авторизації через сторонні сервіси (Google, Facebook та ін.), що дасть змогу спростити вхід до системи та підвищити довіру з боку користувачів. Така реалізація також підвищить безпеку за рахунок використання перевірених протоколів і механізмів захисту даних.

Важливим аспектом також є поліпшення адміністративного інтерфейсу, у якому в подальшому може бути доцільно реалізувати автоматичне отримання повідомлень від користувачів та надсилання відповідей, панель управління замовленнями та вбудовану аналітику продажів, яка може передбачати експорт даних у форматі CSV або Excel для подальшого узагальнення, обробки та аналізу. Це зробить роботу адміністратора продуктивнішою і спростить виконання рутинних операцій.

Таким чином, створений Web-каталог є не тільки прикладом сучасного SPA-додатку, а й якісною основою для побудови повноцінної платформи електронної комерції. Його архітектура та реалізація демонструють відповідність актуальним вимогам у сфері Web-розробки, а закладений потенціал для розширення робить цей Web-ресурс придатним для інтеграції як у рамках малого бізнесу, так і в більш великих комерційних проєктах.

Висновки до розділу 3

Розроблений Web-каталог являє собою повнофункціональний SPA-додаток, інтерфейс якого адаптується під різні ролі користувачів, залишаючись зручним та інтуїтивно зрозумілим. Залежно від рівня доступу, користувачі можуть здійснювати перегляд товарів, взаємодіяти з формою зворотного зв'язку або керувати вмістом каталогу. Такий підхід робить систему зручною та ефективною як для потенційних клієнтів, так і для Адміністратора.

Адаптивна система розмежування прав доступу надає Адміністратору розширений функціонал для додавання та актуалізації інформації в каталозі, на відміну від звичайних користувачів, які можуть вільно переглядати представлені товари без можливості редагування. При цьому, незалежно від ролі, користувач отримує логічно організований зручний інтерфейс, що забезпечує комфортну роботу з Web-каталогом.

Таким чином, створений Web-каталог являє собою якісну основу, на базі якої в подальшому можлива побудова повноцінної системи електронної комерції, яка може стати зручним інструментом цифровізації бізнесу.

ВИСНОВКИ

Розробка Web-каталогу постачальника сонячних батарей охоплює не тільки технічну реалізацію, а й аспекти взаємодії з кінцевим користувачем. У сучасних умовах такі Web-ресурси виконують подвійну функцію - вони слугують інструментом електронної комерції й одночасно виступають у ролі платформи для інформаційної підтримки клієнтів. Саме тому особливого значення набуває якісний UX/UI-дизайн Web-додатку, від якого безпосередньо залежить зручність, зрозумілість і ефективність користувацького інтерфейсу.

Поділ архітектури проєкту на серверну (backend) та клієнтську (frontend) частини, сприяє спрощенню розробки та подальшого масштабування. Зокрема, такий модульний підхід полегшує впровадження необхідних функцій і адаптацію під нові бізнес-потреби. При цьому, створена інформаційна модель досить точно відображає структуру і логіку функціонування Web-каталогу та є якісним підґрунтям для реалізації його клієнтської та серверної частин.

Ключове значення при створенні Web-каталогу має правильний вибір технологічного стека. При цьому, використання зв'язки Node.js і Vue.js дає змогу реалізувати сучасний SPA-додаток з адаптивним інтерфейсом, а використання реляційної системи управління базами даних MySQL забезпечує високу надійність зберігання і швидкий доступ до інформації. Комбінація цих інструментів дає змогу створити якісний Web-ресурс, що відповідає сучасним вимогам до засобів електронної комерції.

Застосування сучасних підходів до Web-розробки забезпечує високу продуктивність і стандартизовану взаємодію між компонентами системи. Зокрема, використання фреймворку Vue.js дозволяє реалізувати клієнтський Web-інтерфейс як набір взаємопов'язаних компонентів, кожен з яких відповідає за окремий функціонал, що спрощує як розробку, так і подальше тестування Web-додатку. При цьому, розроблені програмні компоненти в сукупності формують якісний сучасний Web-ресурс, здатний ефективно виконувати функції електронного каталогу.

