

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

**Навчально-науковий інститут кібернетики,
інформаційних технологій та інженерії**

“До захисту допущена”

Зав. кафедри комп'ютерних
наук та прикладної математики

д.т.н., проф. Турбал Ю.В.

« » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

Проектування та розробка веб-додатку

для магазину

(назва теми роботи)

Виконав: Агапов Євген Віталійович
(прізвище, ім'я, по батькові)

студент групи ПЗ-41

Керівник: к.т.н., доц. Климюк Ю. Є
(науковий ступінь, вчене звання, прізвище, ініціали)

_____ (підпис)

_____ (підпис)

ЗМІСТ

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

РЕФЕРАТ	3
ВСТУП	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1. Характеристика інтернет-магазинів освітніх курсів	6
1.2 Аналіз існуючих рішень	7
1.3 Висновки до розділу	14
РОЗДІЛ 2. АНАЛІЗ ЗАСОБІВ ДЛЯ СТВОРЕННЯ СУЧАСНИХ ВЕБ-ДОДАТКІВ	15
2.1. Аналіз сучасних Frontend фреймворків	15
2.2. Аналіз сучасних Backend фреймворків	18
2.3. Висновки до розділу	19
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБ-ДОДАТКУ	21
3.1. Проектування веб-додатку	21
3.2. Розробка веб-додатку	24
ВИСНОВОК	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
ДОДАТКИ	45

РЕФЕРАТ

Кваліфікаційна робота: 45 сторінок, 48 рисунків, 3 таблиці, 1 додаток, 18 джерел.

Мета роботи – проектування та розробка веб-додатку для магазину з продажу освітніх курсів.

Об'єкт дослідження – електронна комерція, зокрема, інтернет-магазини з продажу освітніх курсів.

Предмет дослідження – методи, інструменти та технології, що використовуються для створення ефективних веб-додатків зі зручним користувацьким інтерфейсом.

Методи дослідження – порівняльний аналіз існуючих рішень, аналіз сучасних методів веб-розробки, проектування, моделювання та розробка програмного забезпечення, а також тестування та оцінка функціональності розробленого веб-додатку.

Наукова новизна – дослідження та впровадження сучасних підходів до розробки веб-додатків.

Практична цінність – розробка повнофункціонального веб-додатку для магазину з продажу освітніх курсів, який може бути використаний як базова модель для подібних комерційних проектів.

Ключові слова: *веб-додаток, електронна комерція, магазин, продаж, освітні курси, Next.js, React.js.*

ВСТУП

Сучасні тенденції розвитку інформаційних технологій значно впливають на всі сфери життя суспільства, зокрема й на торговельну діяльність. Електронна комерція - це процес здійснення покупок та продажу товарів або послуг через Інтернет. В умовах зростаючої конкуренції на ринку, електронна комерція стає дедалі популярнішою, оскільки вона надає зручність та економію часу для споживачів. Інтернет-магазини стають важливим інструментом для ведення бізнесу, що дозволяє залучати нових клієнтів та оптимізувати процеси продажу.

Ринок освітніх послуг також зазнає значних трансформацій. Зростає попит на освітні курси, оскільки вони надають можливість здобути необхідні знання та навички у зручний для користувача час та місці. Інтернет-магазини з продажу курсів стають дедалі популярнішими серед користувачів, оскільки вони забезпечують доступ до високоякісних освітніх матеріалів та сприяють ефективному навчанню. Однак, попри існування значної кількості подібних рішень, залишається актуальною проблема забезпечення їхньої зручності та функціональності.

Метою кваліфікаційної роботи є проектування та розробка веб-додатку для магазину з продажу освітніх курсів, який забезпечить користувачам зручний та інтуїтивно зрозумілий інтерфейс, функціональність, інтеграцію з платіжною системою для забезпечення безпеки фінансових транзакцій, а також зручну панель для власника магазину, яка дозволить ефективно керувати магазином.

Об'єктом дослідження є електронна комерція, зокрема, інтернет-магазини з продажу освітніх курсів.

Предметом дослідження є методи, інструменти та технології, що використовуються для створення ефективних веб-додатків зі зручним користувацьким інтерфейсом.

Методи дослідження, які були використані: порівняльний аналіз існуючих рішень, аналіз сучасних методів веб-розробки, проектування, моделювання та розробка програмного забезпечення, а також тестування та оцінка функціональності розробленого веб-додатку.

Наукова новизна кваліфікаційної роботи полягає у дослідженні та впровадженні сучасних підходів до розробки веб-додатків, які дозволять забезпечити приємний інтерфейс додатку, а також зручність і простоту у його використанні.

Практична цінність роботи полягає в розробці повнофункціонального веб-додатку для магазину з продажу освітніх курсів, який може бути використаний як базовий прототип для подібних комерційних проектів. Запропоноване рішення дозволяє значно підвищити ефективність ведення бізнесу в сфері електронної комерції.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Розвиток електронної комерції значно змінили спосіб, яким споживачі взаємодіють з ринком. Інтернет-магазини дозволяють користувачам здійснювати покупки без необхідності фізичної присутності, що робить процес придбання товарів та послуг значно зручнішим та швидшим.

1.1. Характеристика інтернет-магазинів освітніх курсів

За останні роки ринок інтернет-магазинів освітніх курсів зазнав значного зростання. Це зумовлено такими факторами, як:

- Збільшення кількості бажаючих навчатися дистанційно через зручність та гнучкість такого формату.
- Можливість навчатися у власному темпі, що особливо важливо для тих, хто поєднує навчання з роботою або іншими обов'язками.

Інтернет-магазини освітніх курсів – це спеціалізовані веб-додатки, що пропонують користувачам різноманітні освітні матеріали та курси в різних галузях знань. Основними характеристиками таких магазинів є:

- Величезний асортимент курсів з різних галузей знань: від IT і програмування до мистецтва, музики, бізнесу та мов.
- Можливість завантаження матеріалів для офлайн-використання.
- Доступ до курсів 24/7, що дозволяє вчитися у зручний для користувачів час.
- Інтеграція з іншими сервісами та платформами для зручного управління навчальним процесом.
- Персоналізовані рекомендації, що базуються на інтересах та попередніх навчальних досягненнях користувачів.
- Отримання сертифікату після завершення курсу, що може бути корисним для професійного розвитку та підвищення кваліфікації.

Незважаючи на значний прогрес, ринок інтернет-магазинів освітніх курсів стикається з певними викликами. Одним із них є висока конкуренція,

що вимагає від платформ постійного вдосконалення контенту та технологій для залучення нових користувачів. Також важливою проблемою є забезпечення якості навчання, оскільки не всі онлайн-курси відповідають високим стандартам.

Однак, перспективи розвитку цього ринку залишаються позитивними. Зростання попиту на безперервне навчання, розвиток технологій (таких як штучний інтелект та машинне навчання), а також зростаюча необхідність у нових навичках у швидко змінюваному світі забезпечують стабільне зростання ринку інтернет-магазинів освітніх курсів.

Важливим напрямком розвитку є також персоналізація навчального процесу, що дозволяє підлаштовувати курси під індивідуальні потреби кожного користувача. Використання аналізу даних та алгоритмів машинного навчання дозволяє створювати рекомендації щодо навчальних матеріалів, що підвищує ефективність навчання.

Загалом, інтернет-магазини освітніх курсів продовжують розширювати свої можливості, пропонуючи користувачам нові та покращені способи отримання знань та навичок, що є важливим фактором у сучасному світі.

1.2 Аналіз існуючих рішень

Для створення якісного програмного продукту, перед етапом проектування необхідно провести порівняльний аналіз уже існуючих рішень у відповідній галузі. Це дозволить виявити переваги та недоліки цих продуктів і врахувати їх у власній розробці. Для аналізу були обрані такі веб-додатки, які спеціалізуються на продажі освітніх курсів: Coursera [1], Packt Publishing [2], Silhouette [3].

Coursera – одна з найбільших освітніх платформ з мільйонами користувачів по всьому світу. Це глобальна платформа онлайн-навчання, яка пропонує курси, спеціалізації, сертифікати та ступені від провідних університетів та компаній світу.

Перевагами ресурсу є:

- Приємний користувацький інтерфейс (рис. 1.1, рис. 1.2)
- Широкий функціонал
- Використання популярних платіжних систем (рис. 1.3)

Недоліками ресурсу є:

- Навігація на платформі не є інтуїтивно зрозумілою
- Перевантаженість інформацією
- Пошуковий рядок заміщує кнопку «Оглянути» (рис. 1.4)

