Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розробка інтернет-магазину ”Air Baking” засобами MERN

Виконав: студент IV курсу, групи СНс-42
спеціальності 122 Комп’ютерні науки
(шифр і назва спеціальності)

_____ Лісовий Н.В.
(підпис) (прізвище та ініціали)

Керівник _____ Дуда О.М.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Марценко С.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент _____ Гащин Н.Б.
(підпис) (прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«_28_» _червня_ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Лісовому Назару Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтернет-магазину "Air Baking" засобами MERN

Керівник роботи Дуда Олексій Михайлович, к.т.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*29*» *квітня* 20*24* року № *4/7-472*

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Наукові публікації, Інтернет-джерела та технічна документація щодо розробки засобів електронної комерції

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналіз предметної області та постановка завдання на розробку інтернет-магазину. 1.1 Аналітичний огляд існуючих рішень реалізації інтернет магазинів. 1.2 Технічне завдання на розробку інтернет-магазину "Air Baking". 1.3 Постановка задачі на розробку інтернет-магазину. 2. Проектна частина. Розробка проекту інтернет- магазину "Air baking" 2.1 Обґрунтування вибору архітектури та розробка структури проекту. 2.2 Проектування функціональних можливостей веб-сторінок. 2.3 Проектування та реалізація структури бази. даних 2.4 Програмування інтернет-магазину 2.5 тестування інтернет-магазину 3. Сервісна частина. Адміністрування та підтримка інтернет-магазину "Air baking". 3.1 Інструкція з розміщення сайту в Інтернеті. 3.2 Інструкція з обслуговування та наповнення інтернет-магазину. 3.3 Інструкція з популяризації та підтримки інтернет-магазину
4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Титульна сторінка. 2. Тема та мета роботи. 3. Актуальність теми. 4. Етапи реалізації проекту. 5. Вимоги до проєкту. 6. Аналіз існуючих рішень. 7. Огляд стеку MERN. 8. Архітектура системи. 9. Проектування БД. 10. Розробка API. 11. Розробка клієнта. 12. Розробка адмін панелі. 13. Заходи безпеки. 14. Тестування. 15. Розгортання в інтернеті. 16. Висновки. 17. Завершальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., доцент кафедри МТ	12.06.2024	15.06.2024

7. Дата видачі завдання 29 січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	
2.	Підбір джерел про способів реалізації проектів	31.01.2024-03.02.2024	
	електронної комерції		
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2024-06.02.2024	
4.	Виконання дослідження щодо розробки інтернет-	07.02.2024-11.02.2024	
	магазину ”Air Baking” засобами MERN		
5.	Оформлення розділу «Аналіз предметної області та	03.06.2024-05.06.2024	
	постановка завдання на розробку інтернет-магазину»		
6.	Оформлення розділу «Проектна частина. Розробка	06.06.2024-08.06.2024	
	проекту інтернет- магазину "Air baking" »		
7.	Оформлення розділу «Сервісна частина.	09.06.2024-11.06.2024	
	Адміністрування та підтримка інтернет-магазину "Air		
	baking" »		
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2024-13.06.2024	
9.	Виконання завдання до підрозділу «Основи охорони	14.06.2024-15.06.2024	
	праці»		
10.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	
11.	Нормоконтроль	18.06.2024-19.06.2024	
12.	Перевірка на плагіат	20.06.2024	
13.	Попередній захист кваліфікаційної роботи	21.06.2024	
14.	Захист кваліфікаційної роботи	29.06.2024	

Студент _____ Лісовий Н.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Дуда О.М.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка інтернет-магазину ”Air Baking” засобами MERN // Кваліфікаційна робота освітнього рівня «Бакалавр» // Лісовий Назар Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СНс-42 // Тернопіль, 2024 // С. 74 , рис. – 28 , табл. – 4, кресл. – 17, додат. – 6, бібліогр. – 42.

Ключові слова: MERN-стек, інтернет-магазин, MongoDB, Express.js, React.js, Node.js, електронна комерція, веб-розробка

Кваліфікаційна робота присвячена дослідженню та розробці інтернет-магазину "Air Baking" засобами MERN. В першому розділі кваліфікаційної роботи висвітлено аналітичний огляд існуючих рішень для реалізації інтернет-магазинів, розроблено технічне завдання для "Air Baking". Розглянуто область застосування, призначення розробки, вимоги до функціональності, програмної документації, техніко-економічні показники, стадії та етапи розробки, а також порядок контролю та прийому. Проаналізовано визначення проблеми, мету, задачі розробки, очікувані результати та критерії оцінки успішності проекту.

В другому розділі кваліфікаційної роботи обґрунтовано вибір архітектури та розроблено структуру проекту інтернет-магазину "Air Baking". Розроблено структуру бази даних, реалізовано серверну та клієнтську частин проекту.

В третьому розділі кваліфікаційної роботи проаналізовано інструкції з розміщення сайту в Інтернеті, обслуговування та наповнення контенту, а також популяризації та підтримки інтернет-магазину.

В четвертому розділі кваліфікаційної роботи розглянуто моделювання та прогнозування небезпечних ситуацій, описано вимоги ергономіки до організації робочого місця оператора ПК.

ANNOTATION

Development of the "Air Baking" Online Store using MERN // Bachelor's Qualification Work // Lisovyi Nazar Volodymyrovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-42 // Ternopil, 2024 // Pages - 74, Figures – 28, Tables – 4, Drawings – 17, Appendices – 6, Bibliography – 42.

Keywords: MERN stack, online store, MongoDB, Express.js, React.js, Node.js, e-commerce, web development

The qualification work is dedicated to the research and development of the "Air Baking" online store using the MERN stack. The first chapter of the qualification work presents an analytical review of existing solutions for the implementation of online stores and develops the technical specification for "Air Baking". It covers the scope of application, the purpose of development, requirements for functionality, software documentation, technical and economic indicators, stages and phases of development, as well as the order of control and acceptance. It analyzes the problem definition, goals, development tasks, expected results, and project success criteria.

The second chapter of the qualification work substantiates the choice of architecture and develops the project structure of the "Air Baking" online store. It also develops the database structure and implements the server and client parts of the project.

The third chapter of the qualification work analyzes the instructions for deploying the website on the Internet, maintaining and updating content, as well as promoting and supporting the online store.

The fourth chapter of the qualification work examines the modeling and forecasting of hazardous situations and describes the ergonomic requirements for organizing the workplace of a computer operator.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

CMS (англ. Content Management System) – система управління контентом.

API (англ. Application Programming Interface) – програмний інтерфейс програми.

MERN - MongoDB, Express.js, React, Node.js.

HTTP (англ. HyperText Transfer Protocol) — протокол передачі гіпертексту.

SEO-оптимізація (Search Engine Optimization) — це процес покращення якості та обсягу трафіку із пошукових систем на вебсайт.

ЗМІСТ

ВСТУП

Розвиток електронної комерції є однією з найважливіших тенденцій сучасного світу, що зумовлено активною цифровізацією суспільства та швидким розвитком інформаційних технологій. Інтернет-магазини стали невід'ємною частиною бізнесу, пропонуючи зручний спосіб здійснення покупок, який значно полегшує процес придбання товарів для споживачів та розширює ринки збуту для підприємців.

Актуальність теми. Вибір технологічного стеку для створення інтернет-магазину має критичне значення для його успішної роботи. Технології MERN (MongoDB, Express.js, React.js, Node.js) забезпечують високий рівень продуктивності, гнучкості та масштабованості, що дозволяє створювати сучасні веб-застосунки з багатим функціоналом та інтуїтивно зрозумілим інтерфейсом. Розробка інтернет-магазину "Air Baking" з використанням MERN-стеку є вдалим вибором для досягнення цих цілей.

Мета і задачі дослідження. Метою кваліфікаційної роботи є розробка інтернет-магазину "Air Baking" з використанням технологій MERN, що забезпечить ефективну і зручну платформу для продажу кондитерських виробів. Для досягнення поставленої мети необхідно вирішити такі задачі:

– провести аналіз існуючих рішень для розробки інтернет-магазинів та визначити переваги і недоліки технологій MERN;

– розробити архітектуру інтернет-магазину з урахуванням сучасних вимог до безпеки, продуктивності та масштабованості;

– створити базу даних для зберігання інформації про продукти, користувачів, замовлення та інші сутності;

– реалізувати серверну частину інтернет-магазину з використанням Node.js і Express.js;

– розробити клієнтську частину інтернет-магазину на базі React.js, забезпечивши інтерактивний і зручний користувацький інтерфейс;

– провести тестування і налагодження системи для виявлення та усунення можливих помилок;

– підготувати інструкції з розміщення, обслуговування та популяризації інтернет-магазину.

Практичне значення одержаних результатів. Розробка інтернет-магазину "Air Baking" засобами MERN має практичне значення як для підприємців, так і для розробників програмного забезпечення. Отримані результати дозволять створити ефективний інструмент для ведення бізнесу в сфері електронної комерції, забезпечивши зручний процес замовлення товарів для клієнтів та оптимізацію внутрішніх бізнес-процесів. Використання сучасних технологій забезпечить високу продуктивність та масштабованість системи, що є важливим для подальшого розвитку бізнесу. Крім того, результати даного дослідження можуть бути використані як основа для подальших наукових досліджень у сфері розробки веб-застосунків та електронної комерції.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ІНТЕРНЕТ-МАГАЗИНУ.

1.1 Аналітичний огляд існуючих рішень реалізації інтернет магазинів

Створення інтернет-магазину може бути досягнуто за допомогою різних підходів, кожен з яких має свої переваги та недоліки. Основні способи реалізації включають використання готових платформ, розробку власного рішення з нуля та комбінований підхід, який поєднує обидва методи [1].

Популярні платформи для створення інтернет-магазинів, такі як Shopify, WooCommerce (плагін для WordPress), Magento та BigCommerce, пропонують швидке та просте рішення для запуску онлайн-бізнесу. Вони ідеальні для тих, хто хоче швидко вийти на ринок з мінімальними технічними знаннями [2]. Процес налаштування інтуїтивно зрозумілий, а платформи надають широкий спектр шаблонів і функцій, що дозволяють створити професійний інтернет-магазин без великих зусиль. Однак вибір платформи для інтернет-магазину є критично важливим кроком, який впливає на всі аспекти ведення бізнесу: від налаштування та дизайну до управління замовленнями та обслуговування клієнтів [3]. Наведено короткий порівняльний аналіз кількох популярних платформ: Shopify, WooCommerce, Magento та BigCommerce.

Shopify є однією з найпопулярніших платформ для створення інтернет-магазинів, відомою своєю простотою у використанні та широким спектром функцій [4]. Дана платформа ідеально підходить для малого та середнього бізнесу, який хоче швидко запустити інтернет-магазин без глибоких технічних знань. На рисунку 1.1 подано адміністративну сторінку платформи електронної комерції Shopify, яка пропонує користувачам інтуїтивно зрозумілий інтерфейс, безліч готових шаблонів та потужний конструктор сторінок. Платформа забезпечує високий рівень безпеки та має вбудовані інструменти для SEO, маркетингу та аналітики. Крім того, Shopify інтегрується з великою кількістю платіжних систем і сторонніх сервісів [5].

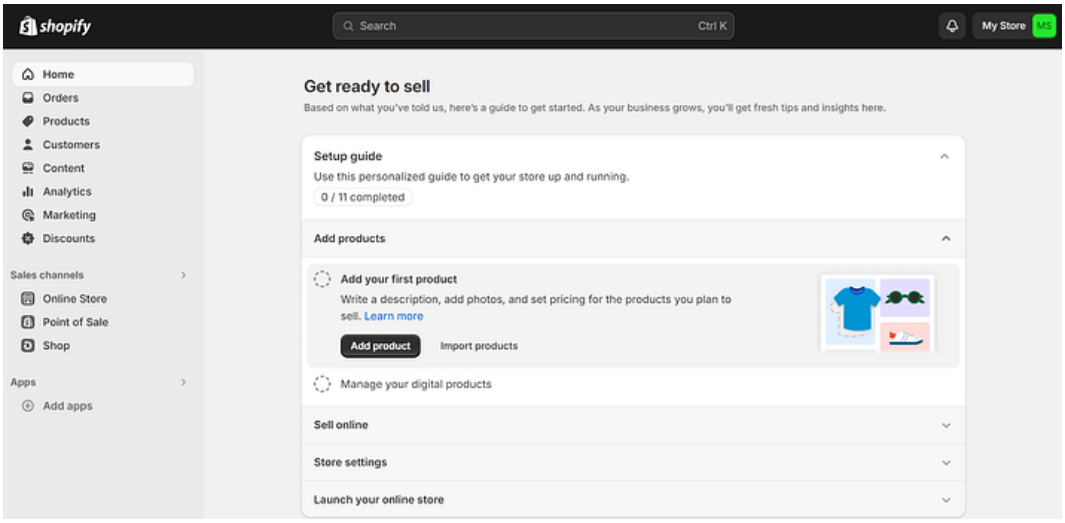


Рисунок 1.1 – Сторінка адміністратора Shopify

Однак Shopify має свої недоліки. Основним є вартість, яка включає місячну підписку та комісії за транзакції. Крім того, можливості для кастомізації обмежені порівняно з відкритими платформами, і для розширення функціональності часто потрібно купувати додаткові плагіни або застосунки.

Платформа WooCommerce – це плагін для WordPress [6], який дозволяє перетворити звичайний сайт на потужний інтернет-магазин. Сторінка адміністратора, подана на рисунку 1.2, інтуїтивно доступна та зручна.

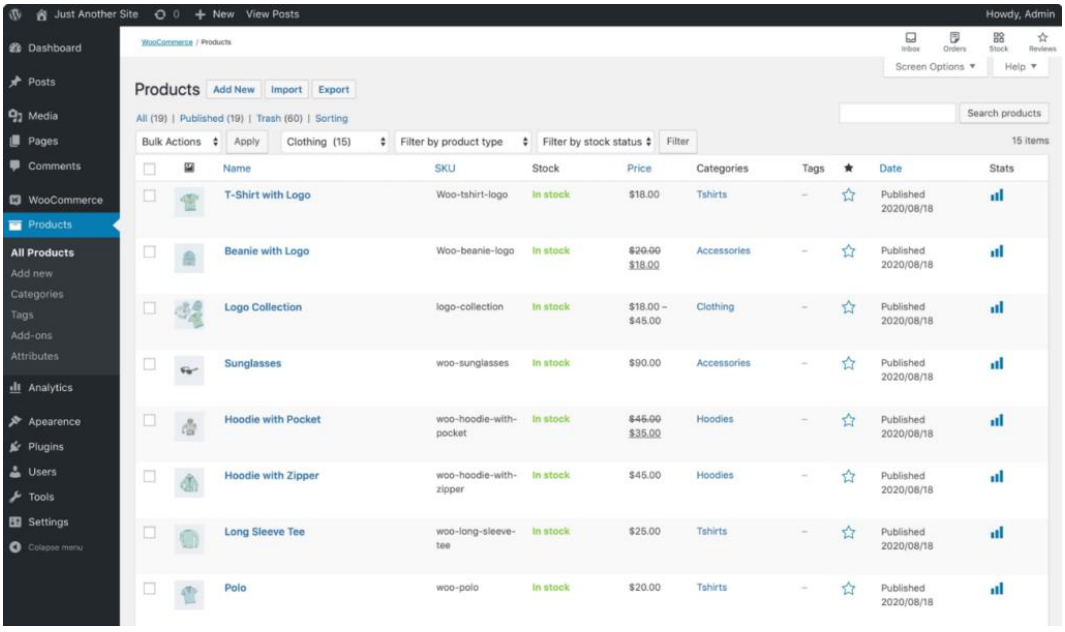


Рисунок 1.2 – Сторінка адміністратора WooCommerce

WooCommerce є відкритою платформою, що надає максимальну гнучкість та можливості для кастомізації, що робить її популярною серед розробників та досвідчених користувачів [7].

Однією з головних переваг WooCommerce є його інтеграція з WordPress, що дозволяє користуватися всіма можливостями цієї CMS. WooCommerce також підтримує широкий спектр тем і плагінів, що дозволяє налаштувати інтернет-магазин під специфічні потреби бізнесу. Платформа не вимагає щомісячних платежів, хоча деякі додаткові функції можуть бути платними.

Недоліки WooCommerce включають складність налаштування для новачків і необхідність технічної підтримки [8]. Крім того, безпека та продуктивність можуть залежати від якості хостингу та правильного налаштування.

CMS Magento є однією з найбільш потужних та гнучких платформ для створення інтернет-магазинів [9]. Вона підходить для середнього та великого бізнесу, який потребує комплексного рішення з розширеними можливостями. У панелі управління CMS Magento, яка подана на рисунку 1.3, існує широкий спектр функцій для управління товарами, замовленнями, клієнтами та маркетингом. Платформа підтримує багатомовність та мультивалютність. Magento також має високий рівень безпеки та масштабованості, що дозволяє обробляти великий обсяг трафіку та замовлень [10].