Розроблений Web-каталог являє собою повнофункціональний SPA-додаток, інтерфейс якого адаптується під різні ролі користувачів і залишається інтуїтивно зрозумілим незалежно від рівня доступу. При цьому, створена гнучка система розмежування прав доступу (залежно від ролі користувача) забезпечує безпечну та зручну роботу з каталогом.

Створений Web-каталог може бути розвинений у повноцінну платформу електронної комерції та стати ефективним інструментом цифровізації бізнес-процесів, що охоплює не тільки управління асортиментом, а й обробку замовлень, інтеграцію з платіжними системами, ведення клієнтської бази та аналітику продажів. При цьому розширення функціоналу дасть змогу адаптувати Web-ресурс під реальні потреби бізнесу та створити якісну основу для подальшого розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Загарій, В. К. Відновлювана енергетика: тенденції розвитку у світі та Україні. Науковий вісник Ужгородського національного університету : серія: Міжнародні економічні відносини та світове господарство. Ужгород: Видавничий дім "Гельветика", 2021. Вип. 36. С. 70-75.
2. Паласевич М. Б., Мурдза П. А. Відновлювана енергетика як фактор підвищення енергоефективності національної економіки. Здобутки економіки: перспективи та інновації. 2024. №. 8. С. 1-14.
3. Жутова М. М. Сонячна енергетика у Харківській області [Електронний ресурс]. Безпека людини у сучасних умовах : зб. доп. 16-ї Міжнар. наук.-метод. конф., 6-7 грудня 2024 р. Нац. техн. ун-т "Харків. політехн. ін-т". Харків : НТУ "ХПІ", 2024. С. 35-37.
4. Краус К. М., Краус Н. М., Манжура О. В. Електронна комерція та інтернет-торгівля : навч. посіб. Київ, 2021. 454 с.
5. Шведа Н. М., Краузе О. І. Електронна комерція: сучасний стан та стратегії розвитку. Міжнародний науковий журнал «Інтернаука» : серія: Економічні науки. 2024. №. 2 (82). С. 35-42.
6. Яшкіна О. І., Савельєв М. С. Стратегії вибору інструментів інтернет-маркетингу в залежності від бізнес-моделей підприємств. Маркетинг і цифрові технології. 2021. № 2 Т. 5. С. 75-83.
7. Карпій О. П., Виноградська Ю. А. Веб-сайт підприємства як засіб управління маркетинговою діяльністю. Scientific notes of Lviv University of Business and Law. 2022. Т. 34. С. 213-219.
8. Каган І. В, Костючко С. М. CRM-системи як інструмент підвищення ефективності бізнесу. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2023. № 52. С. 5-9.
9. Кобилюх О., Гірна О. Сучасні підходи до логістичного обслуговування клієнтів на основі використання CRM-системи. Академічні візії. 2022. №10-11. С. 3-12.

10. Федушко С., Андріяна Н. Сучасний веб-сайт: перспективи та тенденції розвитку. Інформація, комунікація, суспільство 2022 : Матеріали 11-ї Міжнародної наукової конференції ICS-2022. Львів: Видавництво Львівської політехніки, 2022. С. 144-145.

11. Ярхо С.Я. Використання CMS-систем при створенні сучасних мультимедійних веб-ресурсів. Мультимедійні технології в освіті та інших сферах діяльності: науково-практична конференція з міжнародною участю. 2 листопада 2021 р. К.: НАУ, 2022. С.134-137.

12. Потрашкова Л. В. Використання cms wordpress як інструменту цифрового. Поліграфічні, мультимедійні та web-технології : матеріали Молодіжної школи-семінару ІХ Міжнар. наук.-техн. конф., 14-28 травня 2024 р. Харків : ТОВ «Друкарня Мадрид», 2024. Т. 2. С. 64–66.

13. Юстименко Є. А., Труханська В. О., Зелінська О. В. Створення інтернет-магазину з використанням CMS «WordPress». Прикладні інформаційні технології : матеріали ІV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Прикладні інформаційні технології»: збірник наукових праць. Вінниця, ДонНУ імені Василя Стуса, 2023. С. 201-202.