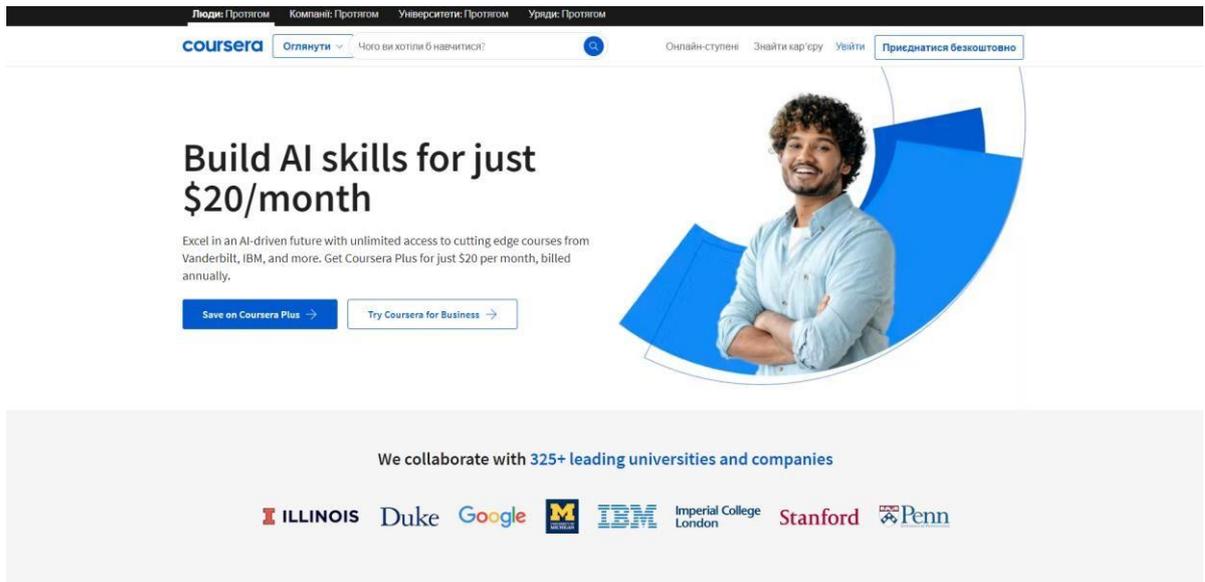


Рис. 1.1. Головна сторінка ресурсу Coursera

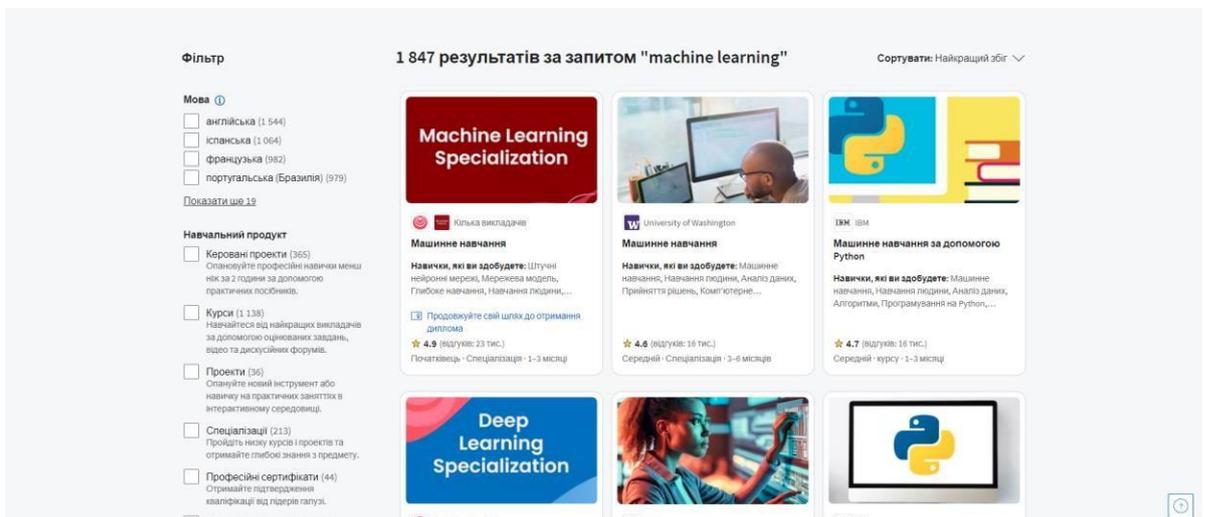


Рис. 1.2. Сторінка з каталогом ресурсу Coursera

Оформлення замовлення

Платіжна інформація

ІМ'Я
Евген Віталійович Агапов

КРАЇНА
Виберіть свою країну

Способи оплати

Card

CARD NUMBER
1234 1234 1234 1234

EXPIRATION DATE
MM / YY

CVC
CVC

Google Pay

PayPal

[Закінчити замовлення](#)

PLUS Підписка на Coursera Plus
Необмежений доступ до 7,000+ курсів, проектів, спеціалізацій і професійних сертифікатів
[Відкрити з кошика](#)

Підписка на Coursera Plus 399 USD USD
33.25 USD на місяць і 12 місяців

Загалом за сьогодні: 399 USD USD

Приміщення з сьогоднішнього дня до оформлення Підписки ваші щорічні виставляються рахунок на суму 399 USD USD на рік (плюс відповідний податок). Ви можете скасувати Підписку в будь-який час, і скасування набере сили наступним розрахунковим періодом. Щоб отримати повне відшкодування, протягом 14 днів після платежу надішліть запит на сторінці «Моя покупка». По закінченні 14 днів кошти за решту можуть не відшкодуватися. Додаткові відомості про умови відшкодування.

Coursera надала інструменти, які дозволили мені досконало вивчити основні поняття у моїй галузі.
—Blake C.

140 мільйонів з гаком
Онлайн-курсів

7,000+
Курсів

Рис. 1.3. Сторінка оплати підписки ресурсу Coursera



Рис. 1.4. Недолік у інтерфейсі ресурсу Coursera

Packt Publishing – це відома платформа для продажу технічних книг, відео та курсів. Основний фокус компанії – інформаційні технології та програмування. Платформа пропонує широкий спектр навчальних матеріалів, які допомагають користувачам здобувати нові знання та покращувати свої навички в галузі ІТ.

Перевагами ресурсу є:

- Зручний користувацький інтерфейс (рис. 1.5, рис. 1.6)
- Широкий функціонал
- Використання популярних платіжних систем (рис. 1.7)

Недоліками ресурсу є:

- Помилки бекенду, що спричиняють повільну швидкість завантаження сторінок (рис. 1.8)

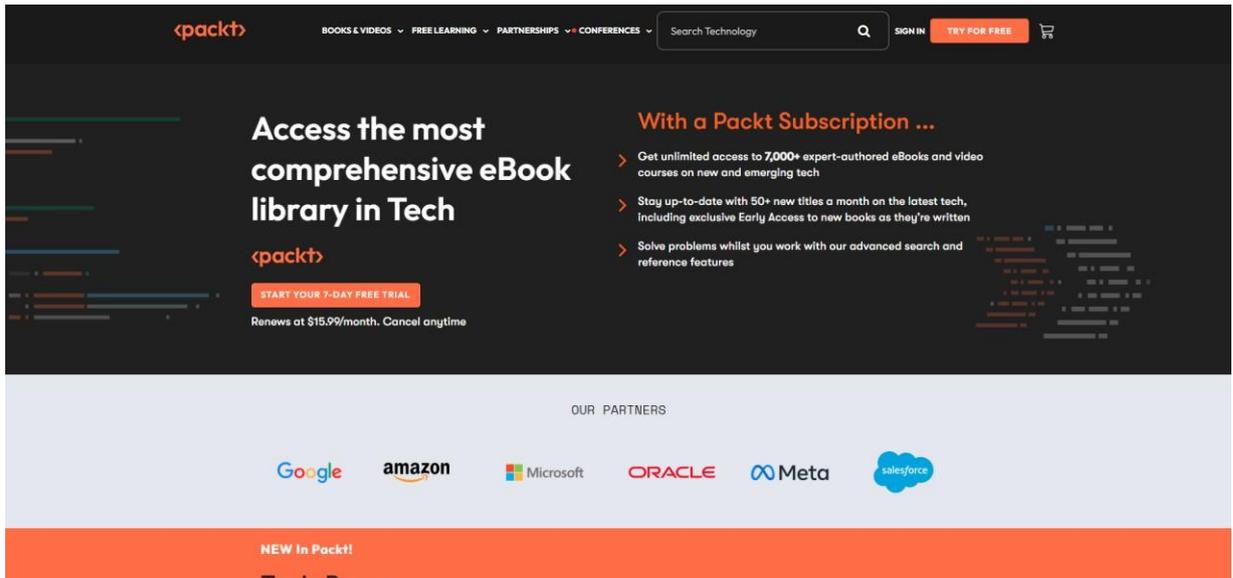


Рис. 1.5. Головна сторінка ресурсу Packt Publishing

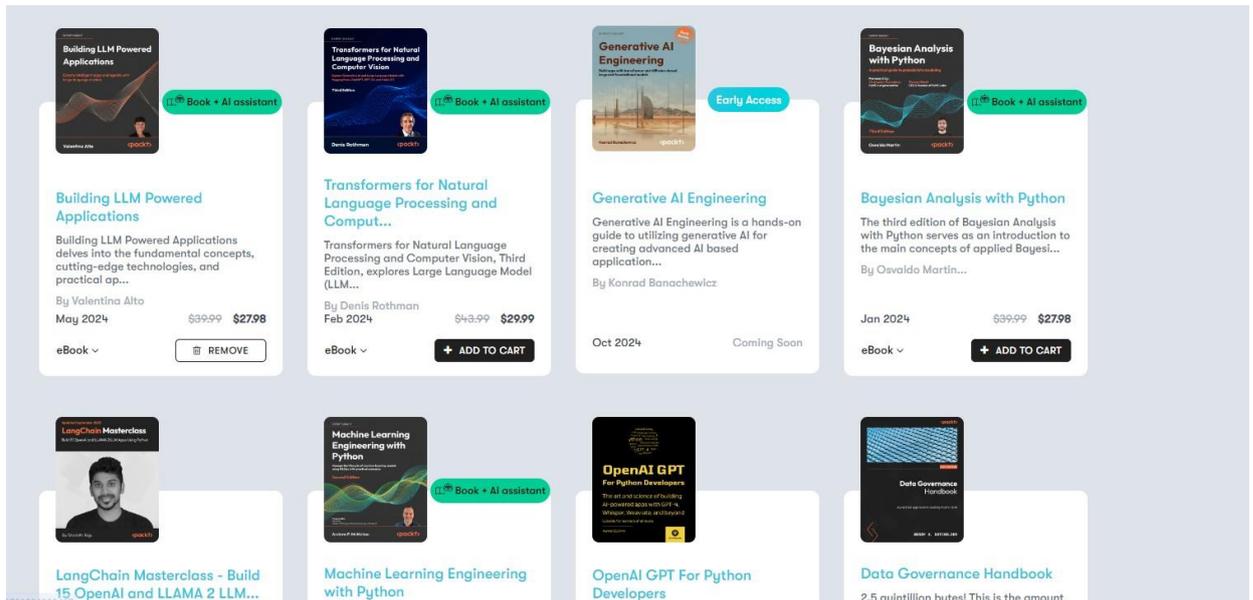


Рис. 1.6. Сторінка з каталогом ресурсу Packt Publishing

← **packt**

TOTAL TO PAY
\$27.98

	Building LLM Powered Applications	\$27.98
	eBook + AI Assistant	\$59.99
Subtotal		\$39.99
Enter discount coupon code		
Total discount		-\$12.01
Total to pay		\$27.98

Contact Details Have a Packt account? **SIGN IN**
Subscribers may be eligible for discount!