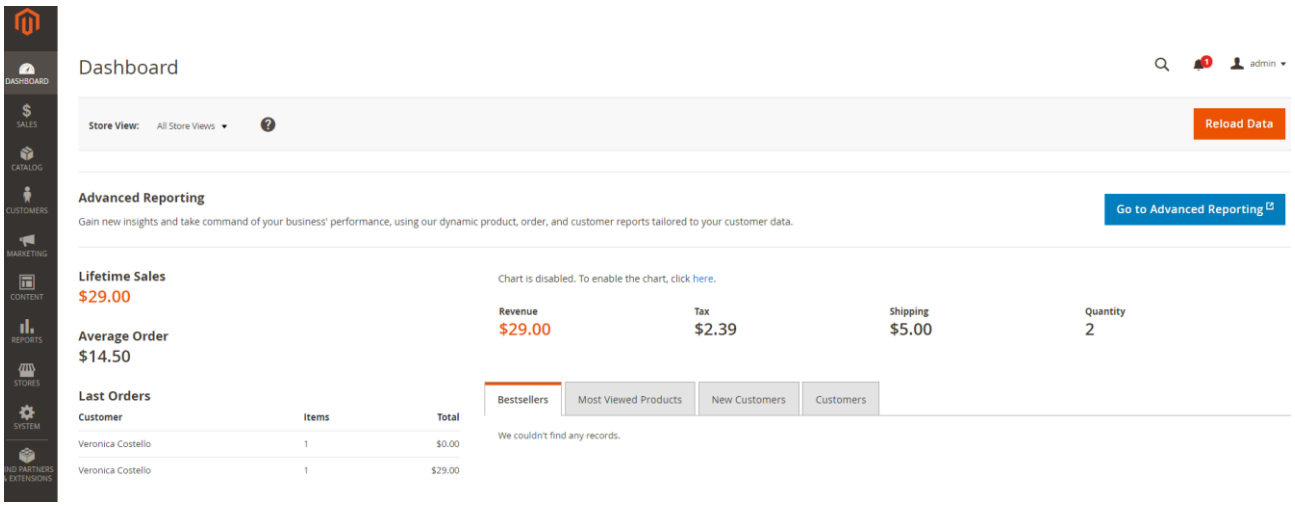


Рисунок 1.3 – Панель управління CMS Magento

Проте Magento є складною у налаштуванні та управлінні, що вимагає професійної технічної підтримки. Вартість може бути високою, особливо для Enterprise-версії, і включає витрати на хостинг, розробку та підтримку [10].

BigCommerce – це SaaS-платформа, яка пропонує повний спектр інструментів для створення та управління інтернет-магазином. Вона орієнтована на малий та середній бізнес, що шукає готове рішення з мінімальними технічними вимогами [11]. BigCommerce забезпечує користувачів безліччю вбудованих функцій, включаючи SEO-оптимізацію, інструменти для маркетингу, аналітику та інтеграцію з багатьма платіжними системами. Платформа також підтримує багатоканальність, дозволяючи продавати товари через різні канали, такі як Amazon, eBay та соціальні мережі.

Одним з основних недоліків BigCommerce є вартість, яка включає щомісячні платежі. Крім того, можливості для кастомізації можуть бути обмежені порівняно з відкритими платформами. Інтеграція з деякими сторонніми сервісами також може вимагати додаткових витрат.

Однією з основних переваг готових платформ є швидкість впровадження [12]. Тобто наявність можливості запустити магазин eCommerce у лічені дні або навіть години. Платформи також забезпечують регулярні оновлення та технічну підтримку, що дозволяє зосередитися на бізнесі, а не на технічних аспектах [13]. Безпека також є сильною стороною, оскільки більшість платформ пропонують вбудовані рішення для захисту даних клієнтів та захисту від шахрайства.

Однак готові платформи мають і свої недоліки. Вартість використання таких платформ може бути значною, оскільки включає місячні або річні підписки, а іноді й комісії за транзакції. Крім того, можливості для кастомізації обмежені, що може бути проблемою для бізнесів з унікальними потребами. Залежність від платформи також може стати серйозним питанням, якщо її політика або технічні вимоги зміняться.

Ще одним варіантом реалізації інтернет магазину можна розглянути комбінований підхід, який би включав в себе використання готової платформи

як основи з подальшою кастомізацією окремих компонентів або інтеграцією кастомних модулів. Таке рішення дозволяє швидко запустити інтернет-магазин з базовими функціями та поступово додавати унікальні можливості. При такому підході можна отримати гнучкість та знизити початкові витрати, поєднуючи переваги готових рішень та кастомної розробки. Тобто можна скористатися швидким впровадженням основних функцій готових платформ і одночасно працювати над унікальними функціями, які би давали переваги бізнесу над конкурентами.

Проте комбінований підхід може бути складнішим у реалізації, оскільки інтеграція кастомних рішень з готовими платформами може потребувати додаткових зусиль. Також важливо враховувати витрати на підтримку та оновлення такого гібридного рішення.

Розробка повністю кастомного рішення з нуля передбачає використання різних мов програмування (таких як PHP, Python, JavaScript) та фреймворків (Laravel, Django, Express.js). Такий підхід підходить для бізнесів, які потребують максимальної гнучкості та повного контролю над своїм інтернет-магазином. Повна кастомізація дозволяє створити унікальний дизайн та функціонал, що може дати конкурентну перевагу.

Кастомні рішення пропонують можливість масштабування під великі обсяги трафіку та складні бізнес-процеси. Вони забезпечують незалежність від сторонніх платформ, що дозволяє реалізовувати будь-які технічні та бізнес-вимоги [14]. Проте розробка такого рішення потребує значних початкових інвестицій як фінансових, так і часових. Крім того, необхідна команда кваліфікованих розробників для створення, підтримки та оновлення системи.

Розробка інтернет-магазину «з нуля» може бути здійснена за допомогою різних мов програмування та фреймворків, кожен з яких має свої особливості, переваги та недоліки. Основні технології, які часто використовуються для цього, включають PHP з Laravel, Python з Django, JavaScript з MERN (MongoDB, Express.js, React, Node.js) та Ruby з Ruby on Rails.

Laravel, популярний PHP-фреймворк, пропонує багатий набір інструментів

для швидкої та ефективної розробки веб-застосунків. Він відомий своєю легкістю у вивченні, особливо для початківців, завдяки добре структурованій документації. Laravel має велику спільноту розробників, що забезпечує доступ до численних ресурсів і підтримки. Завдяки вбудованим функціям, таким як кешування, сесії та аутентифікація, Laravel забезпечує високу продуктивність [15]. Екосистема Laravel підтримує численні пакети і розширення, такі як Laravel Cashier для обробки платежів.

Однак Laravel має певні недоліки, зокрема складність масштабування під великий трафік порівняно з іншими мовами. PHP зазвичай має меншу продуктивність у порівнянні з мовами, такими як Python або Node.js.

Django, високорівневий веб-фреймворк для Python, відомий своєю швидкістю розробки та масштабованістю. Django дозволяє швидко створювати веб-застосунки завдяки вбудованим адміністративним інтерфейсам і автоматичному генеруванню SQL-запитів. Високий рівень безпеки забезпечується вбудованими механізмами захисту від загроз, таких як SQL-ін'єкції та XSS-атаки [16]. Python і Django легко масштабуються, що робить їх ідеальними для великих проектів.

Основні недоліки Django включають високу вимогу до ресурсів серверу порівняно з мовами на кшталт Node.js та менший вибір хостинг-провайдерів, що підтримують Django порівняно з PHP [17].

Ruby on Rails – це потужний веб-фреймворк для Ruby, відомий своєю елегантністю та простотою у використанні. Rails дозволяє швидко створювати прототипи та фінальні продукти завдяки використанню готових модулів та генераторів. Синтаксис Ruby дозволяє легко читати та підтримувати код. Rails поставляється з багатьма вбудованими інструментами для тестування, управління базами даних та безпеки [18].

Проте Ruby може бути повільнішим порівняно з іншими мовами, такими як JavaScript або Python. Популярність Rails дещо зменшилася останніми роками, що може вплинути на доступність ресурсів і підтримки.

JavaScript з MERN – це стек JavaScript-технологій, який включає

MongoDB, Express.js, React та Node.js. Основна перевага MERN полягає в тому, що він дозволяє використовувати JavaScript на всіх рівнях застосунку (клієнт, сервер, база даних), що спрощує розробку та підтримку [19]. Node.js забезпечує асинхронну обробку запитів, що підвищує продуктивність та швидкість. React забезпечує швидкий рендеринг інтерфейсу користувача [20], а MongoDB дозволяє легко масштабувати базу даних [21]. NPM (Node Package Manager) пропонує величезну кількість пакетів і модулів, що робить MERN гнучким і розширюваним.

Однак MERN потребує більше зусиль для налаштування та підтримки порівняно з іншими стеками [22]. Деякі інструменти та бібліотеки можуть бути менш стабільними порівняно з традиційними рішеннями.

React забезпечує компонентний підхід до розробки інтерфейсу користувача, що спрощує підтримку та розширення застосунку [23]. Компоненти можуть бути повторно використані в різних частинах застосунку, що знижує дублювання коду. NPM (Node Package Manager) надає доступ до величезної кількості пакетів і бібліотек, що дозволяє швидко додавати нові функціональності до застосунку. Це скорочує час розробки та дозволяє зосередитися на бізнес-логіці. Активна спільнота розробників забезпечує доступ до великої кількості навчальних матеріалів, прикладів коду та бібліотек, що значно полегшує вирішення проблем і впровадження нових технологій. Враховуючи всі ці переваги, було прийнято рішення розробляти інтернет-магазин ”Air Baking” засобами MERN, які є потужним інструментом для створення сучасних, продуктивних та масштабованих інтернет-магазинів.

1.2 Технічне завдання на розробку інтернет-магазину "Air Baking"

1.2.1 Область застосування інтернет-магазин "Air Baking"

Інтернет-магазин "Air Baking" призначений для продажу кондитерських та хлібобулочних виробів через Інтернет. Магазин буде доступний для кінцевих

споживачів, які бажають купувати свіжу випічку та інші хлібобулочні вироби з доставкою додому.

1.2.2 Призначення розробки інтернет-магазин "Air Baking"

Метою розробки є створення інтернет-магазину "Air Baking" з використанням стеку MERN (MongoDB, Express.js, React, Node.js). Основні задачі включають надання зручного інтерфейсу для користувачів, автоматизацію процесу замовлення та оплати, забезпечення ефективного управління товарними позиціями та обробку замовлень.

1.2.3 Вимоги до інтернет-магазину "Air Baking"

Інтернет-магазин повинен відповідати функціональним і нефункціональним вимогам.

Функціональні вимоги включають реєстрацію та аутентифікацію користувачів, каталог товарів з категоріями, кошик для покупок, оформлення замовлення, управління запасами та товарами, а також панель адміністратора для управління користувачами, замовленнями та товарами.

Нефункціональні вимоги включають високу продуктивність і швидкодію, масштабованість для підтримки зростаючого числа користувачів, високий рівень безпеки для захисту особистих даних користувачів та фінансових транзакцій, а також сумісність з різними веб-браузерами та мобільними пристроями.

1.2.4 Вимоги до програмної документації інтернет-магазину

Програмна документація повинна включати технічну документацію для розробників, яка містить опис [24]:

– архітектури системи;
– використаних технологій;

– структури бази даних та API;

– інструкції для користувачів з описом основних функцій інтернет-магазину та порядку їх використання;

– адміністративну документацію для адміністраторів системи з описом управління товарами, замовленнями та користувачами.

1.2.5 Техніко-економічні показники для розробки інтернет-магазину "Air Baking"

Розробка інтернет-магазину "Air Baking" має на меті зниження витрат на утримання фізичних магазинів та персоналу через автоматизацію процесів. Передбачається підвищення ефективності за рахунок збільшення швидкості обробки замовлень та підвищення задоволеності клієнтів завдяки зручному інтерфейсу і швидкій доставці. Важливим аспектом є також розширення ринку завдяки доступу до ширшої аудиторії через Інтернет та можливість замовлення товарів з будь-якої точки.

1.2.6 Стадії та етапи розробки інтернет-магазину "Air Baking"

Розробка інтернет-магазину здійснюватиметься у кілька етапів. Спочатку буде проведено аналіз вимог, який включає:

– збір та аналіз вимог замовника;

– визначення основних функцій і нефункціональних вимог.

Далі слідує етап проектування, де розробляється архітектура системи, проектується база даних та API.

На етапі розробки реалізуються серверна частина за допомогою Node.js та Express.js, клієнтська частина за допомогою React, а також інтеграція з базою даних MongoDB [25].

Після цього проводиться тестування, яке включає: модульне, інтеграційне, системне та тестування безпеки та продуктивності.

Впровадження передбачає розгортання інтернет-магазину на сервері, налаштування хостингу та домену.

Останнім етапом є підтримка та обслуговування, що включає виправлення помилок та оновлення програмного забезпечення, а також підтримку користувачів та адміністраторів.

1.2.7 Порядок контролю та прийому інтернет-магазину

Контроль та прийом інтернет-магазину проводитимуться поетапно. Спочатку здійснюється проміжний контроль після завершення кожного етапу розробки для перевірки відповідності виконаних робіт встановленим вимогам. Далі проводиться тестування, яке включає модульне, інтеграційне та системне тестування для виявлення та усунення помилок. Після цього виконується альфа-тестування внутрішньою командою розробників для перевірки працездатності всіх функцій. На наступному етапі проводиться бета-тестування з обмеженою кількістю реальних користувачів для виявлення можливих проблем і отримання зворотного зв'язку [26].

Заключний етап включає прийомочні випробування, які проводить замовник для остаточної перевірки відповідності системи технічним вимогам та вимогам до функціональності. Після успішного проходження всіх випробувань інтернет-магазин "Air Baking" вводиться в експлуатацію.

1.3 Постановка задачі на розробку інтернет-магазину "Air Baking"

1.3.1 Визначення проблеми

Створення інтернет-магазину "Air Baking" необхідно для забезпечення зручного та ефективного способу придбання кондитерських та хлібобулочних виробів, а також десертів онлайн. Відсутність сучасної онлайн-платформи для продажу продукції обмежує доступ споживачів до асортименту товарів, знижує

обсяги продажів та перешкоджає росту бізнесу. Розробка інтернет-магазину дозволить розширити ринок збуту, підвищити задоволеність клієнтів та автоматизувати процеси продажу.

1.3.2 Мета розробки інтернет-магазину "Air Baking"

Основною метою є створення інтернет-магазину, який забезпечить користувачам можливість зручно та швидко замовляти кондитерські вироби онлайн, а адміністраторам - ефективно керувати товарними позиціями, обробкою замовлень та доставкою.

1.3.3 Задачі розробки інтернет-магазину "Air Baking"

Розробка архітектури системи включає:

– визначення структури інтернет-магазину;

– вибір відповідних технологій та інструментів.

Проектування бази даних передбачає створення схеми для зберігання інформації про:

– користувачів;

– товари,

– замовлення.

На етапі реалізації серверної частини використовуються Node.js та Express.js для створення серверної логіки, обробки запитів та управління даними. Клієнтська частина реалізується за допомогою React, що забезпечує зручний та інтуїтивно зрозумілий інтерфейс користувача.

Інтеграція з базою даних забезпечує збереження та отримання даних з MongoDB, а також реалізує необхідні CRUD-операції. Розробка функцій реєстрації та аутентифікації передбачає забезпечення безпечного входу та реєстрації користувачів з використанням сучасних методів захисту даних. Створення каталогу товарів включає перегляд товарів з можливістю групування

по категаріях, а реалізація функції кошика дозволяє користувачам додавати товари у кошик, переглядати його вміст та оформляти замовлення.

Розробка адміністративної панелі створює інтерфейс для адміністраторів для управління товарами, замовленнями та користувачами.

На етапі тестування проводиться модульне, інтеграційне та системне тестування для забезпечення якості та надійності програмного забезпечення. Впровадження передбачає розгортання інтернет-магазину на сервері та забезпечення його безперебійної роботи.

1.3.4 Очікувані результати

Результатом розробки стане повнофункціональний інтернет-магазин "Air Baking", який дозволить користувачам зручно купувати кондитерські вироби онлайн, а адміністраторам ефективно керувати процесами продажу. Інтернет-магазин забезпечить високу продуктивність, безпеку даних та можливість масштабування для підтримки зростаючої кількості користувачів.