14. Бойко К. О., Розломій І. О. Сучасні клієнтські фреймворки мови JavaScript для розробки web-додатків. Матеріали V Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології»: (19–20 травня 2022 р.). Кропивницький: ЦНТУ, 2022. С. 65-66.

15. Савіцький А. В., Вовк Р. Б. Сучасні тенденції front-end розробки програмного забезпечення. «Сучасна молодь в світі інформаційних технологій»: матеріали ІІ Всеукр. наук.-практ. інтернет-конф. молодих вчених та здобувачів вищої освіти, присвяченої Дню науки (14 травня 2021р., м. Херсон). Херсон: Книжкове видавництво ФОП Вишемирський В.С., 2021. С. 57-58.

16. Паздрій І. Порівняльний аналіз систем розробки програмного забезпечення на основі фреймворків. Вісник хмельницького національного університету. Технічні науки. Хмельницький, 2023. Т. 317. С. 155-161.

17. Рекомендації щодо правил ергономічного розміщення відомостей на веб-сторінці / Р. Лешко та ін. Наука і техніка сьогодні. 2025. № 3(44). С. 1216-1228.
18. Гринчак О. В. Моцик Р. В. Веб-технології та дизайн. Вісник хмельницького національного університету. Технічні науки. Хмельницький, 2021. Т. 293. С. 22-26.
19. Гнатишин М. Аналіз сучасних тенденцій розвитку технологій веб-розробки. Матеріали VI Міжнародної студентської науково-технічної конференції «Природничі та гуманітарні науки. Актуальні питання». 2023. С. 132-133.
20. Сорокун С. В., Коваленко Т. В. Особливості процесу тестування веб-сайту. The 2nd International scientific and practical conference «Innovations and prospects of world science» (October 6 - 8, 2021) Perfect Publishing, Vancouver, Canada. 2021. С. 191-197.
21. Багрій Р. О., Петровський, С.С. Особливості сучасного тестування веб-додатків. Вісник Хмельницького національного університету. Технічні науки. 2022. № 3. С. 70-74.
22. Булій Д. Ю., Костіков М. П. Використання нового модуля MySQLi для роботи з базами даних на веб-сайті Матеріали 89 Міжнародної наукової конференції молодих учених, аспірантів і студентів «Наукові здобутки молоді – вирішенню проблем харчування людства у XXI столітті» (3-7 квітня 2023 р.) К.: НУХТ, 2023 р. Ч.2. С. 318.
23. Скрипка С. О., Шаров С. В. Порівняння реляційних баз даних для використання в інформаційних системах. Українські студії в європейському контексті: зб. наук. пр. 2024. № 8. С. 284-290.
24. Rajabov S. B. The role of backend and frontend information systems infrastructure. Science and Education. 2023. Т. 4. №. 3. С. 212-216.
25. Sharma M. Full Stack Development with MongoDB: Covers Backend, Frontend, APIs, and Mobile App Development Using PHP, NodeJS, ExpressJS, Python and React Native. BPB Publications, 2022. 358 с.

26. Ткаченко О., Бондар В. Деякі аспекти використання Node.js та MongoDB під час створення освітніх вебзастосунків. Цифрова платформа: інформаційні технології в соціокультурній сфері. 2023. Т. 6. №. 1. С. 76-89.

27. Холод І. П. Розробка методики спрощення інтеграції з мікросервісами та сторонніми API у Node.js додатках. Проблеми експлуатації та захисту інформаційно-комунікаційних систем: Тези науково-практичної конференції (м. Київ, 7 - 9 червня 2023 р.), Національний авіаційний університет. К.: Вид-во НАУ, 2023. С. 117-122.

28. Гафіяк А. М., Чепіга Р.В. Особливості використання Vue.js в розробці комерційних веб додатків. Тези доповідей 73-ї наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів Національного університету «Полтавська політехніка імені Юрія Кондратюка», (21 квітня - 13 травня). Полтава : Національний університет імені Юрія Кондратюка, 2021. Т. 1. С. 454-456.