Email
Email address

Billing address

First name

Last name

Country or region
Ukraine

Address line 1
Street address

We currently don't deliver to the selected country

Payment details

Card

Card number
1234 1234 1234 1234

Expiration
MM / YY

Security code
CVC

Рис. 1.7. Сторінка оплати товарів ресурсу Packt Publishing

```

GET https://subscription.packtpub.com/fonts/et_wslshbn_wrc_media_worff2est30if_ net::ERR_ABORTED 404 (Not Found)
GET https://subscription.packtpub.com/fonts/et_wslshbn_wrc_media_worff2est306_ net::ERR_ABORTED 404 (Not Found)
GET https://subscription.packtpub.com/fonts/et_wslshbn_wrc_media_worff2est307_ net::ERR_ABORTED 404 (Not Found)
GET https://subscription.packtpub.com/api/users/user/trending-titles 400 (Bad Request)
message: 'User Not Found'
GET https://subscription.packtpub.com/api/cock/user/new-releases 400 (Bad Request)
message: 'User Not Found'
Unchecked runtime.lastError: The message port closed before a response was received.

```

Рис. 1.8. Помилки під час тестування ресурсу Packt Publishing

Silhouette – українська платформа, яка спеціалізується на продажі курсів. Платформа зазначає, що всі освітні матеріали є високоякісними, а усі її викладачі - експерти з багаторічним досвідом і практикою.

Перевагами ресурсу є:

- Інтуїтивно-зрозумілий користувацький інтерфейс (рис. 1.9)
- Використання популярних платіжних систем (рис. 1.10)

Недоліками ресурсу є:

- Застарілий дизайн деяких елементів
- Функціональні помилки при додаванні деяких товарів в кошик (рис. 1.11)

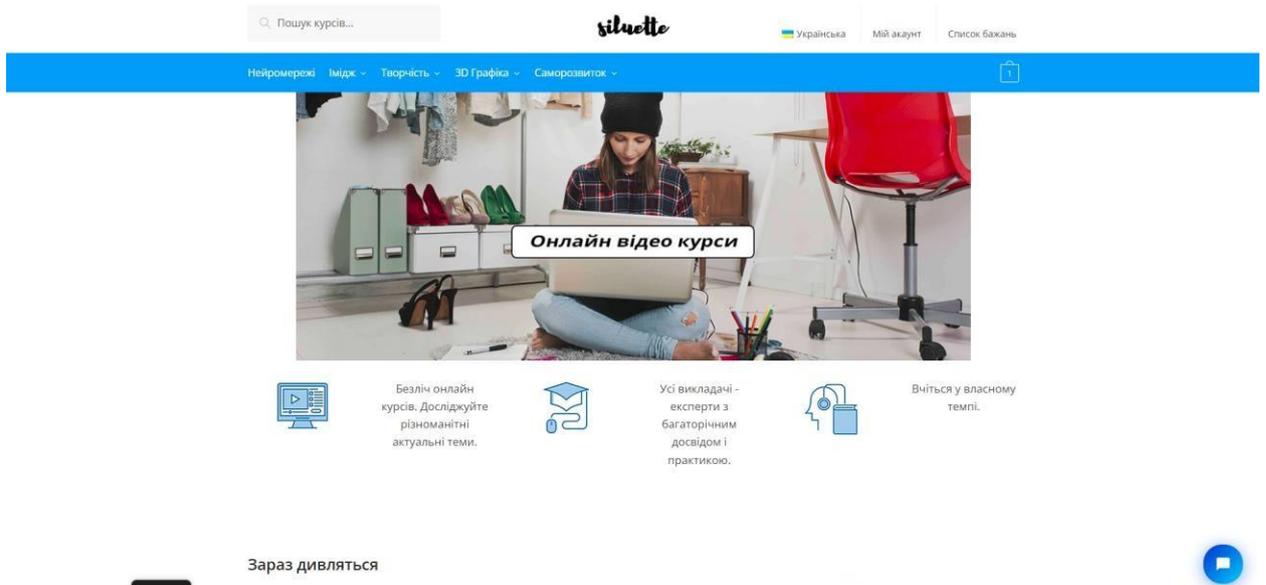


Рис. 1.9. Головна сторінка ресурсу Siluette

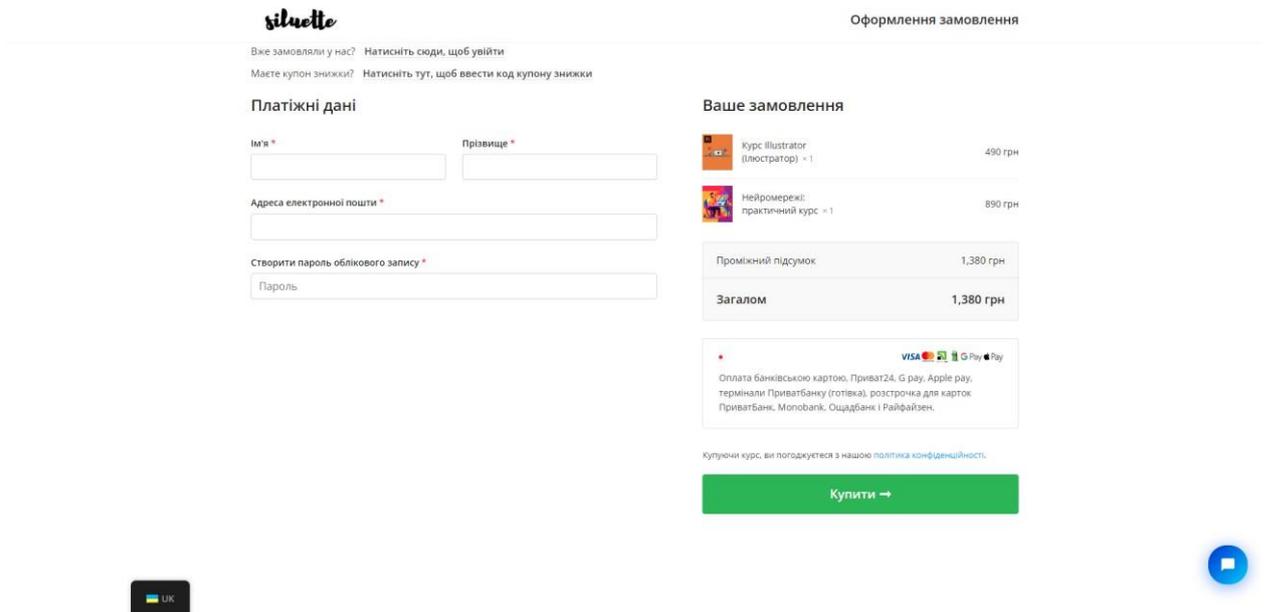


Рис. 1.10. Сторінка оплати товарів ресурсу Siluette

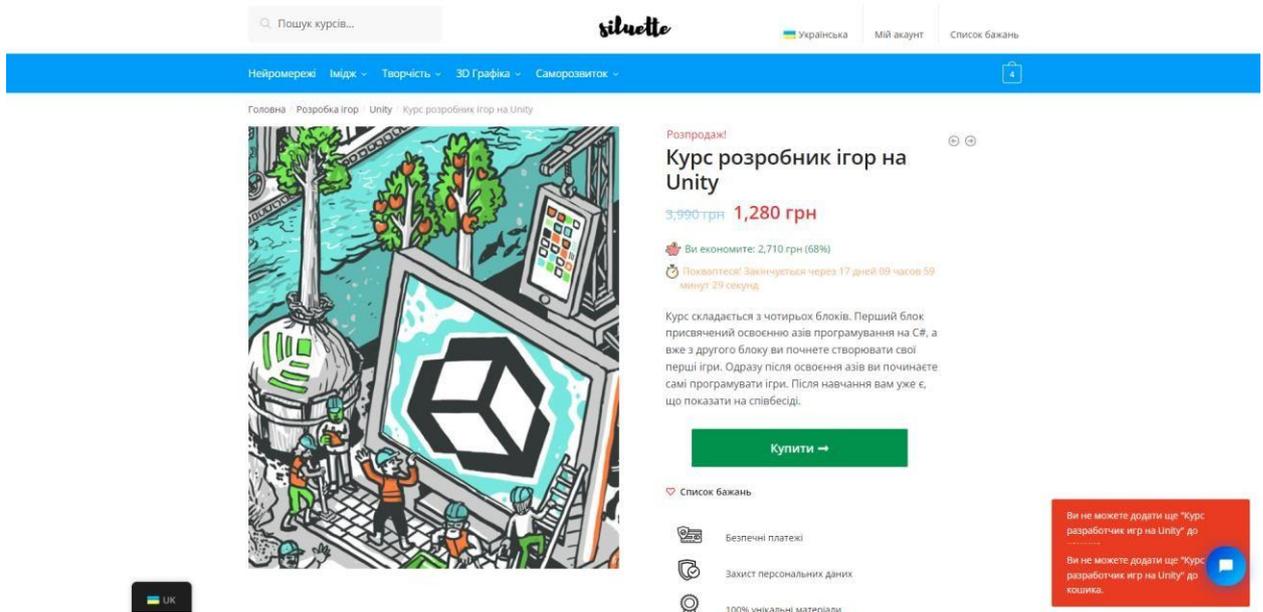


Рис. 1.11. Функціональна помилка ресурсу Siluette

За результатами аналізу ресурсів, які описані вище, була створена порівняльна таблиця у якій було виставлені бали по кожному з параметрів. Результати аналізу представлені у таблиці 1.1.

Таблиця 1.1

Результати аналізу веб-додатків,
які спеціалізуються на продажі освітніх курсів

№	Назва ресурсу	Сучасний користувацький дизайн	Інтуїтивність при користуванні	Відсутність помилок
1	Coursera	1	0.5	0.5
2	Packt Publishing	1	1	0.5
3	Silhouette	0.5	1	0

Результат:

- Coursera – 2 бала
- Packt Publishing – 2.5 бала
- Silhouette – 1.5 бала

1.3 Висновки до розділу

У першому розділі кваліфікаційної роботи було проведено аналіз предметної області електронної комерції та інтернет-магазинів з продажу освітніх курсів. Було виявлено, що інтернет-магазини відіграють важливу роль у сучасному бізнесі, надаючи споживачам зручний доступ до товарів і послуг через Інтернет. Особливу увагу було приділено характеристиці інтернет-магазинів освітніх курсів, які мають свої особливості та переваги, серед яких доступність, інтерактивність і зручність використання.