1.3.5 Критерії оцінки успішності проекту

Успішність проекту оцінюватиметься за нижче переліченими критеріями. Відповідність функціональним вимогам [27]:

– інтернет-магазин повинен реалізовувати всі заплановані функції, включаючи реєстрацію користувачів, управління товарами, оформлення замовлень онлайн;

– продуктивність системи повинна забезпечувати швидку обробку запитів і замовлень без затримок;

– безпека даних користувачів повинна бути захищена сучасними методами шифрування та автентифікації;

– зручність користування інтерфейсом повинна бути інтуїтивно зрозумілою та зручною для використання на різних пристроях;

– масштабованість системи повинна дозволяти обробляти зростаючу кількість користувачів та замовлень без зниження продуктивності;

– підтримка та обслуговування інтернет-магазину повинні легко піддаватися оновленням та підтримці.

Таким чином розробка інтернет-магазину "Air Baking" засобами MERN дозволить забезпечити ефективний та сучасний спосіб продажу кондитерських виробів онлайн, сприяючи розвитку бізнесу та підвищенню задоволеності клієнтів.

РОЗДІЛ 2. ПРОЕКТНА ЧАСТИНА. РОЗРОБКА ПРОЕКТУ ІНТЕРНЕТ-МАГАЗИНУ "AIR BAKING"

2.1 Обґрунтування вибору архітектури та розробка структури проекту інтернет-магазину "Air Baking"

Перед розробкою інтернет-магазину "Air Baking" важливо провести порівняльний аналіз різних архітектурних підходів, які можна реалізувати засобами стеку MERN (MongoDB, Express.js, React, Node.js). Основні архітектури, що заслуговують на увагу, включають [28]: монолітну архітектуру; мікросервісну архітектуру; модульну архітектуру.

Монолітна архітектура передбачає створення єдиного застосунка, де всі функціональні компоненти інтегровані в один код. Вона має свої переваги: простота розробки та розгортання, оскільки є один кодовий базис для всіх функцій, а також легкість тестування завдяки єдиному проекту [28]. Однак, вона має обмежену масштабованість і складність підтримки з часом, що може ускладнити внесення змін та оновлення.

Мікросервісна архітектура розбиває застосунок на окремі сервіси, кожен з яких відповідає за певну функціональність і може бути розроблений, розгорнутий та масштабований незалежно. Основні переваги включають легкість масштабування окремих сервісів відповідно до навантаження, незалежність розробки та стійкість до помилок, оскільки збої в одному сервісі не впливають на роботу інших. Недоліки включають складність управління інфраструктурою, складніше тестування через розподілену природу застосунка та необхідність належного управління комунікацією між сервісами [29].

Модульна архітектура передбачає розподіл системи на окремі незалежні модулі або застосунки, кожен з яких відповідає за певний функціонал. У випадку інтернет-магазину "Air Baking" модульна архітектура включає три основні компоненти [30]:

– клієнтська частина (фронтенд для користувачів);

– адміністративна панель (фронтенд для адміністраторів);

– серверна частина (бекенд).

Переваги модульної архітектури [30]:

– чіткий розподіл обов'язків, при яком існує можливість працювати над фронтендом і бекендом незалежно, що прискорює розробку та оновлення;

– масштабованість, коли фронтенд та бекенд можуть масштабуватись незалежно один від одного, що дозволяє ефективніше використовувати ресурси;

– зручність тестування та розгортання, адже легше тестувати та розгортати окремі компоненти, що знижує ризик помилок та спрощує підтримку;

– можливість використовувати різні технології для фронтенду та бекенду, що дозволяє вибирати оптимальні інструменти для кожного завдання.

Недоліком модульної архітектури є те, що може виникнути складність в управлінні версіями та залежностями між модулями.

Після порівняння трьох підходів було прийнято рішення, що для інтернет-магазину "Air Baking" найкраще підходить модульна архітектура. Вона забезпечує необхідний рівень масштабованості та гнучкості, дозволяючи окремим командам працювати над клієнтською частиною, адміністративною панеллю та серверною частиною незалежно.

Модульна архітектура дозволяє розподілити систему на чітко визначені компоненти, кожен з яких може бути розроблений, розгорнутий та масштабований незалежно. Це особливо важливо для проекту, який включає різні функції, такі як управління користувачами, каталог товарів, обробка замовлень та оплати. Вибір модульної архітектури для розробки інтернет-магазину "Air Baking" забезпечить високий рівень масштабованості, гнучкості та надійності, що є критичним для успіху проекту.

Модульна архітектура інтернет-магазину "Air Baking" дозволить в подальшому безпосередньо реалізувати мобільний застосунок, який буде взаємодіяти з API серверної частини вже існуючого інтернет-магазину. Тобто модульна архітектура дозволяє створювати гнучкі та масштабовані системи, що легко інтегруються з різними сервісами і платформами. Це особливо актуально

для мобільних застосунків. Такий підхід досліджений при аналізі ролі інноваційних технологій та підходів використання мобільних застосунків у концепції розвитку «розумних» міст [31]. Тези конференції з відповідною публікацією подані в додатку Г кваліфікаційної роботи,

На етапі розробки структури сайту та веб-сторінок визначимося з логічною ієрархією сторінок, їх функціональністю і взаємозв'язками. Проект "Air Baking" буде включати три основні компоненти: клієнтська частина (frontend); адміністративна панель (admin); серверна частина (backend).

Клієнтська частина буде забезпечувати взаємодію з кінцевими користувачами, надаючи зручний інтерфейс для перегляду та купівлі товарів.

Основні сторінки клієнтської частини:

– головна сторінка, як вітрина продуктів;

– перелік товарів у відповідній категорії;

– картки продуктів з детальною інформацією про продукт, включаючи зображення, опис та ціну;

– кошик з переглядом обраних товарів, їх кількістю, загальною вартістю, можливістю видалення товарів;

– оформлення замовлення з формою для введення даних покупця та вибору способу доставки.

– сторінка користувача, тобто особистий кабінет користувача та історія його замовлень;

– сторінка входу/реєстрації, тобто форма для авторизації та реєстрації користувачів.

Адміністративна панель забезпечує можливість управління контентом сайту, замовленнями, користувачами та іншими адміністративними функціями.

Основні сторінки адміністративної панелі:

– головна сторінка адмінпанелі;

– управління товарами, тобто список товарів, можливість додавання, редагування та видалення товарів;

– управління замовленнями, тобто список замовлень, статуси замовлень,

можливість перегляду деталей замовлення та зміни статусу.

Серверна частина відповідає за обробку бізнес-логіки, взаємодію з базою даних і забезпечення безпеки застосунка. Вона також обробляє запити від клієнтської частини і адміністративної панелі, забезпечуючи необхідні API для їх роботи.

Основні компоненти серверної частини:

– файли налаштувань і змінних оточення;

– контролери, тобто логіка обробки запитів, спрямованих до серверної частини;

– middleware, тобто проміжні програми для обробки запитів (авторизація, обробка помилок).

– моделі взаємодії з базою даних, визначення структури даних;

– маршрутизація, тобто визначення маршрутів для обробки запитів;

– завантаження файлів, тобто обробка завантажених файлів, зберігання зображень товарів тощо.

Описана структура подана на рисунку 2.1.

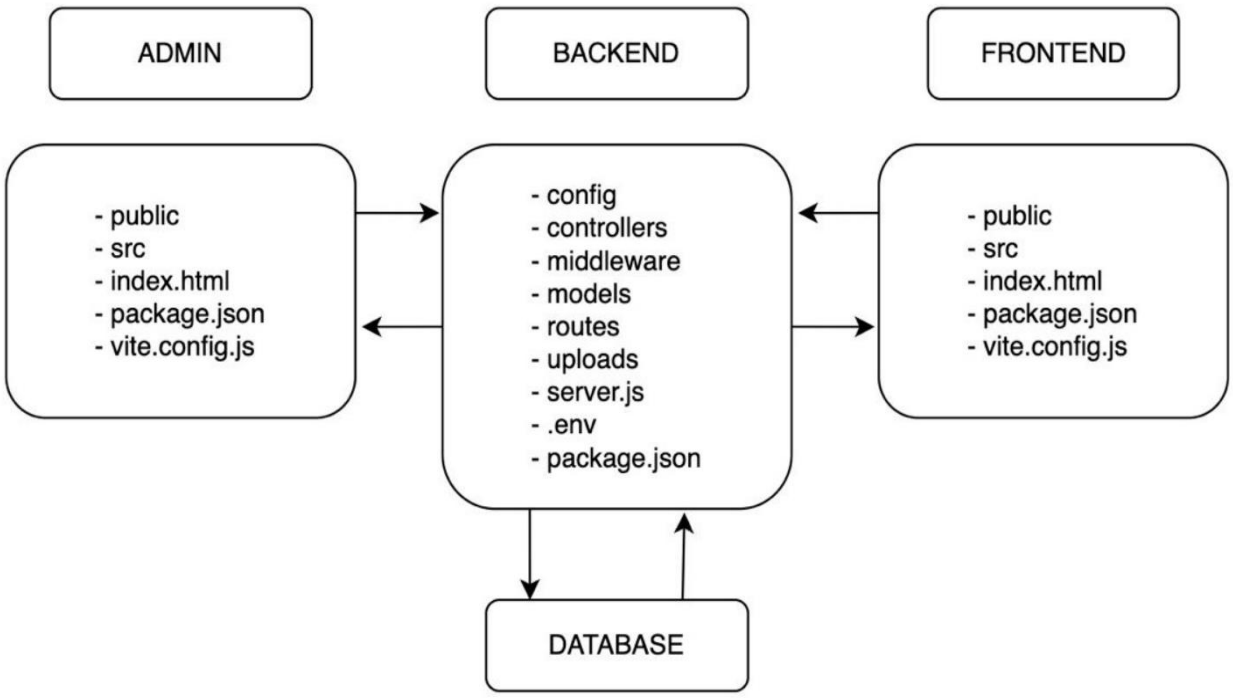


Рисунок 2.1 – Структурна схема проекту Air Baking

Таким чином розробка структури сайту і веб-сторінок є фундаментальним етапом створення інтернет-магазину "Air Baking". Чітка і логічна структура дозволяє забезпечити зручність користування для кінцевих користувачів і адміністраторів, ефективність обробки даних на сервері, а також спрощує подальшу підтримку та масштабування системи.

2.2 Проектування функціональних можливостей веб-сторінок інтернет-магазину "Air Baking"

При проектування функціональних можливостей веб-сторінок інтернет-магазину "Air Baking" важливо визначитися з функціональними можливостями для різних типів користувачів

В кінцевій реалізації проекту інтернет-магазину "Air Baking" ним можуть скористатися:

– неавторизований користувач (гість);
– авторизований користувач (покупець);
– адміністратор інтернет-магазину.

Гість (неавторизований користувач) має обмежений доступ до функціональних можливостей сайту. Він може переглядати товари, включаючи зображення, опис та ціну, переглядати товари за категоріями. Гості також мають можливість зареєструватися на сайті або увійти до існуючого облікового запису.

Покупець (авторизований користувач) має розширені функціональні можливості порівняно з гостями. Він може додавати товари до кошика і переглядати його вміст, оформлювати замовлення, заповнюючи форму з інформацією про доставку, переглядати історію замовлень і їх статуси.

Адміністратор має доступ до функцій управління сайтом. Він може додавати, редагувати та видаляти товари, переглядати замовлення, змінювати їх статуси, керувати статусом замовлень.

На рисунку 2.2 подана діаграма використання інтернет-магазину "Air Baking" різними користувачами.

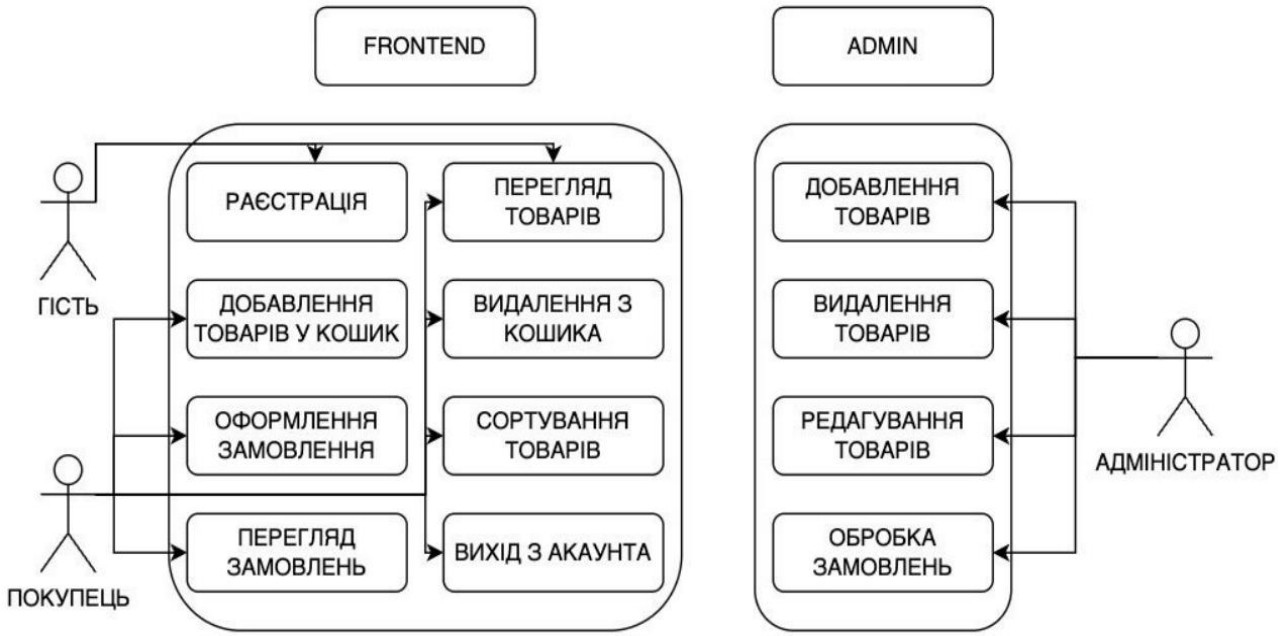


Рисунок 2.2 - Діаграма використання інтернет-магазину "Air Baking" різними користувачами

В результаті аналізу діаграма використання різними користувачами було прийнято рішення реалізувати такі компоненти користувацького інтерфейсу інтернет-магазину "Air Baking":

– Головну сторінку, яка містить вітрину товарів та навігацію по категоріях. Вона має адаптивну верстку для зручного перегляду на різних пристроях.

– Картку товарів, яка включає детальну інформацію про товар, зображення, опис ціну та кнопки для додавання до кошика і переходу до оформлення замовлення.

– Кошик, який містить перелік товарів у кошику, кількість, ціну, можливість змінювати кількість товарів, видаляти товари, і кнопку для переходу до оформлення замовлення.

– Сторінку оформлення замовлення, яка включає форму для введення даних покупця, вибір способу доставки та оплати, огляд замовлення перед підтвердженням.

– Сторінку користувача (особистий кабінет), який містить історію його

замовлень та можливість перегляду їх статусів.

– Сторінку входу/реєстрації містить форму для авторизації та реєстрації нових користувачів.

– Адміністративну панель, яка містить дашборд для управління товарами та замовленнями користувачів.

Таким чином інтерфейс користувача інтернет-магазину "Air Baking" буде реалізований за допомогою компонентів:

– виведення товарів;

– реєстрація;

– авторизація;

– кошик;

– про нас;

– контакти.

Структурна схема інтерфейсу користувача інтернет-магазину "Air Baking" подана на рисунку 2.3.

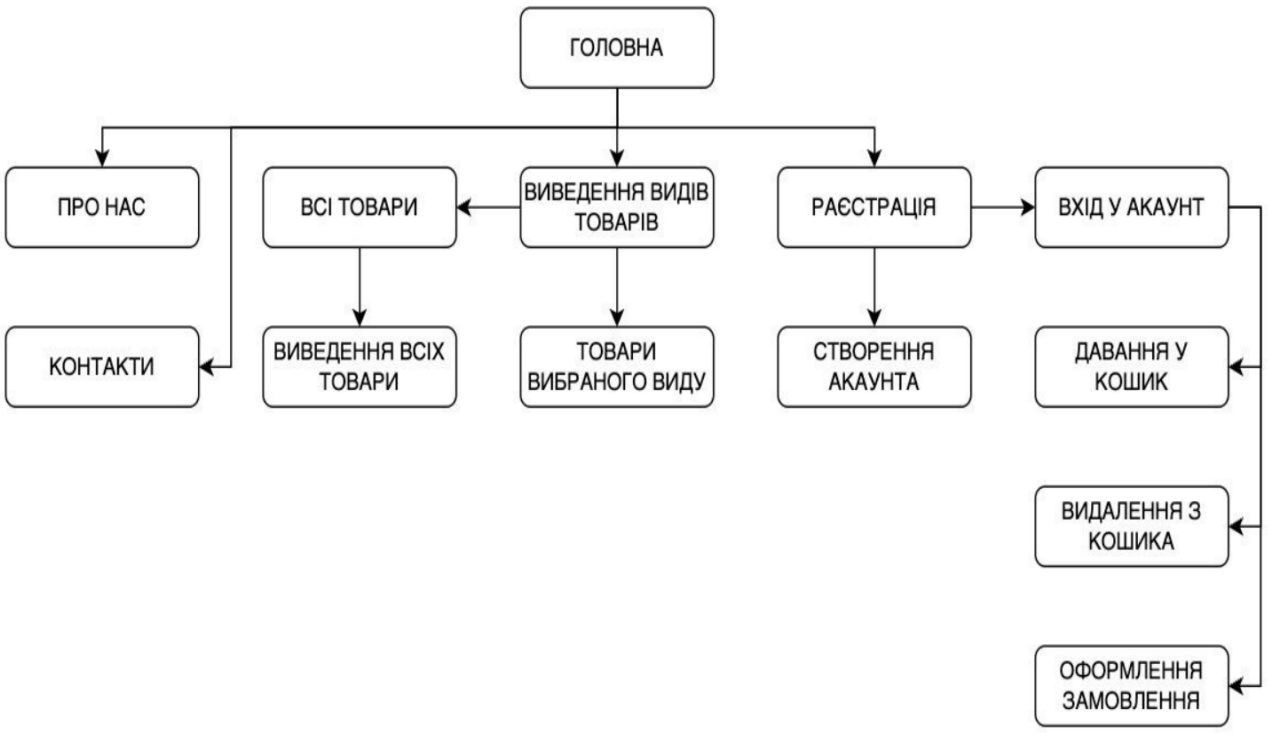


Рисунок 2.3 – Структурна схема інтерфейс користувача інтернет-магазину "Air Baking"

Інтерфейс адміністративної частини інтернет-магазину "Air Baking" буде визначений функціоналом для:

– створення товарів;
– редагування властивостей товарів;
– видалення товарів;
– обробка замовлень покупців.