29. Марадь Ю., Карашецький В. Розроблення вебсистеми обліку продукції підприємств з використанням фреймворків Laravel та Vue.js. Комп'ютерне моделювання та інформаційні технології: матеріали п'ятої науково-практичної конференції студентів, аспірантів та молодих вчених (Львів, 19-21 жовтня 2023р.). Львів: ННІ КНІТНЛТУ України, 2023. С. 243-246.

30. Явдошняк Б., Процик Ю. Розроблення онлайн-платформи для організації подорожей з використанням фреймворків Express та Vue.js. Комп'ютерне моделювання та інформаційні технології: матеріали п'ятої науково-практичної конференції студентів, аспірантів та молодих вчених (Львів, 19-21 жовтня 2023р.). Львів: ННІ КНІТНЛТУ України, 2023. С. 246-250.

31. Мироненко А. А. Тенденції використання JavaScript Frameworks. Інформаційне суспільство: проблеми та перспективи : матеріали VII Всеукраїнської науково-практичної конференції (м. Одеса, 20 травня 2022 р.). Одеса, 2022. С.41-43.

32. Kok L. T. Ultimate Nuxt.js for Full-Stack Web Applications: Build Production-Grade Server-Side Rendering (SSR) and Static-Site Generated (SSG)

Vue.js Applications Using Nuxt.js, Node.js, and Composition API. Orange Education Pvt Ltd, AVA., 2024. 394 с.

33. Velos K. MySQL 8 Cookbook: Ready solutions to achieve highest levels of enterprise database scalability, security, reliability, and uptime. GitforGits, 2024. 335 с.

34. Pettit T. Cosentino S. The MySQL Workshop: A practical guide to working with data and managing databases with MySQL. Birmingham: Packt Publishing, 2022. 725 с.

35. Botros S., Tinley J. High Performance MySQL: Proven Strategies for Running MySQL at Scale. 4th Edition. O'Reilly Media, 2022. 523 с.

36. Lentzner R. Getting started with SQL: Exercises with PhpMyAdmin and MySQL. Remylent, 2023. 105 с.

37. Марченко О. О., Дрейс Ю. О. об'єктно-орієнтовані моделі програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи рекламного агентства. The 8 th International scientific and practical conference «Science and technology: problems, prospects and innovations» (May 11-13, 2023). CPN Publishing Group, Osaka, Japan. 2023. С. 153-158.

38. Романенко Т. В., Русіна Н. Г. Моделювання програмних систем мовою UML. Всеукраїнська науково-практична Інтернет-конференція Автоматизація та комп'ютерноінтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку (11-21 березня 2021 року). Черкаси, 2021. С. 144-146.

39. Петрик М. Р., Петрик О. Ю. Моделювання програмного забезпечення: науково-методичний посібник. Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. 200 с.

40. Ковальчук М. О., Войцехівський В. Б. Моделювання інформаційної системи інтернет-магазину. The 21th International scientific and practical conference «Scientists and methods of using modern technologies» (May 30 – June 02, 2023) Melbourne, Australia. International Science Group. 2023. С. 457-459.

ЗГОДА здобувача вищої освіти
Державного університету економіки і технологій про перевірку
кваліфікаційної роботи на прояви академічного плагіату
та розміщення в Репозитарії Університету

Я, Лалак Давид Сергійович, підтримую політику Державного університету економіки і технологій з академічної доброчесності і відкритого доступу.

Засвідчую, що кваліфікаційна бакалаврська робота “Розробка Web-каталогу компанії - постачальника сонячних батарей” виконана самостійно та не містить академічного плагіату. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Державного університету економіки і технологій ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення норм академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Також я поінформований, що відповідно до «Положення про Репозитарій (електронну базу даних) Державного університету економіки і технологій» зазначена робота буде розміщена в Електронному архіві Університету (Репозитарії ДУЕТ). З умовами такого розміщення ознайомлений.

10.06.2025



Д.С. Лалак