Для детального вивчення предметної області було проаналізовано три популярні веб-додатки: Coursera, Packt Publishing та Silhouette. Аналіз виявив як переваги, так і недоліки цих ресурсів. Наприклад, Coursera вирізняється широким функціоналом та приємним користувацьким інтерфейсом, але має проблеми з інтуїтивністю навігації. Packt Publishing пропонує зручний інтерфейс і великий вибір навчальних матеріалів, але має технічні проблеми, які впливають на швидкість завантаження сторінок. Silhouette, у свою чергу, характеризується інтуїтивно зрозумілим інтерфейсом, але має застарілий дизайн деяких елементів і функціональні помилки.

За результатами порівняльного аналізу було створено таблицю, де кожен з ресурсів оцінювався за трьома параметрами: сучасний користувацький дизайн, інтуїтивність при користуванні та відсутність помилок. Це дозволило зробити обґрунтовані висновки щодо сильних і слабких сторін існуючих рішень та окреслити напрямки для подальшого вдосконалення розробки веб-додатку.

Таким чином, аналіз предметної області та існуючих рішень дозволив виявити ключові аспекти, які слід врахувати при проектуванні та розробці веб-додатку для магазину з продажу освітніх курсів. Основна увага повинна бути приділена зручності і інтуїтивності користувацького інтерфейсу, використанню сучасних платіжних систем та забезпеченню високої продуктивності і відсутності технічних помилок.

РОЗДІЛ 2. АНАЛІЗ ЗАСОБІВ ДЛЯ СТВОРЕННЯ СУЧАСНИХ ВЕБ-ДОДАТКІВ

Сучасні веб-додатки вимагають високої продуктивності, масштабованості та зручності використання як для користувачів, так і для розробників. Для досягнення цих цілей використовуються різні фреймворки та інструменти, що забезпечують ефективний процес розробки як на фронтенді, так і на бекенді.

Висока продуктивність досягається за рахунок оптимізації коду, використання кешування, а також застосування асинхронних операцій для зменшення затримок у роботі додатку. Масштабованість дозволяє додатку ефективно працювати під час збільшення навантаження, завдяки можливості легко додавати нові сервери або використовувати хмарні рішення.

Зручність використання для користувачів означає інтуїтивно зрозумілий інтерфейс та швидку реакцію на дії, що покращує загальний досвід взаємодії з додатком. Для розробників важлива наявність зручних інструментів для налагодження, тестування та підтримки коду, що дозволяє швидко вносити зміни та вдосконалювати функціонал.

Також важливою є підтримка сучасних стандартів безпеки, що забезпечує захист даних користувачів та надійність роботи додатку. Інтеграція з іншими системами та сервісами дозволяє створювати більш гнучкі та функціональні рішення, що відповідають потребам бізнесу та користувачів.

2.1. Аналіз сучасних Frontend фреймворків

React – це бібліотека JavaScript, що використовується для створення користувацьких інтерфейсів. Вона розроблена компанією Facebook і має велику спільноту розробників. Основні особливості React включають:

- Компонентний підхід: дозволяє створювати повторно використовувані компоненти.

- Віртуальний DOM: оптимізує оновлення інтерфейсу, зменшуючи кількість маніпуляцій з реальним DOM.
- JSX: синтаксис, що дозволяє писати HTML-код безпосередньо в JavaScript.
- Уніфікація клієнтського та серверного коду: за допомогою таких інструментів, як Next.js, що дозволяє використовувати React як для клієнтської, так і для серверної частини додатку.

Vue – це прогресивний фреймворк JavaScript для побудови користувацьких інтерфейсів. Він відомий своєю простотою та легкістю у вивченні. Основні особливості Vue:

- Реактивність: автоматично відслідковує зміни в даних та оновлює DOM.
- Компонентний підхід: дозволяє створювати структуровані та повторно використовувані компоненти.
- Екосистема: включає Vue Router для маршрутизації та Vuex для керування станом.

Angular – це повноцінний фреймворк для розробки веб-додатків, розроблений Google. Він забезпечує потужний набір інструментів для розробки складних додатків. Основні особливості Angular:

- Компонентна архітектура: всі додатки в Angular побудовані з компонентів, які є основними будівельними блоками.
- Двостороннє зв'язування даних: фреймворк забезпечує автоматичний синхронізацію між моделлю та представленням, це спрощує взаємодію між компонентами та даними.
- Форми: Angular пропонує два підходи до роботи з формами: шаблонно-орієнтовані та реактивні форми.
- Підтримка модулів: модульна структура дозволяє розбивати додаток на більш керовані частини.

Основні переваги і недоліки кожного з вищеописаного фреймворку відображені в таблиці 2.1.

Таблиця 2.1

Результати аналізу фронтенд фреймворків

№	Назва фреймворку	Основні переваги	Основні недоліки
1	React	<p>Велике ком'юніті та підтримка від Facebook</p> <p>Висока продуктивність завдяки віртуальному DOM</p> <p>Гнучкість у виборі інструментів та бібліотек</p>	<p>Потребує додаткових бібліотек для повноцінного функціоналу (роутинг, state management)</p> <p>Постійне оновлення документації</p> <p>Висока крива навчання для початківців</p>
2	Vue	<p>Легкість у навчанні та використанні</p> <p>Висока продуктивність завдяки віртуальному DOM</p> <p>Вбудовані інструменти для роутингу та state management</p>	<p>Менше ком'юніті порівняно з React та Angular</p> <p>Обмежена масштабованість для дуже великих проєктів</p> <p>Менше ресурсів та плагінів у порівнянні з React</p>
3	Angular	<p>Двостороннє зв'язування даних</p> <p>Вичерпна документація та великий набір вбудованих інструментів</p> <p>Підтримка Google та велике ком'юніті</p>	<p>Вимагає великої кількості коду для виконання простих завдань порівняно з іншими фреймворками</p> <p>Висока складність та крива навчання</p>

2.2. Аналіз сучасних Backend фреймворків

Express – це мінімалістичний і гнучкий фреймворк для Node.js, який забезпечує потужний набір інструментів для створення веб-додатків та API.

Основні особливості Express:

- Легкість та гнучкість: мінімальний базовий код, що дозволяє додавати лише необхідні компоненти.
- Широка екосистема: підтримка великої кількості сторонніх модулів і middleware.
- Простота у використанні: зручний API та велика кількість прикладів.

Django – це високорівневий фреймворк для Python, який спрямований на швидкий розвиток і чистий дизайн. Він надає багато вбудованих функцій, що знижують кількість рутинної роботи. Основні особливості Django:

- Швидкий розвиток: дозволяє швидко створювати прототипи та повнофункціональні додатки.
- Вбудована панель адміністратора: автоматично генерує панель адміністрування для управління контентом.
- Безпека: включає захист від найбільш поширених атак, таких як SQL-ін'єкції та XSS.

Next.js – це популярний фреймворк для React, який поєднує в собі функції фронтенду та бекенду, що робить його потужним інструментом для створення повноцінних веб-додатків. Основні особливості Next.js:

- Серверний рендеринг: забезпечує швидке завантаження сторінок.
- Сторінки на вимогу: дозволяє створювати сторінки, які завантажуються лише за потреби, що знижує навантаження на сервер.
- API маршрути: включає вбудовані API маршрути, що дозволяє створювати бекенд-функціонал без необхідності використовувати окремий сервер.

Основні переваги і недоліки кожного з вищеприписаного фреймворку відображені в таблиці 2.2.

Таблиця 2.2

Результати аналізу бекенд фреймворків

№	Назва фреймворку	Основні переваги	Основні недоліки
1	Express	Легкість та гнучкість Велика екосистема Простота у використанні	Відсутність строгих структур Потреба у використанні сторонніх модулів
2	Django	Швидкість розробки Універсальність Гнучкість Висока безпека	Повільний розвиток Обмеження в роботі зі складними запитами до бази даних
3	Next.js	Серверний рендеринг Вбудовані API маршрути Сторінки на вимогу	Може бути складним для налаштування для початківців, потребує розуміння як фронтенду, так і бекенду

2.3. Висновки до розділу

Вибір інструментів для розробки сучасних веб-додатків залежить від багатьох факторів, таких як вимоги до продуктивності, масштабованості, досвіду команди та особливостей проекту. На фронтенді популярними рішеннями є React, Vue та Angular, кожен з яких має свої переваги та недоліки. React забезпечує високу продуктивність і гнучкість, але вимагає додаткових бібліотек. Vue відомий своєю простотою у вивченні, тоді як Angular пропонує велику кількість вбудованих інструментів, але має високу криву навчання.

На бекенді популярні фреймворки включають Express, Django та Next.js. Express відзначається своєю гнучкістю та легкістю, Django забезпечує швидкий розвиток та високу безпеку, а Next.js пропонує потужні можливості для серверного рендерингу та API маршрутів, поєднуючи фронтенд і бекенд в одному фреймворку.

Для розробки веб-додатку було вибрано Next.js через його здатність поєднувати функціонал фронтенду і бекенду, що значно спрощує процес розробки та забезпечує високу продуктивність і гнучкість.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РОЗРОБКА ВЕБ-ДОДАТКУ

Процес створення веб-додатку для магазину включає кілька етапів: визначення вимог, створення архітектури додатку, розробка користувацького інтерфейсу, реалізація бекенду, інтеграція з платіжними сервісами та тестування.

3.1. Проектування веб-додатку

На першому етапі проектування веб-додатку необхідно визначити функціональні та нефункціональні вимоги до веб-додатку.