Структурна схема інтерфейсу адміністративної частини інтернет-магазину "Air Baking" подана на рисунку 2.4.

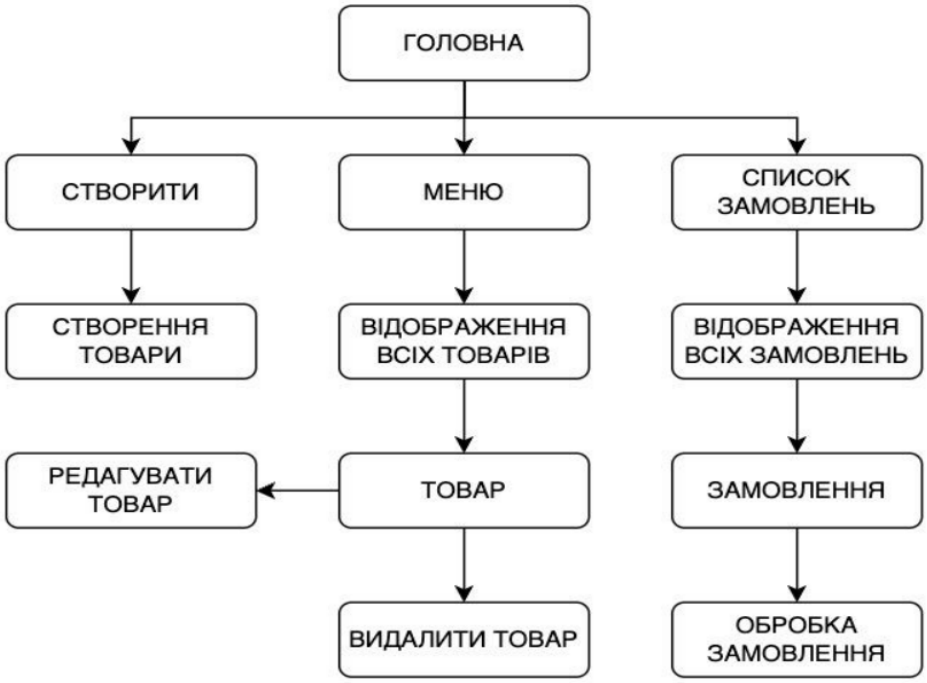


Рисунок 2.4 – Структурна схема інтерфейсу адміністративної частини інтернет-магазину "Air Baking"

Проектування функціональних можливостей сторінок для інтернет-магазину "Air Baking" включає розробку різноманітних сторінок з різними функціональними можливостями для гостей, авторизованих користувачів та адміністраторів. Це забезпечує зручність користування, ефективність управління контентом та замовленнями, а також підтримує високий рівень користувацького досвіду.

2.3 Проектування та реалізація структури бази даних інтернет-магазину "Air Baking"

В якості бази даних інтернет-магазину "Air Baking" було обрано MongoDB. Такий підхід при реалізації інтернет магазину має суттєві переваги з огляду на гнучкість, масштабованість та зручність роботи з даними про товари та користувачів. Ключові аргументи на користь MongoDB [32]:

– гнучкість структури даних, адже MongoDB використовує нереляційну модель даних, що робить її ідеальною для зберігання різноманітних даних про продукти, включаючи описи, зображення та характеристики;

– можливість легко додавати нові поля або змінювати структуру даних без шкоди для існуючих даних, що робить її гнучкою для адаптації до мінливих вимог магазину.

Це особливо корисно для динамічних продуктів, таких як одяг, де характеристики та описи можуть часто змінюватися.

Також MongoDB має можливість горизонтально масштабуватися. Це означає, що існує можливість легко додавати більше серверів в умовах обробки зростаючого навантаження та обсягу даних. Це робить її ідеальною для інтернет-магазинів, які очікують значного зростання трафіку та продажів.

MERN stack також добре підходить для горизонтального масштабування, що робить його чудовим вибором для такого рішення [33].

MongoDB забезпечує зручність роботи з даними, адже використовує JSON-подібний формат документів для зберігання даних, що робить його простим для розуміння та використання розробниками. Це полегшує інтеграцію з фронт-ендом MERN, адже JSON є загальноприйнятим форматом обміну даними.

Також MongoDB пропонує потужні функції запитів, які дозволяють легко отримувати доступ до потрібних даних про продукти, замовлення та користувачів.

Додатковими перевагами MongoDB є те, що її використання забезпечує

високу продуктивність та швидкість читання/запису [34], які є важливими критеріями для забезпечення швидкої продуктивні інтерфейсу користувачів магазину інтернет-магазину "Air Baking". Також MongoDB має вбудовану підтримку розподілених систем, що робить її надійною та стійкою до відмов.

Структура бази даних інтернет-магазину "Air Baking" буде включати три колекції:

– users;

– foods;

– orders;

Кожна з цих колекцій матиме свою специфічну структуру, яка визначається схемами userSchema, foodSchema та orderSchema. Розглянемо детально структури кожної з колекцій:

Структура колекції users:

– name – рядок, обов'язкове поле, яке зберігає ім'я користувача;

– email – рядок, обов'язкове унікальне поле, яке зберігає електронну адресу користувача;

– password – рядок, обов'язкове поле, яке зберігає хешований пароль користувача;

– cartData - об'єкт, який зберігає дані кошика користувача;

– type – визначає роль користувача.

Структура колекція foods:

– name – рядок, обов'язкове поле, яке зберігає назву страви;

– description – рядок, обов'язкове поле, яке зберігає опис страви;

– price – рядок, обов'язкове поле, яке зберігає ціну страви;

– image – рядок, обов'язкове поле, яке зберігає URL зображення страви;

– category – рядок, обов'язкове поле, яке зберігає категорію страви.

Структура колекції orders:

– userId – рядок, обов'язкове поле, яке зберігає ідентифікатор користувача, який зробив замовлення;

– items - масив, обов'язкове поле, яке зберігає список замовлених товарів;

– amount - число, обов'язкове поле, яке зберігає загальну суму замовлення;

– address - об'єкт, обов'язкове поле, яке зберігає адресу доставки;

– status – рядок, за замовчуванням "Обробка замовлення", яке зберігає статус замовлення;

– date – дата, за замовчуванням поточна дата, яке зберігає дату створення замовлення;

– payment - булеве значення, за замовчуванням false, яке зберігає статус оплати.

Взаємозв'язки між колекціями:

– користувачі (users) можуть мати кілька замовлень (orders), що реалізується через поле userId в колекції orders;

– кожне замовлення (orders) містить інформацію про товари (foods), що зберігається в масиві items.

Структура бази даних подана на рисунку 2.5.

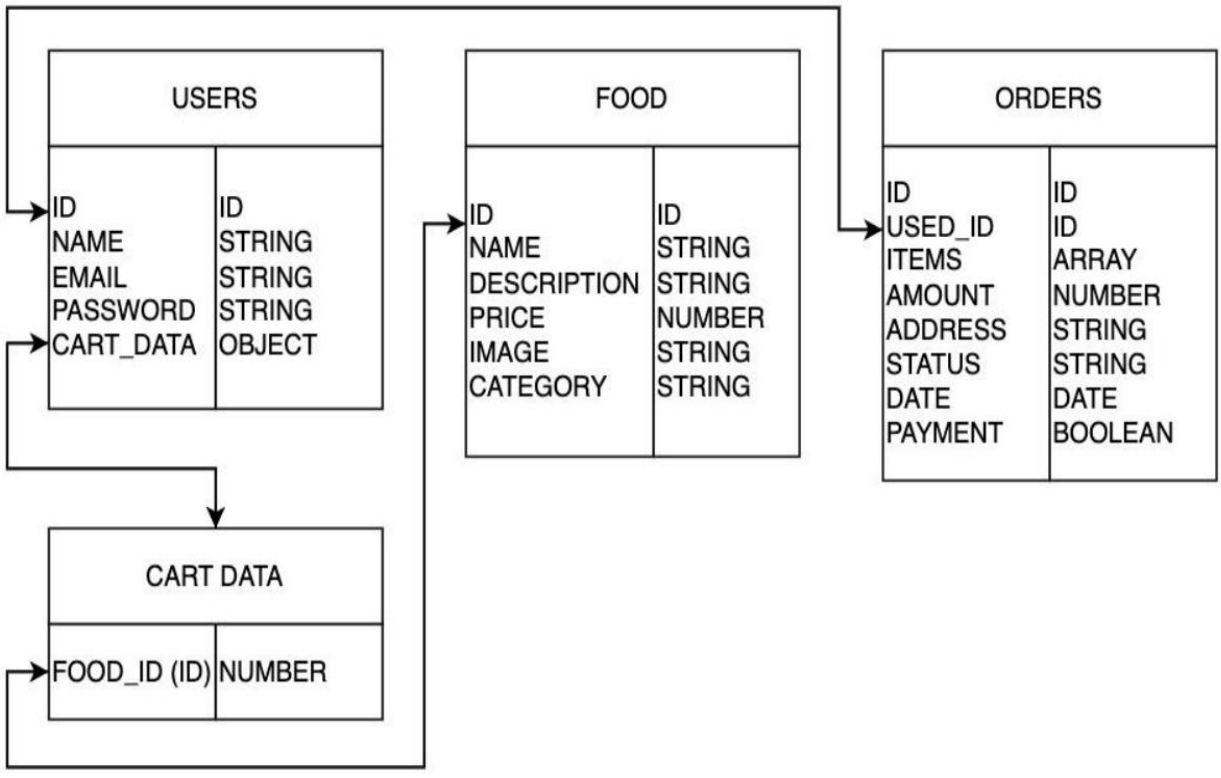


Рисунок 2.5 – Структура бази даних інтернет-магазину "Air Baking"

Лістинг коду для оголошення схеми та моделі кондитерського виробу реалізований у файлі backend\models\foodModel.js та поданий в додатку А кваліфікаційної роботи. Схема та модель для користувачів та їх замовлень реалізовані аналогічно до схеми кондитерських виробів та визначені у файлах проекту:

– backend\models\orderModel.js;
– backend\models\userModel.js.

Розроблені схеми та колекції забезпечують базову структуру для управління користувачами, товарами та замовленнями в системі, дозволяючи реалізувати основні функції, необхідні для інтернет-магазину "Air Baking"

2.4 Програмування інтернет-магазину "Air Baking"

Процес створення інтернет-магазину "Air Baking" з використанням стеку технологій MERN (включає кілька етапів, починаючи від налаштування середовища розробки до впровадження функціональних можливостей і тестування.

Для налаштування середовища розробки потрібно встановити Node.js та npm. Для цього необхідно завантажити та встановити останню версію Node.js з офіційного сайту (https://nodejs.org/), яка також включає npm (Node Package Manager).

Для ініціалізації проекту потрібно створити основну папку проекту, наприклад, "airbaking".

Оскільки серверна частина інтернет-магазину "Air Baking", клієнтська частина та панель адміністратора будуть реалізовані як окремі підпроекти, то у папці «airbaking» які створити три папки:

– backend – для реалізації бекенду інтернет магазину;
– frontend – для реалізації клієнтської частини;
– admin – для реалізації панелі адміністратора.

Початкова структура папок проекту подана на рисунку 2.6.

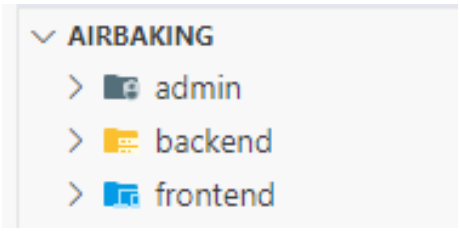


Рисунок 2.6 – Початкова структура папок проекту інтернет-магазину "Air Baking"

Подальша реалізація цілісного інтернет-магазину "Air Baking" буде реалізована у якості трьох окремих проектів.

2.4.1 Реалізація серверної частини інтернет-магазину "Air Baking"

Серверна частину інтернет-магазину "Air Baking" реалізована на основі фреймворку Express.js. Це мінімалістичний і гнучкий веб-фреймворк для Node.js, який забезпечує потужний набір функцій для створення веб- та мобільних застосунків [35]. Він дозволяє легко налаштовувати маршрути, обробляти запити та відповіді HTTP, а також інтегрувати різні середовища виконання та бази даних. Express.js є одним з найпопулярніших фреймворків для розробки серверної частини веб-застосунків завдяки своїй простоті використання та розширюваності.

Для реалізації серверної частини інтернет-магазину "Air Baking" з використанням Express.js необхідно спершу ініціалізавати проект та встановити залежності. Для цього потрібно:

– створити директорію backend для серверної частини;

– ініціалізувати проект командою npm init –y;

– встановити необхідні пакети: Express, Mongoose (для роботи з MongoDB), dotenv (для керування змінними середовища), та інші залежності за допомогою команди: npm install express mongoose dotenv cors та перелічити інші

пакети, необхідні для реалізації проекту.

В результаті цього було створено конфігураційний файл package.json, у якому прописані залежності для пакетів, які використані при реалізації бекенду. При реалізації бекенд-частини проекту інтернет-магазину "Air Baking" було використано низку важливих пакетів, кожен з яких відіграє свою роль у забезпеченні функціональності та безпеки застосунку. Розглянемо короткий опис кожного пакета та його призначення [36]:

– bcrypt - використовується для хешування паролів користувачів перед їх збереженням у базі даних. Це забезпечує безпеку паролів, оскільки навіть у разі витоку даних паролі будуть захищені.

– body-parser - використовується для парсингу тіла запитів HTTP, що дозволяє зручно обробляти дані, надіслані у форматі JSON, URL-encoded або інші.

– cors - дозволяє налаштовувати CORS (Cross-Origin Resource Sharing) політику для сервера, що визначає, які домени можуть взаємодіяти з сервером.

– dotenv - використовується для зберігання конфіденційних даних та конфігураційних змінних у файлі .env. Це дозволяє зручно керувати налаштуваннями середовища виконання.

– express - основний веб-фреймворк, який використовується для створення серверної частини застосунку. Express.js забезпечує маршрутизацію, обробку запитів та інші основні функції веб-сервера.

– jsonwebtoken - використовується для створення та перевірки JSON Web Tokens (JWT), які часто застосовуються для аутентифікації та авторизації користувачів.

– mongoose - об'єктно-документний моделюючий інструмент для MongoDB, який дозволяє зручно працювати з базою даних, використовуючи схеми та моделі.

– multer - використовується для обробки завантажень файлів у застосунку. Це дозволяє зручно зберігати файли, такі як зображення продуктів.

– nodemon - інструмент для автоматичного перезапуску Node.js сервера

при внесенні змін до коду. Зручний для розробки, оскільки зменшує час на ручний перезапуск сервера.

– validator - пакет для валідації даних, таких як електронна пошта, URL-адреси, та інші введені користувачем дані.

Кожен з цих пакетів виконує важливу роль у створенні безпечного, функціонального та зручного у використанні інтернет-магазину "Air Baking". Вони забезпечують різні аспекти розробки, включаючи безпеку, обробку даних, з'єднання з базою даних, маршрутизацію, обробку платежів та інше. Використання цих пакетів дозволяє ефективно реалізувати необхідні функціональні можливості та забезпечити надійність застосунку.

Далі було створено конфігураційний файл для підключення до бази даних. Для цього у директорії config реалізовано файл db.js для налаштування з'єднання з MongoDB.

На наступному етапі створено основний файлу сервера. Для цього у кореневій директорії backend реалізовано файл server.js. Файл server.js є центральним елементом серверної частини проекту "Air Baking". Його основні функції включають налаштування та запуск веб-сервера, підключення до бази даних, визначення маршрутизації та налаштування проміжного програмного забезпечення (middleware).

В даному файлі (server.js) створено екземпляр сервера Express, який слухає на певному порту і обробляє вхідні HTTP-запити. У даному випадку, сервер слухає на порту, який визначений у змінних середовища або на порту 4000.

Express.js використовує middleware для обробки запитів перед їх передачею до кінцевих точок (endpoints). У файлі server.js налаштовуються такі middleware, як express.json() для парсингу JSON у запитах і cors() для налаштування політики CORS.

Файл server.js підключає сервер до бази даних MongoDB за допомогою функції connectDB(), яка визначена в окремому файлі конфігурації (db.js).

В кінці файлу server.js запускається сервер і виводиться повідомлення в консоль, яке підтверджує успішний запуск і вказує на порт, на якому працює

сервер. Таким чином файл server.js виконує критично важливі функції для роботи серверної частини проекту "Air Baking". Він налаштовує та запускає сервер, встановлює з'єднання з базою даних, визначає маршрути API, налаштовує проміжне програмне забезпечення і забезпечує основну точку доступу для перевірки роботи API. Завдяки цьому файлу забезпечується ефективне та організоване управління серверною логікою застосунку. Лістинг коду файлу server.js поданий в додатку Б кваліфікаційної роботи.

Також у файлі server.js визначаються маршрути для різних ресурсів, таких як користувачі, продукти, корзини та замовлення. Це забезпечує чітку організацію маршрутизації і зручне керування API-запитами.

Для реалізації API бекенду інтернет-магазину "Air Baking" у файлі server.js реалізовано проміжні обробники, які будуть виконувати роль маршрутизатора. Для цього створено директорію routes у яку додано файли userRoute.js, foodRoute.js, cartRoute.js, orderRoute.js для обробки відповідних API-запитів. У файлі server.js визначено, що:

– маршрут для користувачів /api/user буде використовувати файл маршрутизатора userRouter.js для обробки запитів, пов'язаних з користувачами;

– маршрут для продуктів /api/food буде використовувати файл маршрутизатора foodRouter.js для обробки запитів, пов'язаних з продуктами;

– маршрут для зображень /images використовує Express для надання статичного контенту із директорії uploads;

– маршрут для корзин /api/cart буде використовувати файл маршрутизатора cartRouter.js для обробки запитів, пов'язаних з корзинами;

– маршрут для замовлень /api/order буде використовувати файл маршрутизатора orderRouter.js для обробки запитів, пов'язаних з замовленнями.

У таблицях 2.1, 2.2, 2.3 та 2.4 подано опис ендпойнтів бекенду з наступними полями таблиць: HTTP-метод (для визначення методу запиту); маршрут (власне ендпойнт); призначення – дія, пов’язана з маршрутом.

В табл. 2.1 подано опис ендпойнтів, визначених у файлі foodRoute.js

Таблиця 2.1 – Параметри ендпойнтів, визначений у файлі userRoute.js

HTTP-метод	**Маршрут**	**Призначення**
POST	/api/user/register	реєстрація нового користувача через функцію registerUser
POST	/api/user/login	вхід користувача через функцію loginUser
POST	/api/user/admin	вхід адміністратора через функцію loginAdmin

В табл. 2.2 подано опис ендпойнтів бекенду, визначених у файлі foodRoute.js.

Таблиця 2.2 – Параметри ендпойнтів, визначений у файлі foodRoute.js

HTTP-метод	**Маршрут**	**Призначення**
GET	/api/food/list	отримання списку всіх продуктів ерез функцію listFood
GET	/api/food/:id	отримання інформації про продукт за його ID через функцію getFoodById
POST	/api/food/add	додавання нового продукту з зображенням через функцію addFood, використовуючи middleware upload.single('image')
POST	/api/food/remove	видалення продукту через функцію removeFood
PUT	/api/food/edit/:id	редагування продукту за його ID з можливістю завантаження нового зображення через функцію editFoodById, використовуючи middleware upload.single('image')

В табл. 2.3 подано опис ендпойнтів бекенду, визначених у файлі orderRoute.js.

Таблиця 2.3 – Параметри ендпойнтів, визначений у файлі orderRoute.js

HTTP-метод	**Маршрут**	**Призначення**
GET	/api/order/list	отримання списку всіх замовлень через функцію listOrders
POST	/api/order/userorders	отримання замовлень окремого користувача через функцію userOrders, використовуючи middleware authMiddleware
POST	/api/order/place	оформлення нового замовлення через функцію placeOrder, використовуючи middleware authMiddleware
POST	/api/order/status	оновлення статусу замовлення через функцію updateStatus
POST	/api/order/verify	верифікація замовлення через функцію verifyOrder
POST	/api/order/placecod	оформлення замовлення з оплатою при доставці через функцію placeOrderCod

В табл. 2.4 подано опис ендпойнтів бекенду, визначених у файлі cartRoute.js.

Таблиця 2.4 – Параметри ендпойнтів, визначений у файлі cartRoute.js

HTTP-метод	**Маршрут**	**Призначення**
1	**2**	**3**
POST	/api/cart/get	отримання корзини користувача через функцію getCart, використовуючи middleware authMiddleware

Продовження таблиці 2.4

1	2	3
POST	/api/cart/add	додавання продукту до корзини ерез функцію addToCart, використовуючи middleware authMiddleware
POST	/api/cart/remove	видалення продукту з корзини через функцію removeFromCart, використовуючи middleware authMiddleware.

Ці ендпойнти забезпечують основну функціональність для користувачів, продуктів, корзин та замовлень у проекті "Air Baking". Вони дозволяють виконувати всі необхідні операції, такі як реєстрація та вхід користувачів, управління продуктами, обробка замовлень та управління корзиною покупок.

Файл server.js також визначає основну точку доступу, яка відповідає на GET-запити до кореневого маршруту ("/") простим повідомленням, що підтверджує роботу API.

Після створення маршрутизаторів для обробки API-запитів було створено директорію models в якій реалізовано файли для моделей даних, які використовуються для взаємодії з MongoDB. Структура моделей описана у підрозділі 2.3 кваліфікаційної роботи.

На наступному етапі було створено файл .env у кореневій директорії backend для зберігання конфігураційних змінних (змінних оточення).

Далі було створено директорію controllers, у якій реалізовано файли контролерів, які будуть містити логіку обробки запитів, функції яких будуть співставлятися маршрутизатором для обробки ендпойнтів.

На заключному етапі реалізації бекенду було запущено сервер командою node server.js та перевірено роботу API за допомогою інструменту Postman.

Отже реалізація серверної частини інтернет-магазину "Air Baking" з

використанням Express.js включає кілька ключових етапів: налаштування середовища розробки, створення конфігураційного файлу для підключення до бази даних, налаштування основного файлу сервера, створення маршрутизаторів, моделей і контролерів, а також налаштування змінних середовища. Кожен з цих кроків є важливим для забезпечення надійної та ефективної роботи серверної частини застосунку.

2.4.2 Реалізація клієнтської частини інтернет-магазину "Air Baking"

Клієнтська частина інтернет-магазину Air Baking розробляється з використанням React. Щоб забезпечити високу продуктивність та швидкий рендеринг, для створення застосунку використовується Vite — сучасний інструмент для збору JavaScript-застосунків. Vite забезпечує швидке завантаження та гаряче оновлення модулів, що значно спрощує розробку.

Для створення застосунку на React спочатку необхідно встановити Vite та створити новий проект. Це можна зробити за допомогою наступних команд:

npm create vite@latest airbaking --template react

cd airbaking

npm install

Ці команди створять новий проект у директорії airbaking з шаблоном React та встановлять всі необхідні залежності.

Після створення проекту, структура папок виглядатиме наступним чином:

– src/ — основна папка для вихідного коду застосунку;

– public/ — статичні файли, які будуть сервісуватися як є;

– index.html — головний HTML-файл, який буде використовуватися для завантаження застосунку

Для управління маршрутизацією у застосунку використовується бібліотека react-router-dom. Для її встановлення необхідно в терміналі використати команду:

```
npm install react-router-dom
```

Файл src/main.jsx є точкою входу для React-застосунку, створеного за допомогою Vite. Його основне завдання полягає в ініціалізації застосунку та підключенні головного компонента до HTML-документа. У цьому файлі імпортуються всі необхідні бібліотеки, включаючи:

– React;
– ReactDOM;
– основний компонент застосунку (App.jsx);
– маршрутизатор (BrowserRouter);
– провайдер контексту (StoreContextProvider);
– глобальні стилі (index.css).

Використовуючи метод ReactDOM.createRoot, створюється кореневий елемент, в який буде вмонтований застосунок, що вказує React, де саме в DOM має бути розміщений застосунок.

Маршрутизація забезпечується включенням BrowserRouter з react-router-dom, що дозволяє використовувати маршрути та навігацію між різними сторінками застосунку. Обгортання застосунку у StoreContextProvider надає доступ до глобального контексту стану для всіх компонентів застосунку, спрощуючи управління станом і передачу даних між компонентами. Приклад коду для створення кореневого елементу та рендеригу застосунку поданий у лістингу 2.1.

Лістинг 2.1 - Код для створення кореневого елементу та рендерингу застосунку у файлі src/main.jsx

```
ReactDOM.createRoot(document.getElementById('root')).render(
  <BrowserRouter> {/* Налаштування маршрутизації */}
    <StoreContextProvider> {/* Обгортання застосунку у провайдер контексту */}
      <App /> {/* Рендеринг головного компонента застосунку */}
    </StoreContextProvider>
  </BrowserRouter>,
)
```

В основному виклику рендерингу компонент App обгортається у

BrowserRouter та StoreContextProvider, що забезпечує доступ до функціоналу маршрутизації та контексту для всіх дочірніх компонентів. Таким чином, файл src/main.jsx є критично важливим для налаштування та запуску React-застосунку, забезпечуючи початкову конфігурацію, підключення основних функціональних компонентів та інтеграцію з DOM.

Вся клієнтська частина інтернет-магазину ”Air Baking” складається з компонентів, які відповідають за різні частини інтерфейсу. Компонент App є основним компонентом React-застосунку, який відповідає за загальну структуру та організацію застосунку. Він забезпечує маршрутизацію, використовуючи бібліотеку react-router-dom, що дозволяє визначати, які компоненти мають відображатися для різних URL-адрес. Програмний код маршрутизатора клієнтської наведено у лістингу 2.2.

Лістинг 2.2 – Програмний код маршрутизатора клієнтської частини

```
<Routes> {/* Налаштування маршрутів */}
     <Route path='/' element={<Home />} /> {/* Маршрут для головної сторінки */}
     <Route path='/cart' element={<Cart />} /> {/* Маршрут для сторінки корзини */}
     <Route path='/order' element={<PlaceOrder />} /> {/* Маршрут для сторінки оформлення замовлення */}
     <Route path='/myorders' element={<MyOrders />} /> {/* Маршрут для сторінки моїх замовлень */}
     <Route path='/verify' element={<Verify />} /> {/* Маршрут для сторінки верифікації */}
</Routes>
```

Повний лістинг коду файлу frontend\src\App.jsx поданий в додатку В кваліфікаційної роботи.

У компоненті App також включена навігаційна панель (Navbar), яка дозволяє користувачам переміщуватися між різними сторінками застосунку.

Компонент App використовує стан для управління відображенням вікна входу (LoginPopup). Використання хука useState дозволяє показувати або приховувати вікно входу залежно від стану.

Компонент ToastContainer з бібліотеки react-toastify додається для відображення повідомлень, що дозволяє інформувати користувачів про різні події, такі як успішне виконання дії або помилка.

Компонент App визначає основну структуру сторінок застосунку, включаючи:

– заголовок;
– навігаційну панель;
– основний контент
– футер.

Для кожного маршруту визначається відповідний компонент, що відображається на сторінці. App інкапсулює всі основні компоненти застосунку, забезпечуючи централізоване місце для управління ними, що дозволяє легше підтримувати та розширювати застосунок.

Загалом, компонент App виконує ключову роль у забезпеченні чіткої структури застосунку, організації навігації, управлінні станом та інтеграції функціональних компонентів, що робить його важливим елементом у створенні інтерактивного та зручного веб-застосунку.

Кожна сторінка застосунку реалізується як окремий компонент у папці src/pages. Наприклад, Home.jsx містить список продуктів, а Product.jsx — деталі конкретного продукту.

Для стилізації компонентів використовується CSS або CSS-in-JS бібліотеки, такі як styled-components. Приклад використання styled-components:

Управління станом в клієнттській частині проекту "Air Baking" здійснюється за допомогою контексту React (React Context API). Це підхід, що дозволяє передавати дані через компонентне дерево без необхідності явно передавати пропси на кожному рівні. Контекст дозволяє легше керувати глобальним станом застосунку, роблячи його доступним для будь-якого компонента в застосунку. Для цього у файлі src/main.jsx застосунок обгорнуто в провайдер контексту StoreContextProvider, що забезпечує доступ до глобального стану для всіх компонентів. Компонент StoreContextProvider визначає

глобальний стан і методи для його оновлення. Він надає доступ до цього стану та методів для всіх компонентів, які використовують контекст. Компоненти можуть використовувати контекст для доступу до глобального стану та методів для його оновлення. Наприклад, компонент Navbar може отримати доступ до стану користувача та методу для його оновлення.

Таким чином, процес розробки клієнтської частини інтернет-магазину Air Baking передбачає використання Vite для швидкого старту, налаштування роутінгу з допомогою react-router-dom, створення компонентів та сторінок, застосування стилізації з допомогою styled-components та управління станом застосунку з допомогою context API. Такий підхід забезпечує ефективну розробку, підтримку та розширення функціональності інтернет-магазину.

2.5 Тестування інтернет-магазину "Air Baking"

Тестування інтернет-магазину "Air Baking" є важливим етапом розробки, який дозволяє переконатися в правильності роботи всіх функцій та компонентів системи. У проекті "Air Baking" тестування включає функціональне тестування як бекенду, так і фронтенду, що забезпечує належну роботу всіх частин застосунку.

Функціональне тестування бекенду включає перевірку API, забезпечуючи правильність їх виконання відповідно до вимог. Для тестування API використовувався інструмент Postman, який дозволяє виконувати HTTP-запити і перевіряти відповіді сервера.

На етапі тестування користувацьких маршрутів перевірялася реєстрація та вхід користувачів. Маршрут POST /api/user/register тестується на створення нового користувача, а POST /api/user/login перевіряється на успішний вхід з правильними даними та відмову з некоректними. Приклад результату роботи Postman при успішній авторизації користувача поданий на рисунку 2.7. Приклад відповіді сервера при вводі некоректному вводі електронної пошти поданий на рисунку 2.8.