Функціональні вимоги:

1. Користувач (рис. 3.1):

- Перегляд каталогу магазину
- Здійснення покупки
- Скачування файлів товару після успішної покупки
- Отримання чеку на електронну пошту після успішної покупки

2. Адміністратор (рис. 3.2):

- Перегляд статистики магазину
- Додавання товарів в каталог магазину
- Редагування товарів
- Видалення товарів
- Активація товарів
- Деактивація товарів
- Перегляд клієнтських покупок
- Перегляд клієнтської інформації

Нефункціональні вимоги:

- Висока продуктивність та швидкодія.
- Інтуїтивно зрозумілий користувацький інтерфейс.
- Забезпечення безпеки даних користувачів.
- Масштабованість та можливість розширення функціоналу.



Рис. 3.1 Діаграма функціональних вимог для користувача магазину

Магазин освітніх курсів

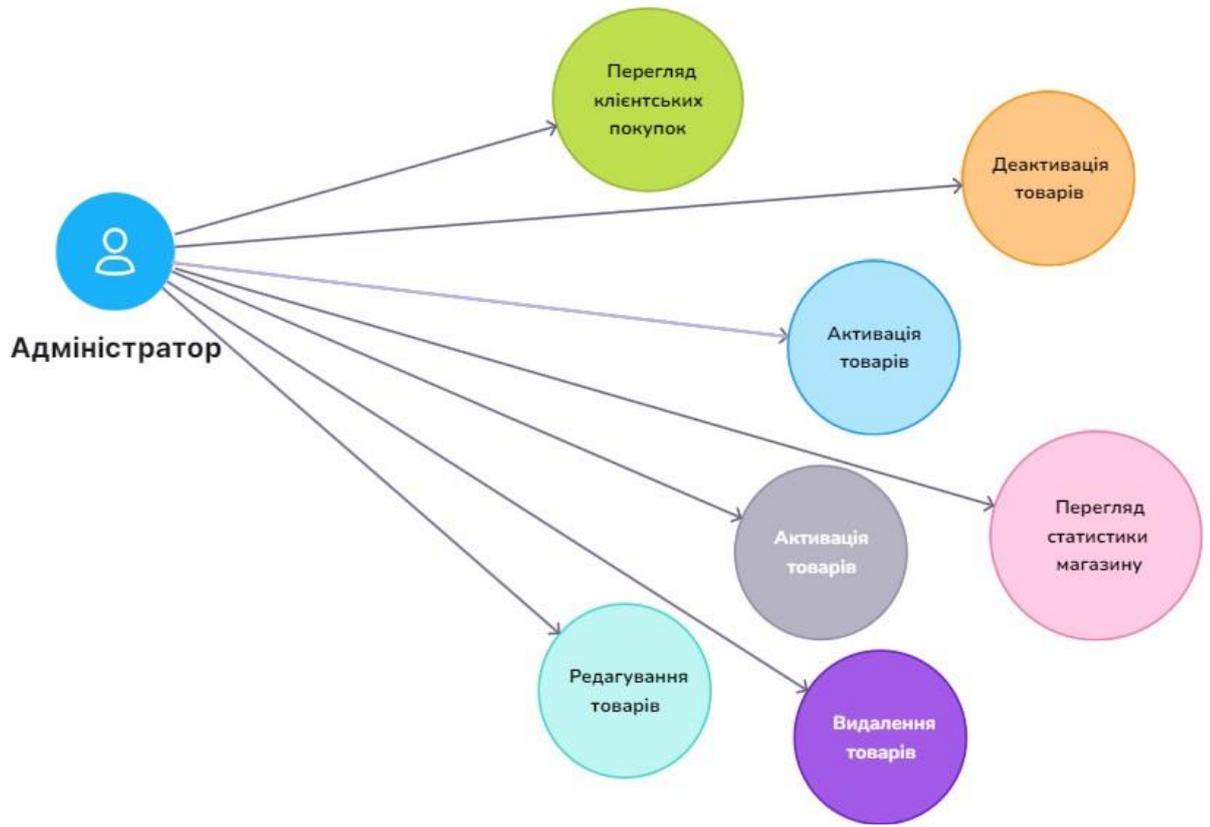


Рис. 3.2 Діаграма функціональних вимог для адміністратора магазину

Архітектура веб-додатку складається з кількох основних компонентів:

- Клієнтська частина: Відповідає за відображення інтерфейсу та взаємодію з користувачем.
- Серверна частина: Обробляє запити клієнтів, взаємодіє з базою даних та виконує логіку додатку.
- База даних: Зберігає інформацію про користувачів, товари, транзакції.

3.2. Розробка веб-додатку

Для початку етапу розробки потрібно налаштувати робоче середовище, завантажити фреймворк Next.js, а також для зручності подальшої розробки веб-додатку підключити ряд бібліотек (рис. 3.4), які допомагають спростити різні аспекти розробки, від взаємодії з базою даних і обробки платежів до управління стилями і створення адаптивних компонентів інтерфейсу:

- Prisma: Бібліотека для зручної взаємодії з базою даних. Вона генерує ORM (Object-Relational Mapping) для зручної роботи з базою даних [4].
- Radix UI: Бібліотека React-компонентів, яка забезпечує побудову якісних, доступних і налаштовуваних інтерфейсів користувача [5].
- React Email: Бібліотека, яка надає доступ до набору компонентів для створення електронних листів з використанням React. Дозволяє зручно створювати та стилізувати електронні листи [6].
- Stripe: Офіційна бібліотека від міжнародної платіжної системи Stripe для інтеграції платіжних функцій у React-додатки. Дозволяє створювати форми оплати та обробляти транзакції [7].
- CVA: Бібліотека для керування класами CSS в залежності від різних станів компонентів. Дозволяє легко змінювати стилі компонентів залежно від умов [8].
- CLX: Бібліотека для об'єднання класів CSS з підтримкою умовного додавання класів. Дозволяє простіше керувати класами в React-компонентах [9].
- Lucide React: Бібліотека, яка надає доступ до великого набору іконок, які легко використовувати та стилізувати [10].
- Resend: Бібліотека для відправки електронних листів [11].
- Tailwind: Бібліотека, яка пропонує широкий каталог класів та інструментів і дозволяє створювати високостилізовані інтерфейси користувача [12].

- Zod: Бібліотека для валідації та парсингу даних у TypeScript та JavaScript. Дозволяє створювати схеми для визначення структури та типів даних, забезпечуючи надійну перевірку на етапі виконання [13].

```

"dependencies": {
  "@prisma/client": "^5.10.2",
  "@radix-ui/react-dropdown-menu": "^2.0.6",
  "@radix-ui/react-label": "^2.0.2",
  "@radix-ui/react-slot": "^1.0.2",
  "@react-email/components": "^0.0.15",
  "@stripe/react-stripe-js": "^2.5.1",
  "@stripe/stripe-js": "^3.0.6",
  "class-variance-authority": "^0.7.0",
  "clsx": "^2.1.0",
  "lucide-react": "^0.341.0",
  "next": "^14.2.3",
  "react": "^18",
  "react-dom": "^18",
  "react-email": "^2.1.0",
  "resend": "^3.2.0",
  "stripe": "^14.18.0",
  "tailwind-merge": "^2.2.1",
  "tailwindcss-animate": "^1.0.7",
  "zod": "^3.22.4"
},
"devDependencies": {
  "@types/node": "^20",
  "@types/react": "^18",
  "@types/react-dom": "^18",
  "autoprefixer": "^10.0.1",
  "eslint": "^8",
  "eslint-config-next": "14.1.0",
  "postcss": "^8",
  "prisma": "^5.10.2",
  "tailwindcss": "^3.3.0",
  "ts-node": "^10.9.2",
  "typescript": "^5"
}

```

Рис. 3.3 Список залежностей

```

"devDependencies": {
  "@types/node": "^20",
  "@types/react": "^18",
  "@types/react-dom": "^18",
  "autoprefixer": "^10.0.1",
  "eslint": "^8",
  "eslint-config-next": "14.1.0",
  "postcss": "^8",
  "prisma": "^5.10.2",
  "tailwindcss": "^3.3.0",
  "ts-node": "^10.9.2",
  "typescript": "^5"
}

```

Рис. 3.4 Список залежностей,
які необхідні тільки для розробки застосунку

Основою веб-додатку є реляційна база даних, яка складається з наступних таблиць: Product (рис. 3.5), User (рис. 3.6), Order (рис. 3.7), та DownloadVerification (рис. 3.8). Кожна таблиця виконує свою роль у зберіганні та обробці даних.

```
model Product {
  id          String          @id @default(uuid())
  name        String
  priceInCents Int
  filePath    String
  imagePath   String
  description  String
  isAvailableForPurchase Boolean @default(true)
  createdAt   DateTime        @default(now())
  updatedAt   DateTime        @updatedAt
  orders      Order[]
  downloadVerifications DownloadVerification[]
}
```

Рис 3.5. Програмна реалізація моделі таблиці
Product

```
model User {
  id          String @id @default(uuid())
  email       String @unique
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
  orders      Order[]
}
```

Рис 3.6. Програмна реалізація моделі таблиці
User

```

model Order {
  id          String  @id @default(uuid())
  pricePaidInCents Int
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt

  userId      String
  productId   String
  user        User    @relation(fields: [userId], references: [id], onDelete: Cascade)
  product     Product @relation(fields: [productId], references: [id], onDelete: Restrict)
}

```

Рис 3.7. Програмна реалізація моделі таблиці
Order

```

model DownloadVerification {
  id          String  @id @default(uuid())
  expiresAt   DateTime
  createdAt   DateTime @default(now())
  productId   String
  product     Product @relation(fields: [productId], references: [id], onDelete: Cascade)
}

```

Рис 3.8. Програмна реалізація моделі таблиці
DownloadVerification

Наступним кроком в реалізації веб-додатку є створення його функціоналу і користувацького інтерфейсу.

Основні компоненти функціоналу веб-додатку та їх реалізація:

- Авторизація адміністратора: Вхід адміністратора в систему виконується за допомогою вводу логіну і паролю, який знаходиться в захешованому виді. Програмна реалізація даного функціоналу відображена на рис. 3.9-3.10.

```

export async function isValidPassword(
  password: string,
  hashedPassword: string
) {
  return (await hashPassword(password)) === hashedPassword
}

Codeium: Refactor | Explain | Generate JSDoc | X
async function hashPassword(password: string) {
  const arrayBuffer = await crypto.subtle.digest(
    "SHA-512",
    new TextEncoder().encode(password)
  )

  return Buffer.from(arrayBuffer).toString("base64")
}

```

Рис. 3.9. Перевірка введеного паролю з уже наявниму системі

```

import { NextRequest, NextResponse } from "next/server"
import { isValidPassword } from "../lib/isValidPassword"

Codeium: Refactor | Explain | Generate JSDoc | X
export async function middleware(req: NextRequest) {
  if ((await isAuthenticated(req)) === false) {
    return new NextResponse("Unauthorized", {
      status: 401,
      headers: { "WWW-Authenticate": "Basic" },
    })
  }
}

Codeium: Refactor | Explain | Generate JSDoc | X
async function isAuthenticated(req: NextRequest) {
  const authHeader = req.headers.get("authorization") || req.headers.get("Authorization")

  if (authHeader == null) return false

  const [username, password] = Buffer.from(authHeader.split(" ")[1], "base64")
    .toString()
    .split(":")

  return (
    username === process.env.ADMIN_USERNAME &&
    (await isValidPassword(
      password,
      process.env.HASHED_ADMIN_PASSWORD as string
    ))
  )
}

```

Рис 3.10. Програмна реалізація перевірки даних для аутентифікації в адміністраторській панелі

- Збереження файлів на сервері: Після того як адміністратор створює або оновлює файловий вміст товару, файл зберігається в каталозі на сервері. Програмна реалізація даного функціоналу відображена на рис. 3.11.

```

Codeium: Refactor | Explain | Generate JSDoc | X
export async function addProduct(prevState: unknown, formData: FormData) {
  const result = addSchema.safeParse(Object.fromEntries(formData.entries()))
  if (result.success === false) {
    return result.error.formErrors.fieldErrors
  }

  const data = result.data

  await fs.mkdir("products", { recursive: true })
  const filePath = `products/${crypto.randomUUID()}-${data.file.name}`
  await fs.writeFile(filePath, Buffer.from(await data.file.arrayBuffer()))

  await fs.mkdir("public/products", { recursive: true })
  const imagePath = `/products/${crypto.randomUUID()}-${data.image.name}`
  await fs.writeFile(
    `public${imagePath}`,
    Buffer.from(await data.image.arrayBuffer())
  )
}

await db.product.create([
  data: {
    isAvailableForPurchase: false,
    name: data.name,
    description: data.description,
    priceInCents: data.priceInCents,
    filePath,
    imagePath,
  },
])

```

Рис 3.11. Програмна реалізація збереження файлів товару на сервері

- Пошук популярних та нових товарів. Програмна реалізація даного функціоналу відображена на рис. 3.12.

```

const getMostPopularProducts = cache(
  () => {
    return db.product.findMany({
      where: { isAvailableForPurchase: true },
      orderBy: { orders: { _count: "desc" } },
      take: 6,
    })
  },
  ["/", "getMostPopularProducts"],
  { revalidate: 60 * 60 * 24 }
)

const getNewestProducts = cache(() => {
  return db.product.findMany({
    where: { isAvailableForPurchase: true },
    orderBy: { createdAt: "desc" },
    take: 6,
  })
}, ["/", "getNewestProducts"])

```

Рис 3.12. Програмна реалізація пошуку популярних та нових товарів

- CRUD-операції: Реалізовані основні функції управління даними «створення, читання, оновлення і вилучення». Програмна реалізація даного функціоналу відображена на рис. 3.13-3.15.

```
export function ProductForm({ product }: { product?: Product | null }) {
  const [error, action] = useFormState(
    product == null ? addProduct : updateProduct.bind(null, product.id),
    {}
  )
  const [priceInCents, setPriceInCents] = useState<number | undefined>(
    product?.priceInCents
  )
}
```

Рис 3.13. Програмна реалізація CRUD-операції

```
import db from "@db/db"
import { PageHeader } from "../../_components/PageHeader"
import { ProductForm } from "../../_components/ProductForm"

Codeium: Refactor | Explain | Generate JSDoc | X
export default async function EditProductPage({
  params: { id },
}): {
  params: { id: string }
}) {
  const product = await db.product.findUnique({ where: { id } })

  return (
    <>
    <PageHeader>Редагувати товар</PageHeader>
    <ProductForm product={product} />
    </>
  )
}
```

Рис 3.14. Програмна реалізація CRUD-операції

```
Codeium: Refactor | Explain | Generate JSDoc | X
export function DeleteDropdownItem({
  id,
  disabled,
}): {
  id: string
  disabled: boolean
}) {
  const [isPending, startTransition] = useTransition()
  const router = useRouter()
  return (
    <DropdownMenuItem
      variant="destructive"
      disabled={disabled || isPending}
      onClick={() => {
        startTransition(async () => {
          await deleteProduct(id)
          router.refresh()
        })
      }}
    >
    Видалити
  </DropdownMenuItem>
  )
}
```

Рис 3.15. Програмна реалізація CRUD-операції

- Надсилання електронного листа: Реалізовано надсилання чеку після покупки та історію покупок у виді електронного листа. Програмна реалізація даного функціоналу відображена на рис 3.16. Приклад роботи наведений у рис. 3.17-3.18.

```
await resend.emails.send({
  from: `Support <${process.env.SENDER_EMAIL}>`,
  to: email,
  subject: "Order Confirmation",
  react: (
    <PurchaseReceiptEmail
      order={order}
      product={product}
      downloadVerificationId={downloadVerification.id}
    />
  ),
})
```

Рис. 3.16. Програмна реалізація відправки повідомлення

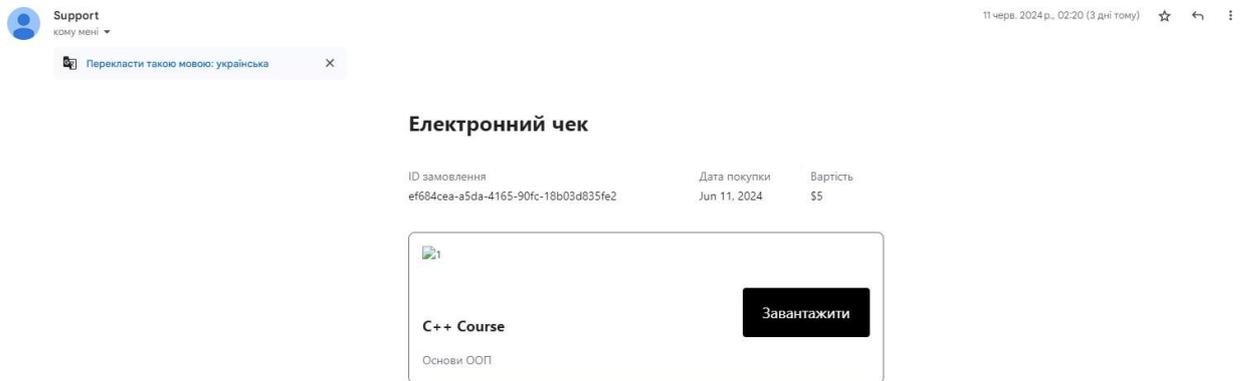


Рис. 3.17. Повідомлення з чеком після успішної оплати

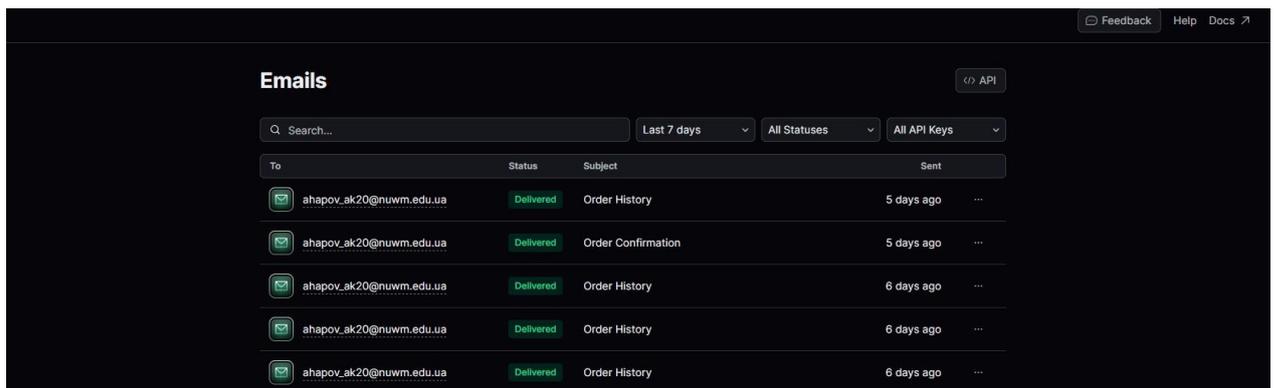


Рис. 3.18. Інформація з ресурсу Resend про успішну відправку повідомлення

Основні компоненти інтерфейсу веб-додатку та їх реалізація:

- ProductCard - це компонент, який відображає інформацію про продукт, включаючи зображення, назву, ціну, опис і кнопку для покупки. Програмна реалізація даного компоненту відображена на рис. 3.19 та рис. 3.20. Рендеринг компоненту зображений на рис. 3.21.

```

1  import { formatCurrency } from "@lib/formatters"
2  import {
3    Card,
4    CardContent,
5    CardDescription,
6    CardFooter,
7    CardHeader,
8    CardTitle,
9  } from "./ui/card"
10 import { Button } from "./ui/button"
11 import Link from "next/link"
12 import Image from "next/image"
13
14 Codeium: Refactor | Explain
15 type ProductCardProps = {
16   id: string
17   name: string
18   priceInCents: number
19   description: string
20   imagePath: string
21 }
22 Codeium: Refactor | Explain | Generate JSDoc | X
23 export function ProductCard({
24   id,
25   name,
26   priceInCents,
27   description,
28   imagePath,
29 }: ProductCardProps) {
30   return (
31     <Card className="flex overflow-hidden flex-col">
32       <div className="relative w-full h-auto aspect-video">
33         <Image src={imagePath} fill alt={name} />
34       </div>