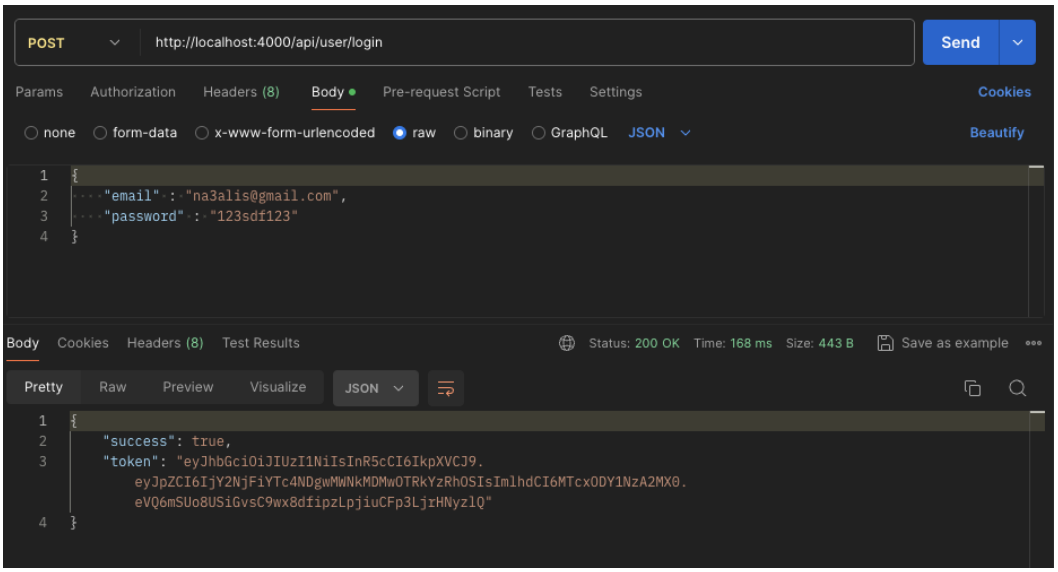


Рисунок 2.7 – Відповідь сервера при успішній авторизації

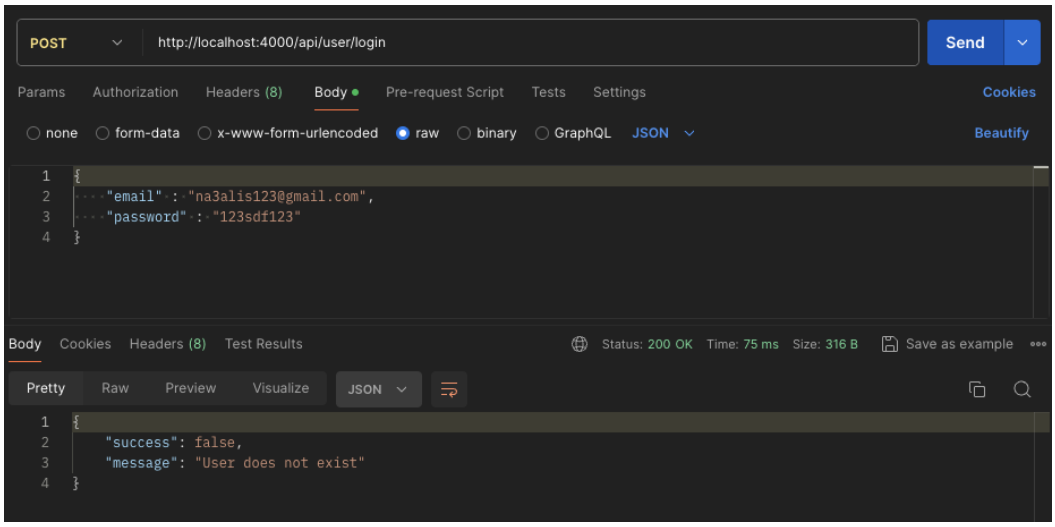


Рисунок 2.8 – Відповідь сервера при некоректному вводі електронної пошти

Тестування маршрутів продуктів передбачало перевірку отримання списку продуктів, додавання нового продукту, редагування та видалення. Маршрут GET /api/food/list тестувався на повернення списку продуктів, а POST /api/food/add на додавання нового продукту до бази даних.

Маршрути корзини тестувалися на додавання та видалення товарів. Маршрут POST /api/cart/add перевірялися на додавання товару до корзини, а POST /api/cart/remove на видалення товару.

На етапі тестування маршрутів замовлень перевірялося створення нового замовлення, отримання списку замовлень та оновлення статусу. Маршрут POST

/api/order/place тестувався на створення нового замовлення, а POST /api/order/status на оновлення його статусу.

Таким чином для тестування серверної частини інтернет-магазину у інструменті Postman був створений перелік запитів, поданий на рисунку 2.9, для перевірки коректності роботи API серверної частини, тобто усіх маршрутів, описаних в таблицях 2.1 – 2.4.

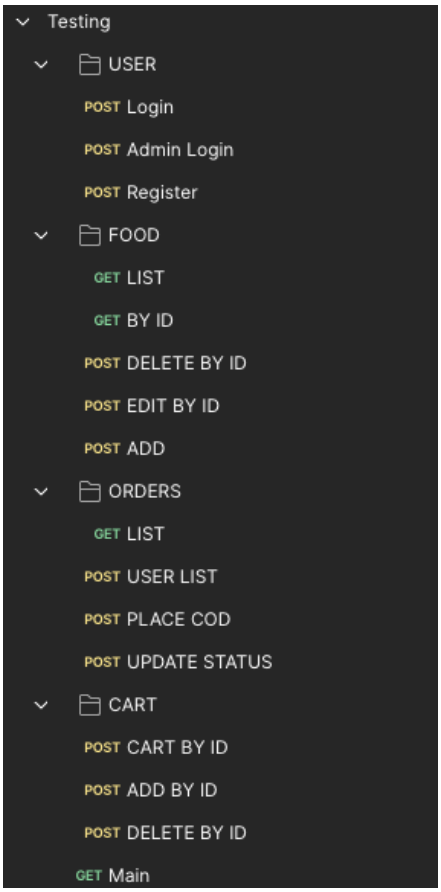


Рисунок 2.9 – Перелік запитів у інструменті Postman для тестування серверної частини інтернет-магазину "Air Baking"

Для функціонального тестування фронтенду перевірялася правильність роботи користувацького інтерфейсу та коректність інтеграції з бекендом.

На етапі тестування компонентів кожен компонент застосунку перевіряється окремо для перевірки коректності відображення та роботи. Наприклад, компонент Navbar був протестований на коректність відображення різних його елементів залежно від стану аутентифікації користувача. На рисунку

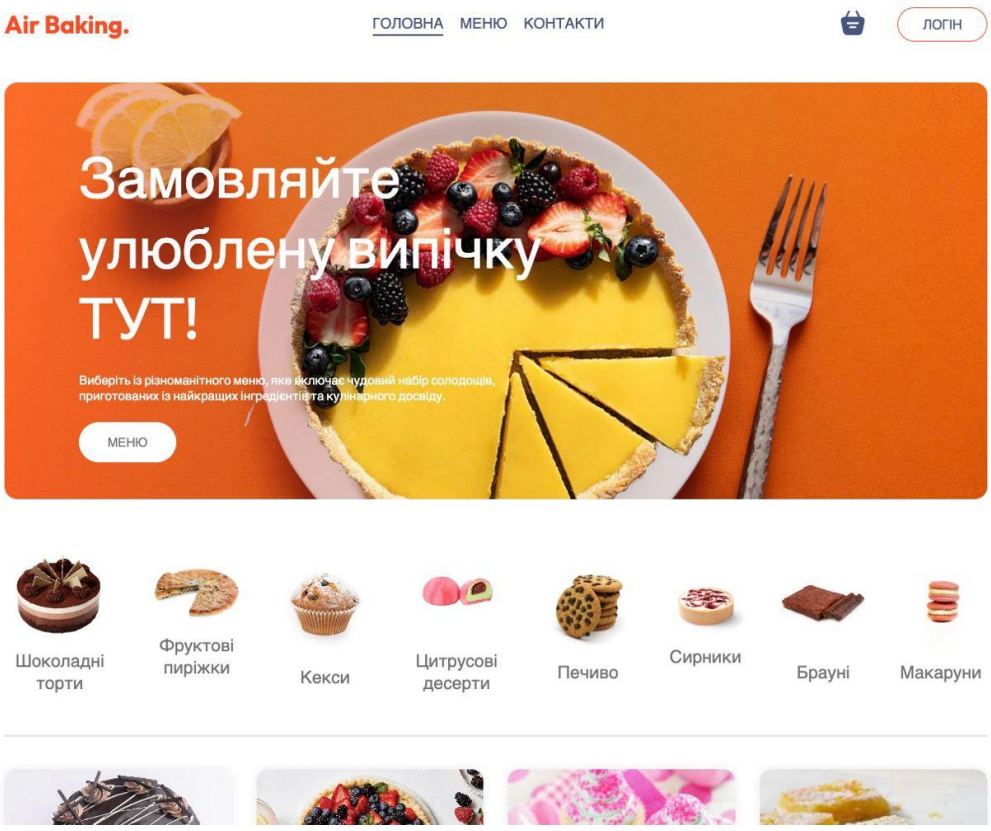
2.10 наведено вигляд навігаційної панелі для неавторизованого користувача.

Рисунок 2.10 – Навігаційна панель неавторизованого користувача

На рисунку 2.11 подано вигляд навігаційної панелі для авторизованого користувача при натисканні іконки «Особистий кабінет»

Рисунок 2.11 – Навігаційна панель авторизованого користувача

Маршрутизація перевірялася на правильність перемикання між різними сторінками застосунку.



Рисунку 2.12 – Головна сторінка інтернет-магазину "Air Baking"

Маршрути, такі як "/" для головної сторінки, зображеної на рисунку 2.12, "/cart" для сторінки корзини, зображеної на рисунку 2.13 та "/orders" для сторінки замовлення, зображеної на рисунку 2.14, були перевірені на коректність перемикання між компонентами.

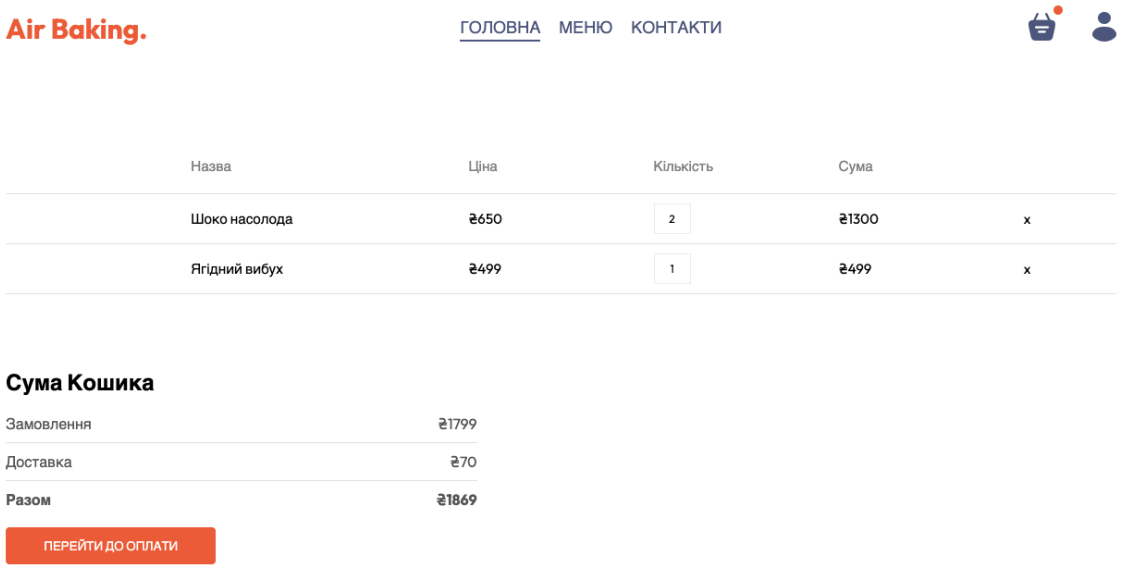


Рисунок 2.13 – Сторінка кошика інтернет-магазину "Air Baking"

Рисунок 2.14 – Список замовлень інтернет-магазину "Air Baking"

Інтеграція з бекендом перевіряється на правильність взаємодії фронтенду з API бекенду. Наприклад, при додаванні товару до корзини фронтенд відправляв

відповідний запит до сервера та коректно відображав результат.

При тестуванні форм перевірялася коректність роботи форм, включаючи валідацію даних та обробку помилок. Наприклад, форма входу, подана на рисунку 2.15, перевіряє ведені дані та відображає помилки у випадку неправильного введення.

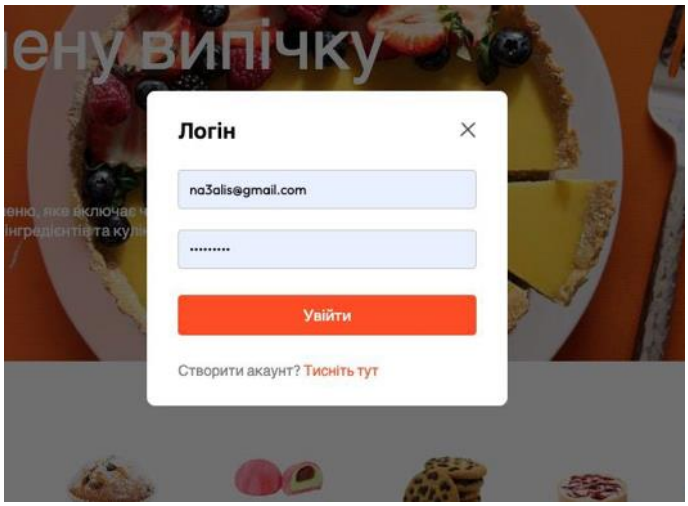


Рисунок 2.15 – Форма авторизації

Для тестування роботи компонента FoodItem, що відповідає за вигляд карток продуктів, було використано дію додавання продукту в кошик. Вигляд карток продуктів, які додали у кошик двічі, одиночно та недодали подано на рисунку 2.16.

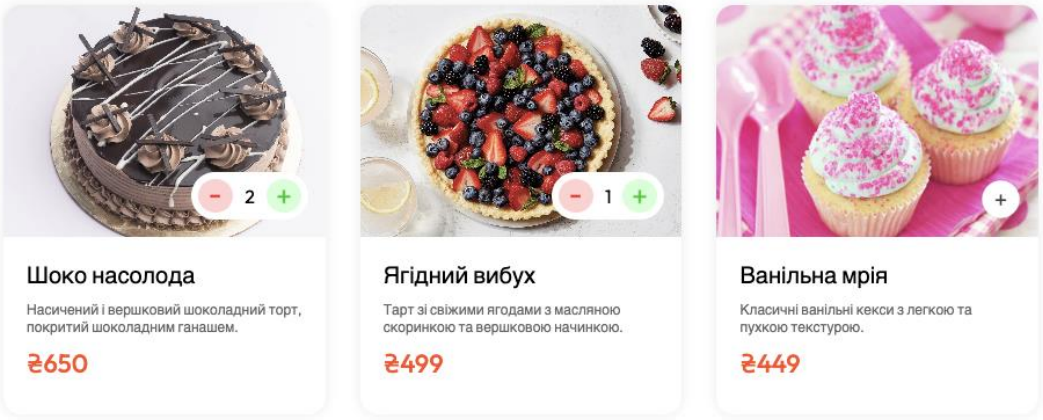


Рисунок 2.16 – Картки продуктів

Тестування інтернет-магазину "Air Baking" засвідчило належну якість реалізації серверної та клієнтської його частин. При тестування були виявлені та усунені помилки на ранніх етапах розробки, що дозволило підвищити надійність та стабільність системи, а також гарантувати, що всі функціональні можливості веб-сайту працюють належним чином.

РОЗДІЛ 3. СЕРВІСНА ЧАСТИНА. АДМІНІСТРУВАННЯ ТА ПІДТРИМКА ІНТЕРНЕТ-МАГАЗИНУ "AIR BAKING"

3.1 Інструкція з розміщення сайту в Інтернеті

Розглянемо порядок розміщення інтернет-магазину на хостингу (деплой), який складається з трьох частин:

– розміщення база даних (MongoDB Atlas);
– розміщення backend (Node.js, деплой на Render);
– розміщення frontend (Vite + React, деплой на GitHub Pages).

Перейдемо до розміщення в хмарі бази даних.

Для початку реєструємося та входимо в MongoDB Atlas. Після входу створюємо новий проект, натиснувши кнопку "New Project", вводимо назву проекту та натискаємо "Next".

У проекті натискаємо "Build a Cluster", обираємо безкоштовний кластер (M0 Sandbox) та налаштовуємо його за власним бажанням, після чого натискаємо "Create Cluster".

Після створення кластеру налаштовуємо доступ до бази даних: натискаємо "Database Access" у меню зліва, натискаємо "Add New Database User" та створюємо користувача з паролем.

Далі переходимо до "Network Access", натискаємо "Add IP Address" та обираємо "Allow access from anywhere" або додаємо свою IP-адресу.

Після налаштування кластеру створюємо нову базу даних та колекцію, натиснувши "Clusters" у меню зліва і "Collections" біля кластеру.

Для отримання URI для підключення натискаємо "Clusters", біля кластеру натискаємо "Connect", обираємо "Connect your application" та копіюємо URI для підключення до бази даних, який буде використовуватися у вашому backend-коді.

Розглянемо порядок розміщення в інтернеті backend-частини інтернет-магазину "Air Baking". Деплой серверної частини будемо здійснювати на

платформі хостингу render.com, яка пропонує різні послуги для розгортання веб-застосунків, статичних сайтів, баз даних та інших сервісів.

Після авторизації на Render створюємо новий веб-сервіс, натиснувши "New" -> "Web Service". Початок створення нового веб-сервісу поданий на рисунку 3.1.

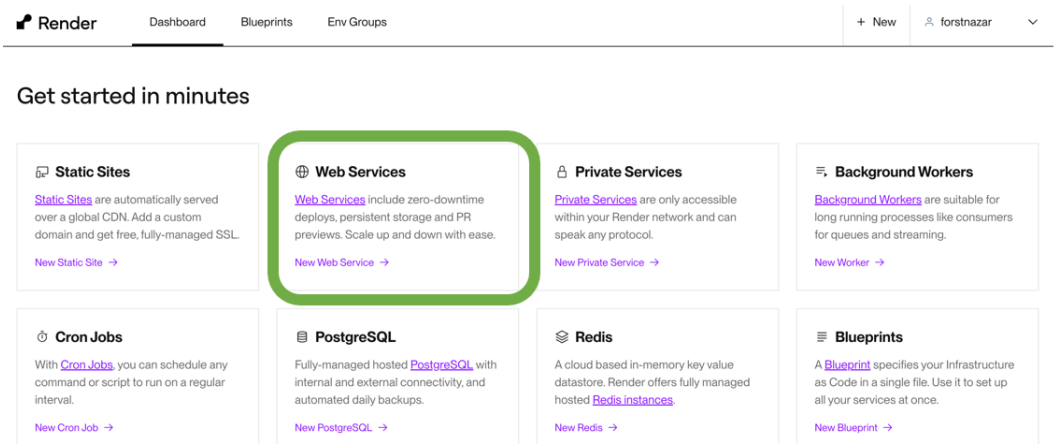


Рисунок 3.1 – Створення веб-сервісу на платформі render.com

Підключаємо GitHub репозиторій, що містить код серверної частини інтернет-магазину "Air Baking". Для цього вибираємо “Build and deploy from Git repository”, а далі – “Configure account” біля іконки GitHub. Вибір GitHub-репозиторію поданий на рисунку 3.2.

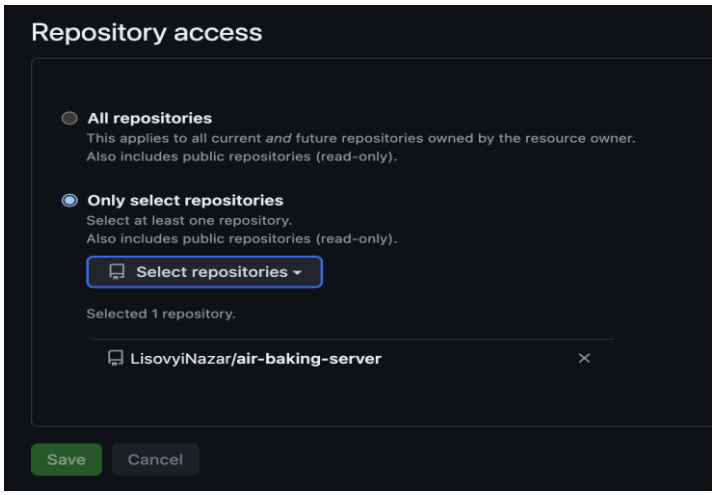


Рисунок 3.2 – Вибір GitHub-репозиторію

Тобто обираємо репозиторій та гілку з кодом backend частини інтернет-магазину "Air Baking", заповнюємо форму, вибираючи Node.js як середовище виконання, та вказуємо команду для запуску (наприклад, `npm start` або `node index.js`).

Далі додаємо змінні оточення, такі як URI для підключення до MongoDB Atlas. Вікно конфігурації змінних оточення подано на рисунку 3.3.

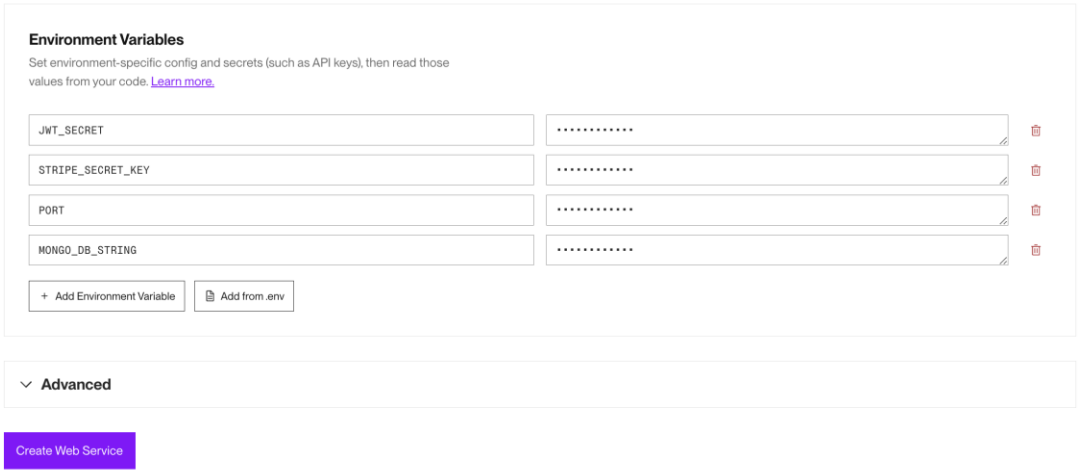


Рисунок 3.3 – Налаштування змінних оточення

Після налаштування натискаємо "Create Web Service", Render автоматично створює та запускає backend інтернет-магазину "Air Baking" та видає відповідне повідомлення, подане на рисунку 3.4, про успішний старт сервісу.

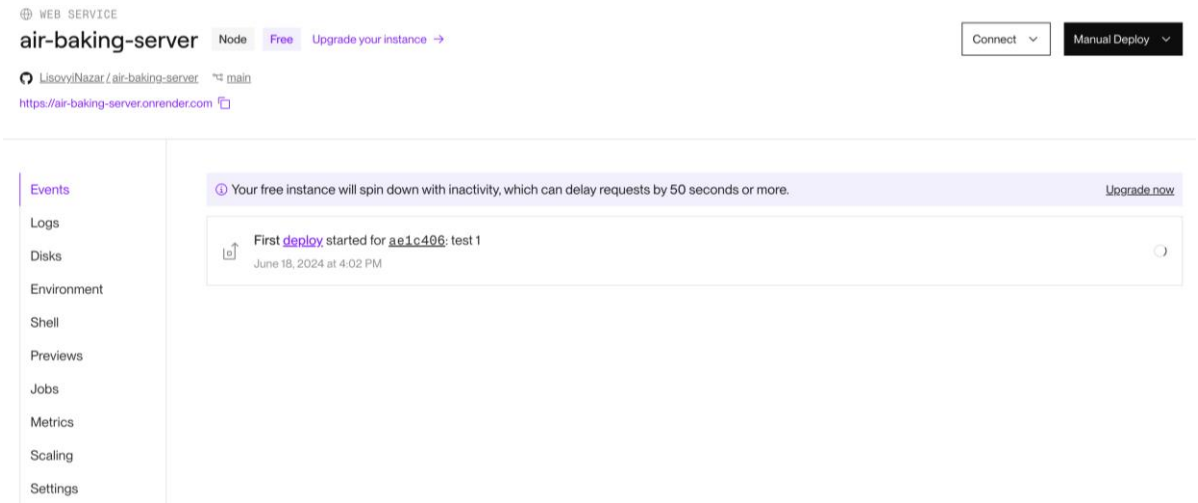


Рисунок 3.4 – Повідомлення Render про успішний старт сервісу

Деплой frontend-частини інтернет-магазину "Air Baking" реалізовано на GitHub Pages.

Для початку налаштовуємо репозиторій на GitHub, створюючи новий або використовуючи вже існуючий. Переконуємося, що фронтенд-код знаходиться в цьому репозиторії.

У проекті встановлюємо Vite, виконавши команду `npm install vite`.

Додаємо у файл `package.json` наступний скрипт:

```
"scripts": {
  "build": "vite build",
  "deploy": "gh-pages -d dist" }
```

Встановлюємо пакет `gh-pages` командою `npm install gh-pages --save-dev`.

Далі запускаємо команду для створення білду `npm run build` та команду для деплою `npm run deploy`.

Переходимо до налаштувань репозиторію на GitHub, у розділі "Pages" обираємо гілку `gh-pages` як джерело для GitHub Pages.

Після активації GitHub Pages сайт буде доступний за отриманою URL-адресою.

3.2 Інструкція з обслуговування та наповнення інтернет-магазину "Air Baking"

Розглянемо процеси, необхідні для підтримки функціональності сайту та регулярного оновлення його вмісту. З цією метою реалізована адміністративна частина сайту як окремий клієнт-проект, що забезпечує зручний інтерфейс для керування контентом.

З метою забезпечення належного рівня обслуговування інтернет-магазину "Air Baking" важливо організувати його моніторинг та підтримку працездатності. Для цього потрібно регулярно перевіряти стан основних компонентів системи, включаючи базу даних, серверну та клієнтську частини. З цією метою будуть

використовуватися відповідні інструменти для моніторингу серверів та мережевої інфраструктури, тобто Render.com для бекенду та GitHub Pages для фронтенду. В разі виявлення проблем потрібно негайно виконати необхідні дії для їх усунення, включаючи рестарт серверів, перевірку логів та відновлення з резервних копій.

Для підтримки належного рівня безпеки експлуатації інтернет-магазину "Air Baking" важливо постійно стежити за оновленнями безпеки для всіх використовуваних бібліотек і фреймворків. Регулярно перевіряти конфігурацію сервера на наявність вразливостей, використовувати SSL сертифікати для забезпечення безпеки передачі даних, а також дотримуватися політики безпеки для доступу до адміністративної частини сайту.

Для внесення змін на сайті потрібно увійдіть в адміністративну частину, використовуючи надані облікові дані. Адміністративна частина реалізована як окремий клієнтський проект і забезпечує зручний інтерфейс для керування вмістом сайту.

Для додавання нового продукту потрібно перейти до закладки "Створити" в адміністративній частині. Вигляд закладки поданий на рисунку 3.5.

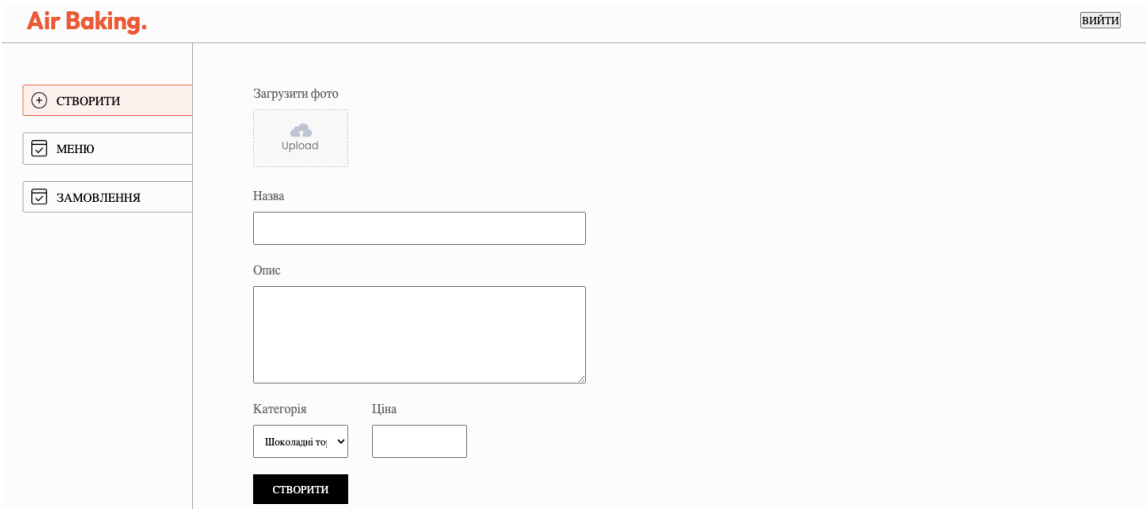


Рисунок 3.5 – Форма створення продукту у закладці «Створити»

В закладці "Створити" необхідно заповнити необхідні поля (назва, опис, ціна, категорія, зображення) та зберегти зміни.

Для редагування існуючого продукту, потрібно знайти його у списку, натиснути кнопку "Редагувати", внести необхідні зміни та зберегти їх. Для видалення продукту потрібно знайти його у списку та натиснути відповідну кнопку "Видалити". Перелік продуктів знаходиться у закладці “Меню”, яка подана на рисунку 3.6.

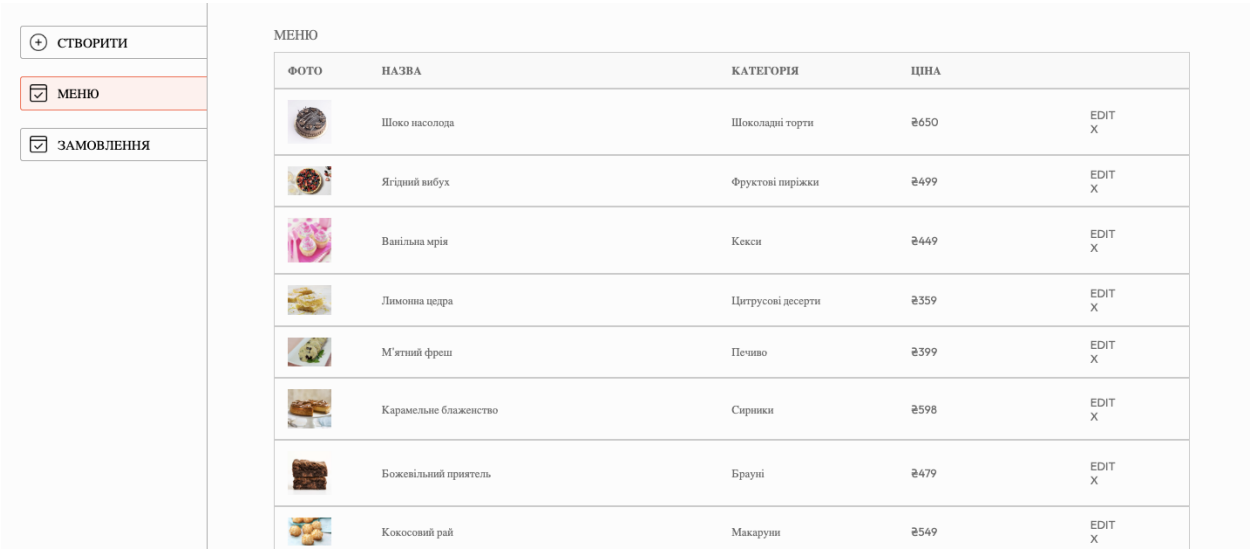


Рисунок 3.6 – Каталог продуктів інтернет-магазину "Air Baking"

Для керування замовленнями потрібно перейти до розділу "Замовлення" в адміністративній частині, яка подана на рисунку 3.7. Тут можна переглядати та керування замовленнями клієнтів.

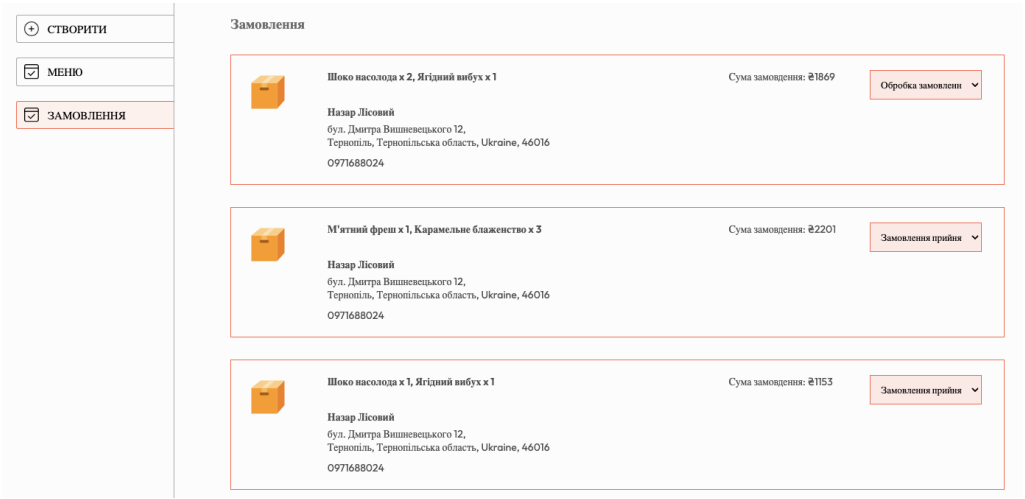


Рисунок 3.7 – Керування замовленнями

В даній закладці можна переглядати деталі замовлення, змінювати статус (обробляється, відправлено, доставлено) та контактувати з клієнтами у разі потреби.

Таким чином, дотримання даних інструкцій забезпечить ефективне управління та підтримку інтернет-магазину "Air Baking", дозволяючи зберігати його актуальність та конкурентоспроможність.

3.3 Інструкція з популяризації та підтримки інтернет-магазину "Air Baking"

Кроки для популяризації інтернет-магазину "Air Baking" та підтримки його активності охоплюють стратегію маркетингу, взаємодії з клієнтами та оптимізації роботи сайту. Для популяризація інтернет-магазину необхідно розробити привабливий логотип та слоган, які легко запам’ятовуються і асоціюються з продуктами інтернет-магазину "Air Baking". Важливо підтримувати єдиний стиль у всіх маркетингових матеріалах, включаючи веб-сайт, соціальні мережі, упаковку та рекламні оголошення.

Для інтернет-магазину важливо здійснити оптимізацію для пошукових систем (SEO). Для цього потрібно забезпечити оптимізацію контенту на сайті, включаючи ключові слова, мета-теги, заголовки та опис продуктів. Регулярно оновлювати рецепти та новинки, що допоможе підвищити видимість сайту у пошукових системах [37]. Важливо використовувати інструменти аналітики, такі як Google Analytics, для відстеження трафіку та ефективності SEO-стратегії.