```

Рис 3.19. Програмна реалізація компоненту
ProductCard

```

33 </div>
34 <CardHeader>
35   <CardTitle>{name}</CardTitle>
36   <CardDescription>{formatCurrency(priceInCents / 100)}</CardDescription>
37 </CardHeader>
38 <CardContent className="flex-grow">
39   <p className="line-clamp-4">{description}</p>
40 </CardContent>
41 <CardFooter>
42   <Button asChild size="lg" className="w-full">
43     <Link href={` /products/${id}/purchase `}>Придбати</Link>
44   </Button>
45 </CardFooter>
46 </Card>
47 )
48 }
49

```

Рис 3.20. Програмна реалізація компоненту
ProductCard



Рис. 3.21. Рендеринг компоненту
ProductCard

- ProductCardSkeleton - це компонент, який відображає "скелет" під час завантаження даних продукту. Програмна реалізація даного компоненту відображена на рис. 3.22. Рендеринг компоненту зображений на рис. 3.23.

```

export function ProductCardSkeleton() {
  return (
    <Card className="overflow-hidden flex flex-col animate-pulse">
      <div className="w-full aspect-video bg-gray-300" />
      <CardHeader>
        <CardTitle>
          <div className="w-3/4 h-6 rounded-full bg-gray-300" />
        </CardTitle>
        <CardDescription>
          <div className="w-1/2 h-4 rounded-full bg-gray-300" />
        </CardDescription>
      </CardHeader>
      <CardContent className="space-y-2">
        <div className="w-full h-4 rounded-full bg-gray-300" />
        <div className="w-full h-4 rounded-full bg-gray-300" />
        <div className="w-3/4 h-4 rounded-full bg-gray-300" />
      </CardContent>
      <CardFooter>
        <Button className="w-full disabled size="lg"></Button>
      </CardFooter>
    </Card>
  )
}

```

Рис 3.22. Програмна реалізація компоненту
ProductCardSkeleton

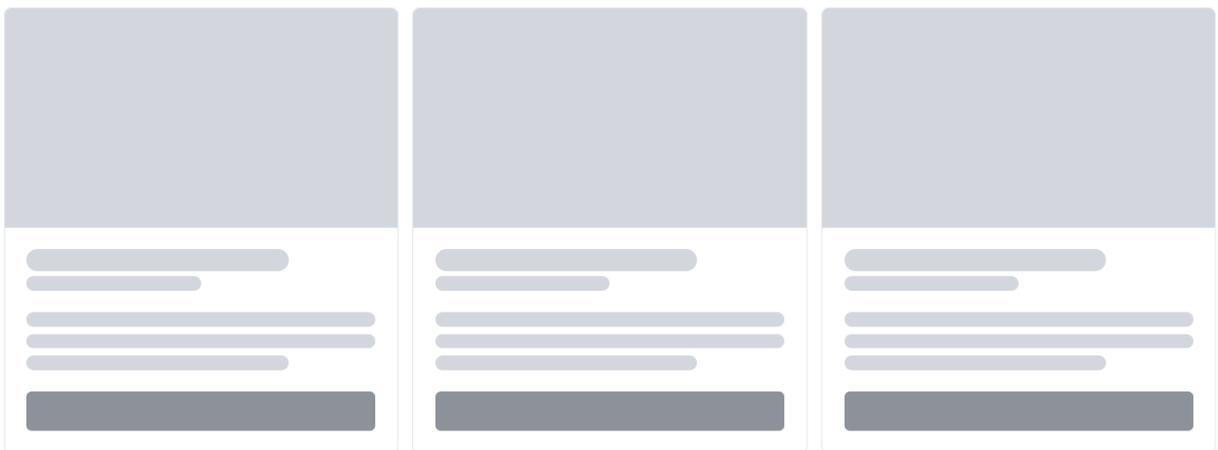


Рис 3.23. Рендеринг компоненту
ProductCardSkeleton

- CheckoutForm - реалізує форму для оплати товару через платіжну систему Stripe. Програмна реалізація даного компоненту відображена на рис. 3.24. Рендеринг компоненту зображений на рис. 3.25.

```

type CheckoutFormProps = {
  product: {
    id: string
    imagePath: string
    name: string
    priceInCents: number
    description: string
  }
  clientSecret: string
}

const stripePromise = loadStripe(
  process.env.NEXT_PUBLIC_STRIPE_PUBLIC_KEY as string
)

Codeium: Refactor | Explain | Generate JSDoc | X
export function CheckoutForm({ product, clientSecret }: CheckoutFormProps) {
  return (
    <div className="max-w-5xl w-full mx-auto space-y-8">
      <div className="flex gap-4 items-center">
        <div className="aspect-video flex-shrink-0 w-1/3 relative">
          <Image
            src={product.imagePath}
            fill
            alt={product.name}
            className="object-cover"
          />
        </div>
        <div>
          <div className="text-lg">
            {formatCurrency(product.priceInCents / 100)}
          </div>
          <h1 className="text-2xl font-bold">{product.name}</h1>
          <div className="line-clamp-3 text-muted-foreground">
            {product.description}
          </div>
        </div>
      </div>
      <Elements options={{ clientSecret }} stripe={stripePromise}>
        <Form priceInCents={product.priceInCents} productId={product.id} />
      </Elements>
    </div>
  )
}

```

Рис 3.24. Програмна реалізація компоненту
CheckoutForm

Checkout

Card number: 1234 1234 1234 1234

Expiration: MM / YY

CVC: CVC

Country: Ukraine

Email: Email address

Придбати - \$10

Рис 3.25. Рендеринг компоненту
CheckoutForm

Весь розроблений інтерфейс продемонстровано на рис. 3.26-3.37.

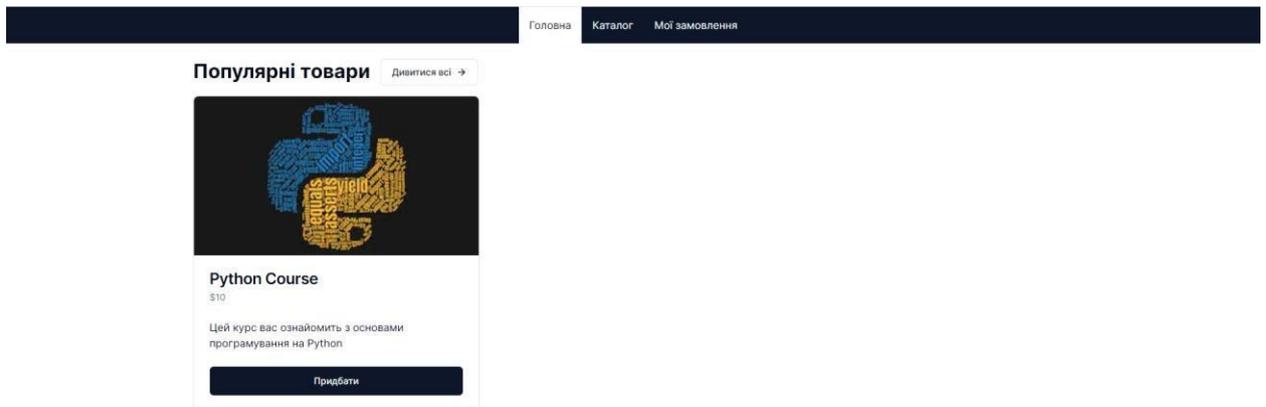


Рис. 3.26. Головна сторінка веб-додатку



Рис. 3.27. Головна сторінка веб-додатку

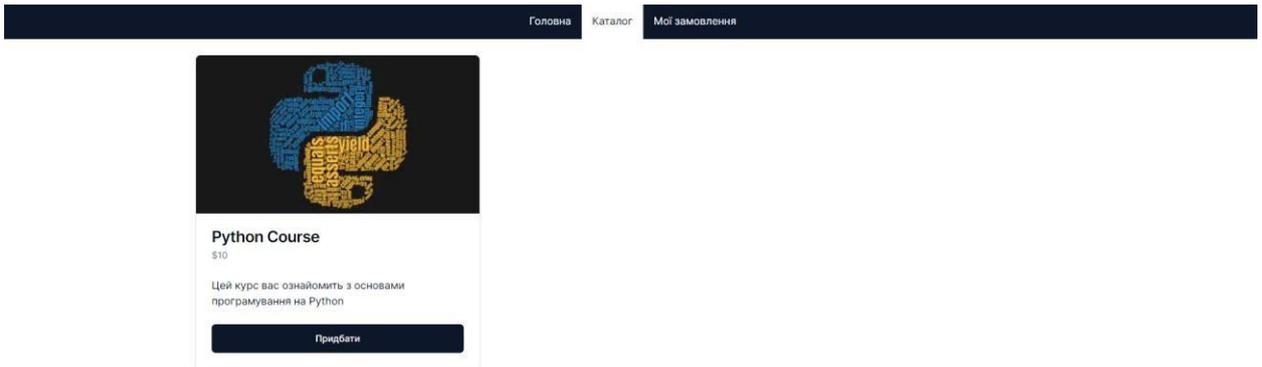


Рис. 3.28. Користувацька сторінка веб-додатку – «Каталог»



Рис. 3.29. Сторінка оплати товару

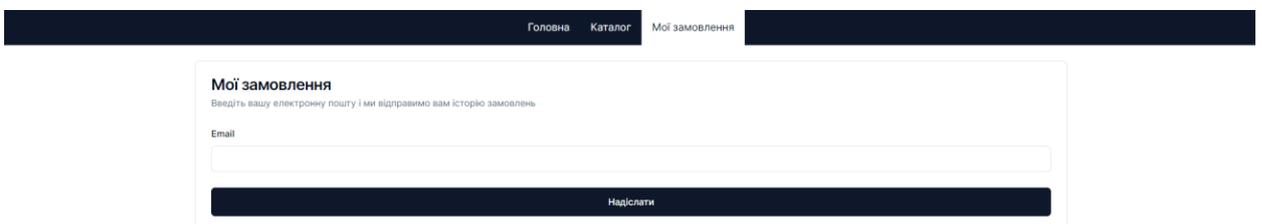


Рис. 3.30. Користувацька сторінка веб-додатку – «Мої замовлення»

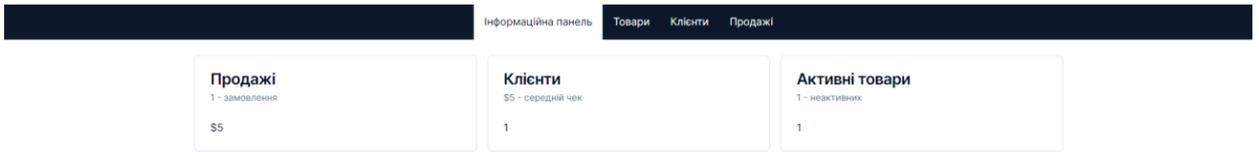


Рис. 3.31. Адміністраторська сторінка веб-додатку – «Інформаційна панель»

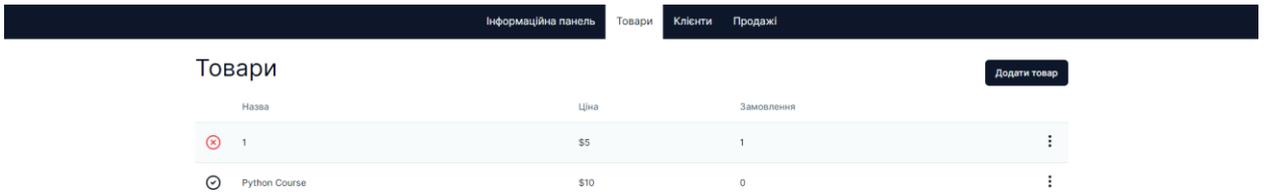


Рис. 3.32. Адміністраторська сторінка веб-додатку – «Товари»

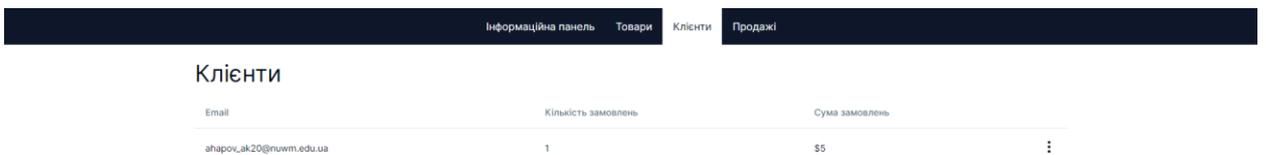


Рис. 3.33. Адміністраторська сторінка веб-додатку – «Клієнти»

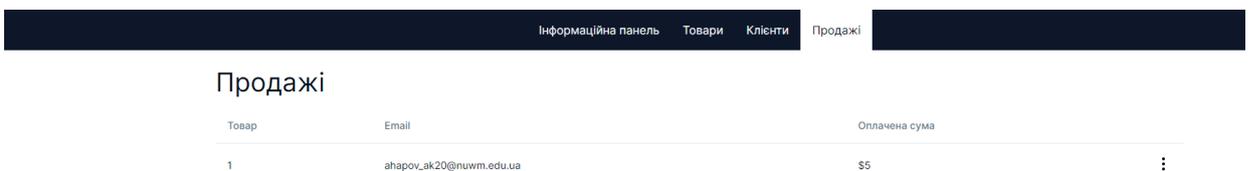


Рис. 3.34. Адміністраторська сторінка веб-додатку – «Продажі»



Рис. 3.35. Інтерфейс користувача при успішній оплаті

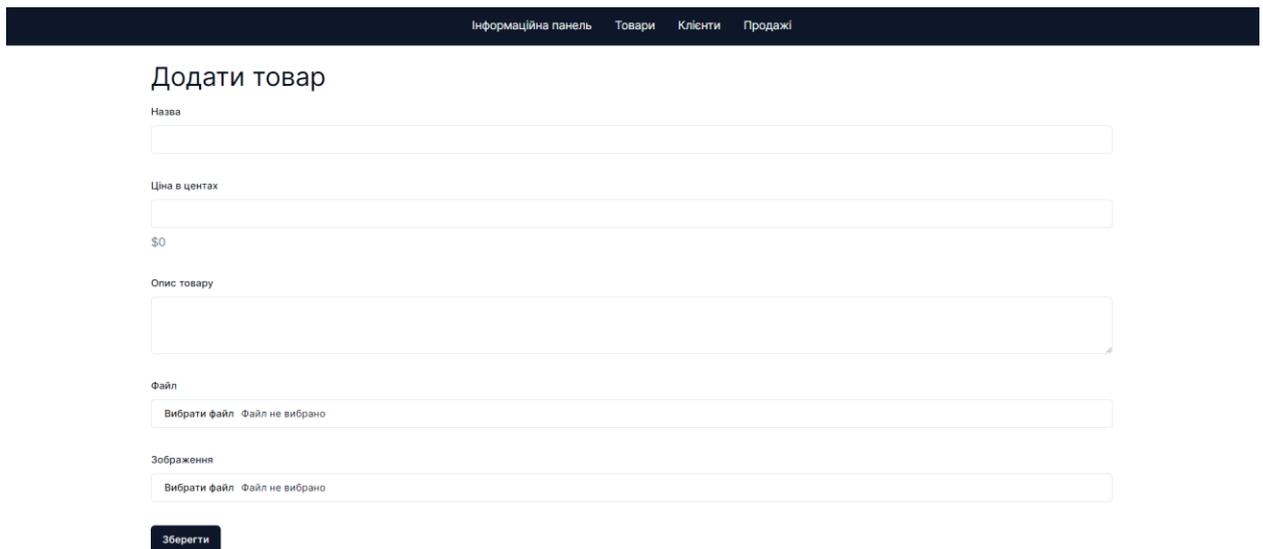


Рис. 3.36. Адміністраторська сторінка для створення
нового товару

Інформаційна панель Товари Клієнти Продажі

Редагувати товар

Назва
Python Course

Ціна в центрах
1000
\$10

Опис товару
Цей курс вас ознайомить з основами програмування на Python

Файл
Вибрати файл Файл не вибрано
products/dc00d922-1fcc-4346-83cb-a074ad2a943e-dipkoma.txt

Зображення
Вибрати файл Файл не вибрано



Рис. 3.37. Адміністраторська сторінка для редагування товару

ВИСНОВОК

У кваліфікаційній роботі було виконано проектування та розробку веб-додатку для магазину з продажу освітніх курсів. Проведений аналіз предметної області та існуючих рішень дозволив визначити ключові аспекти, які слід враховувати при розробці додатку. Було обрано оптимальні технології та інструменти для створення зручного та функціонального веб-додатку.

Розроблений веб-додаток забезпечує користувачам зручний та інтуїтивно зрозумілий інтерфейс, інтеграцію з платіжною системою Stripe для забезпечення безпеки фінансових транзакцій, а також зручну панель для адміністратора для ефективного керування магазином.

Практична цінність роботи полягає в тому, що розроблений веб-додаток може бути використаний як базовий прототип для подібних комерційних проектів у сфері електронної комерції. Подальші дослідження можуть бути спрямовані на розширення функціональності додатку, покращення його продуктивності та впровадження нових технологій для підвищення користувацького досвіду.

Таким чином, мета роботи була досягнута, а поставлені завдання виконані у повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтернет-магазин Packt Publishing [Електронний ресурс] – режим доступу: <https://www.packtpub.com> (дата звернення 05.05.2024 р.)
2. Інтернет-магазин Coursera [Електронний ресурс] – режим доступу: <https://www.coursera.org> (дата звернення 05.05.2024 р.)
3. Інтернет-магазин Silhouette [Електронний ресурс] – режим доступу: <https://silhouette.com.ua> (дата звернення 05.05.2024 р.)
4. Документація Prisma [Електронний ресурс] – режим доступу: <https://www.prisma.io/docs> (дата звернення 10.05.2024 р.)
5. Документація Radix UI [Електронний ресурс] – режим доступу: <https://www.radix-ui.com/docs> (дата звернення 10.05.2024 р.)
6. Документація React Email [Електронний ресурс] – режим доступу: <https://react.email/docs> (дата звернення 10.05.2024 р.)
7. Документація Stripe [Електронний ресурс] – режим доступу: <https://stripe.com/docs> (дата звернення 10.05.2024 р.)
8. Документація CVA [Електронний ресурс] – режим доступу: <https://cva.style/docs> (дата звернення 10.05.2024 р.)
9. Документація CLSX [Електронний ресурс] – режим доступу: <https://github.com/lukeed/clsx> (дата звернення 10.05.2024 р.)
10. Документація Lucide React [Електронний ресурс] – режим доступу: <https://lucide.dev/docs/lucide-react> (дата звернення 10.05.2024 р.)
11. Документація Resend [Електронний ресурс] – режим доступу: <https://resend.com/docs> (дата звернення 10.05.2024 р.)
12. Документація Tailwind [Електронний ресурс] – режим доступу: <https://v2.tailwindcss.com/docs> (дата звернення 10.05.2024 р.)
13. Документація Zod [Електронний ресурс] – режим доступу: <https://zod.dev> (дата звернення 10.06.2024 р.)

14. Документація Next.js [Електронний ресурс] – режим доступу: <https://nextjs.org/docs> (дата звернення 15.05.2024 р.)
15. Документація React.js [Електронний ресурс] – режим доступу: <https://react.dev> (дата звернення 15.05.2024 р.)
16. Griffiths D. React Cookbook: Recipes for Mastering the React Framework / D. Griffiths, D. Griffiths.. – O'Reilly Media, 2021. – 510 с.
17. Schwarzmüller M. React Key Concepts: Consolidate your knowledge of React's core features / Maximilian Schwarzmüller.. – Packt Publishing, 2023. – 590 с.
18. Banks A. Learning React: Modern Patterns for Developing React Apps / A. Banks, E. Porcello.. – O'Reilly Media, 2017. – 307 с.

ДОДАТКИ

Додаток А. Вихідний код проекту

Ознайомитись з вихідний кодом проекту можна за посиланням -
<https://github.com/yevenaharov999/diploma-project>