З метою реклами в Інтернеті важливо використовувати контекстну рекламу через платформи Google Ads та соціальні мережі, такі як Facebook, Instagram, TikTok. Тобто потрібно створювати цільові рекламні кампанії для залучення нових клієнтів, збільшення продажів та підвищення впізнаваності бренду. Особливу увагу слід приділити ремаркетингу для повернення користувачів, які відвідали сайт, але не здійснили покупку.

У соціальних мережах можна створити профілі, у яких регулярно

публікувати якісний контент, включаючи фотографії продукції, рецепти, відгуки клієнтів і новини про акції та знижки.

Потрібно також регулярно аналізувати дані про відвідуваність сайту, конверсії, середній чек та інші ключові показники. Для цього можна скористатися хмарними інформаційно-технологічними платформами аналітичного опрацювання даних. Тези про відповідне дослідження [38] були опубліковані у матеріалах XI Міжнародної науково-практичної конференції молодих учених та студентів та подані в додатку Д кваліфікаційної роботи.

Аналітичне опрацювання великих за обсягом даних є ключовим аспектом підтримки інтернет-магазину "Air Baking". Завдяки сучасним технологіям аналітики даних, інтернет-магазин може збирати та аналізувати величезні масиви інформації, що надходять від користувачів, включаючи дані про покупки, перегляди товарів, поведінку на сайті та зворотний зв'язок [39]. Це дозволяє глибше розуміти потреби та вподобання клієнтів, оптимізувати асортимент товарів, вдосконалювати маркетингові стратегії та покращувати користувацький досвід. Крім того, аналітика даних допомагає виявляти тенденції ринку, прогнозувати попит і управляти запасами, що є критично важливим для ефективного функціонування інтернет-магазину. Тези про відповідне дослідження [40] були опубліковані у матеріалах XI Міжнародної науково-практичної конференції молодих учених та студентів та подані в додатку Д кваліфікаційної роботи. Використовувати ці дані для коригування маркетингової стратегії, асортименту продуктів та цінової політики. Впроваджувати нові функції та покращення на основі зворотного зв'язку від клієнтів та аналізу конкурентів.

Інформація на сайті, включаючи описи продуктів, ціни, новинки та статті в блозі, повинна бути завжди в актуальному стані.

Ці кроки допоможуть ефективно популяризувати та підтримувати інтернет-магазин "Air Baking", забезпечуючи його стабільний розвиток та зростання.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Моделювання та прогнозування небезпечних ситуацій

Інтернет-магазин "Air Baking", як і будь-який інший онлайн-бізнес, піддається численним загрозам, які можуть вплинути на його функціонування, безпеку даних користувачів та фінансові операції. Тому актуальність розкриття питання "Моделювання та прогнозування небезпечних ситуацій" в контексті інтернет-магазину "Air Baking" є вкрай важливою

Оскільки інтернет-магазину "Air Baking" оперує великою кількістю особистих даних клієнтів, включаючи імена, адреси, номери телефонів та платіжну інформацію, то важливо забезпечити захист персональних даних. Адже незахищеність цих даних може призвести до їх витоку або зловживання, що може спричинити фінансові втрати та юридичні наслідки для компанії. Моделювання небезпечних ситуацій допомагає ідентифікувати потенційні загрози та впровадити відповідні заходи для їх запобігання.

Також інтернет-магазини часто стають мішенню для шахраїв, які можуть здійснювати фальшиві транзакції або зламувати платіжні системи, тому важливою є фінансова безпека. Прогнозування таких ситуацій дозволяє розробити стратегії для мінімізації ризиків, включаючи використання сучасних методів шифрування та двофакторної аутентифікації.

Технічні проблеми, такі як відмови серверів або збої в мережі, можуть призвести до недоступності магазину для клієнтів. Крім того, кібернапади, такі як DDoS-атаки, можуть паралізувати роботу магазину. Моделювання таких ситуацій допомагає розробити плани дій у випадку надзвичайних ситуацій, включаючи резервне копіювання даних та плани відновлення після збоїв.

Клієнти та співробітники можуть стати жертвами атак соціальної інженерії або фішингу, коли зловмисники намагаються отримати доступ до конфіденційної інформації шляхом обману. Прогнозування таких загроз

допомагає розробити освітні програми для підвищення обізнаності та навичок безпечної роботи в інтернеті.

Моделювання та прогнозування небезпечних ситуацій у контексті інтернет-магазину "Air Baking" передбачає визначення потенційних ризиків і впровадження заходів для їхньої мінімізації. Для AirBaking основними питаннями можуть бути порушення безпеки даних, кібератаки та збої в системі. Ці проблеми можуть поставити під загрозу дані клієнтів, порушити роботу бізнесу та завдати шкоди репутації компанії.

Регулярна оцінка потенційних ризиків для інтернет-магазину, включаючи кіберзагрози та вразливості системи [41], є важливою для стабільної роботи. На рисунку 4.1 подана матриця оцінки ризиків небезпечних ситуацій.

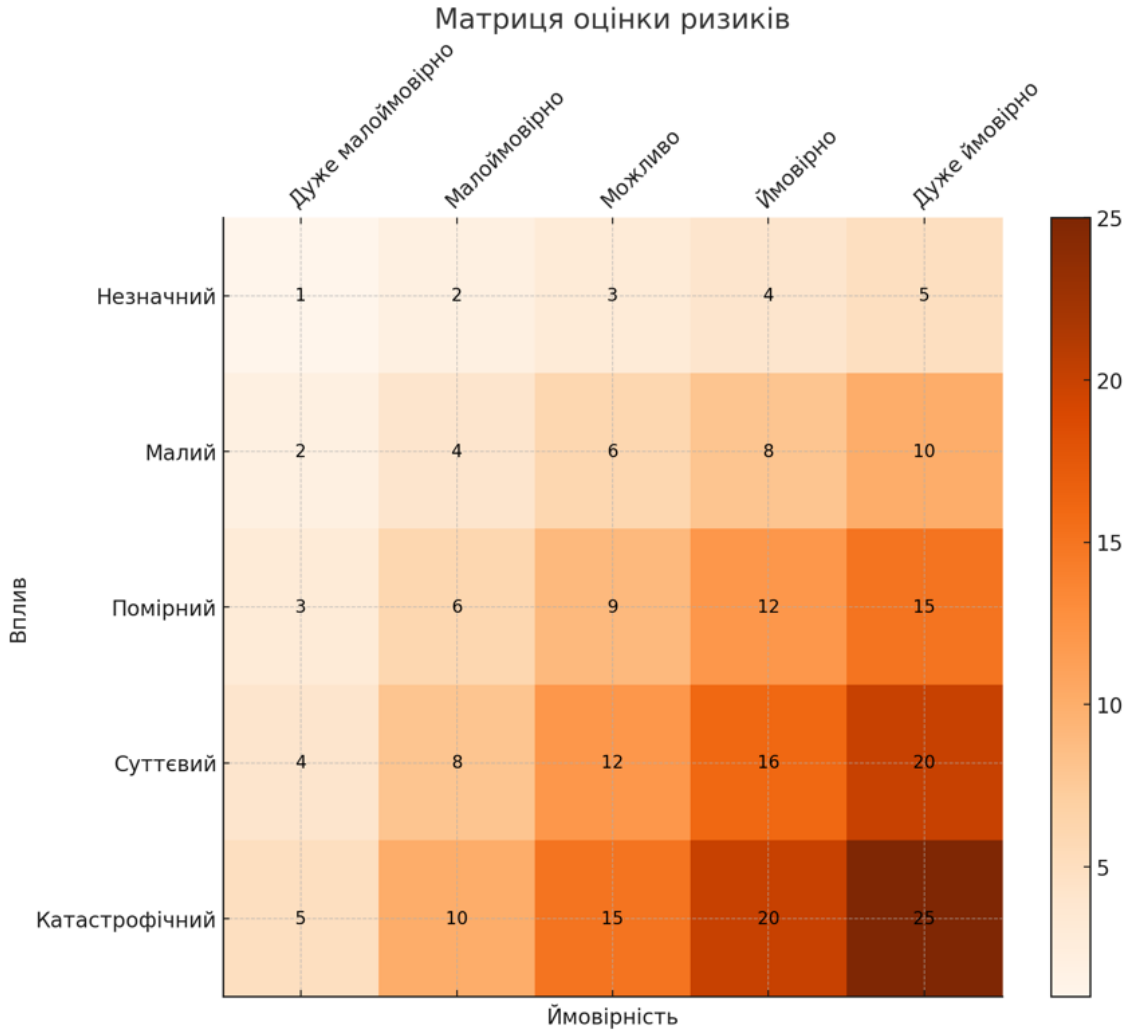


Рисунок 4.1 – Матриця оцінки ризиків

Розглянемо потенційні ризики для інтернет-магазину "Air Baking" та їх оцінка за створеною матрицею оцінки ризиків [41]:

1. Кібератаки. Ймовірність: дуже ймовірно. Вплив: катастрофічний. Оцінка - 25 (дуже високий ризик).

2. Витоки даних клієнтів. Ймовірність: ймовірно. Вплив: суттєвий. Оцінка:16 (високий ризик).

3. Технічні збої в роботі інтернет-магазину "Air Baking". Ймовірність: можливо. Вплив: помірний. Оцінка: 9 (середній ризик).

4. Проблеми з постачанням товарів. Ймовірність: малоймовірно. Вплив: незначний. Оцінка: 4 (низький ризик).

5. Неправильне оброблення замовлень. Ймовірність: можливо. Вплив: незначний. Оцінка: 6 (низький ризик).

6. Пожежа на складі. Ймовірність: дуже малоймовірно. Вплив: катастрофічний. Оцінка: 5 (низький ризик).

7. Відмова від співпраці ключових постачальників. Ймовірність: малоймовірно. Вплив: суттєвий. Оцінка: 8 (середній ризик).

8. Регуляторні зміни (нові закони). Ймовірність: можливо. Вплив: суттєвий. Оцінка: 12 (середній ризик).

Ці оцінки ризиків допоможуть ідентифікувати найбільш критичні зони, на які потрібно звернути увагу для забезпечення стабільної та безпечної роботи інтернет-магазину.Ці оцінки допоможуть ідентифікувати найбільш критичні зони, на які потрібно звернути увагу для забезпечення стабільної та безпечної роботи інтернет-магазину.

Розглянемо заходи безпеки для інтернет-магазину "Air Baking", які потрібно впровадити з метою мінімізації ризиків виникнення небезпечних ситуацій. До таких заходів належить впровадження надійних протоколів безпеки включає SSL-шифрування, фаєрволи та системи виявлення вторгнень (IDS/IPS). SSL-шифрування захищає дані, передані між сервером та клієнтом, від перехоплення та маніпуляцій. Фаєрволи захищають мережу від несанкціонованого доступу, фільтруючи вхідний та вихідний трафік. Системи

виявлення вторгнень моніторять мережевий трафік та виявляють потенційні загрози в реальному часі.

Іншим актуальним заходом безпеки інтернет-магазину "Air Baking" є резервне копіювання даних. Регулярне зберігання резервних копій є критично важливим для захисту даних. Частота резервного копіювання може бути щоденною або щотижневою, залежно від обсягів змін. Резервні копії слід зберігати в декількох фізичних та хмарних локаціях для забезпечення надійності. Регулярне тестування процесів відновлення даних з резервних копій допомагає упевнитися у їхній ефективності.

При моделюванні та прогнозуванні небезпечних ситуацій для інтернет-магазину "Air Baking" важливо мати план реагування на інциденти. Розробка та регулярне оновлення плану реагування на інциденти включає визначення потенційних загроз, таких як кібератаки, витоки даних та технічні збої. Призначення відповідальних осіб та груп для реагування на інциденти є необхідним для ефективного управління. Процедури реагування мають включати кроки для швидкого виявлення, ізоляції та нейтралізації інцидентів. Регулярні тренування та симуляції інцидентів допомагають підготувати команду до реальних ситуацій.

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Вимоги ергономіки до організації робочого місця оператора ПК є вкрай актуальним при розробці інтернет-магазину "AirBaking". Це пояснюється кількома ключовими аспектами [42]:

– Правильно організоване робоче місце з урахуванням ергономічних вимог допомагає знизити ризик розвитку професійних захворювань, таких як синдром зап'ястного каналу, болі в спині та шиї, проблеми з зором та інші. Це особливо важливо для працівників, які проводять значний час за комп'ютером.

– Ергономічно облаштоване робоче місце сприяє підвищенню ефективності роботи. Зручне розташування обладнання та оптимальна робоча

поза зменшують втому та дискомфорт, що дозволяє працівникам концентруватися на своїх завданнях і працювати продуктивніше.

– Забезпечення комфортних умов праці демонструє турботу роботодавця про своїх співробітників. Це позитивно впливає на мотивацію, зменшує плинність кадрів та підвищує загальний рівень задоволеності роботою.

– Комфортне робоче середовище дозволяє працівникам більш уважно та точно виконувати свої завдання, що важливо для розробки якісного та функціонального інтернет-магазину.

– У багатьох країнах існують нормативні акти, які регулюють умови праці, включаючи вимоги до ергономіки робочих місць. Виконання цих вимог допомагає уникнути юридичних проблем і штрафів.

Організація робочого місця оператора ПК повинна забезпечувати не лише комфорт, але й сприяти збереженню здоров'я та підвищенню продуктивності. Особливо це важливо під час розробки таких проектів, як інтернет-магазин "AirBaking", де оператори працюють довгий час за комп'ютером.

Робочий стіл має бути достатньо просторим для розміщення всіх необхідних пристроїв та матеріалів. Важливо, щоб на столі вистачало місця для монітора, клавіатури, мишки та інших інструментів, таких як документи чи блокноти. Це допоможе зберігати порядок і забезпечить легкий доступ до всього необхідного. Робочий стілець повинен бути зручним, з регульованою висотою та підтримкою спини. Це дозволить уникнути дискомфорту та болю в спині, що часто виникають при довготривалому сидінні.

Правильне розташування обладнання є критичним для збереження здоров'я. Монітор повинен бути розташований на відстані 50-70 см від очей користувача і на рівні очей або трохи нижче. Це зменшить навантаження на очі та шию. Клавіатура і мишка повинні бути розташовані так, щоб руки залишалися в природній, розслабленій позиції. Ергономічні параметри облаштування робочого місця подані на рисунку 4.2.

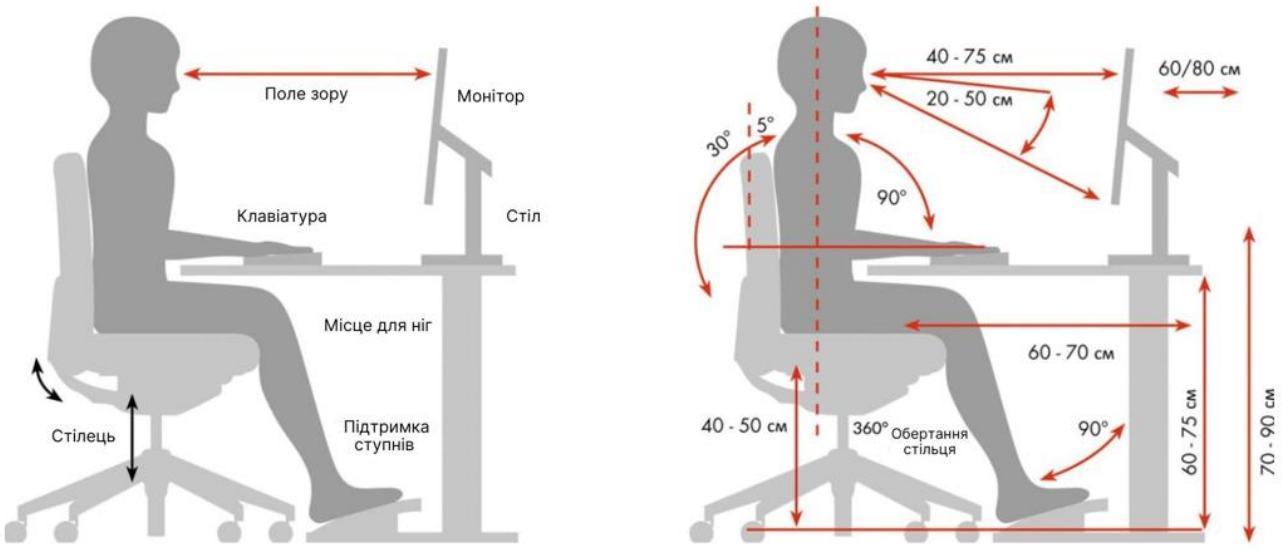


Рисунок 4.2 - Ергономічні параметри облаштування робочого місця

Робоче місце повинно мати хороше освітлення. Найкраще, якщо це буде природне світло, адже воно найменш шкідливе для очей. Однак, штучне освітлення також повинно бути належним чином організоване: воно має бути рівномірним і не створювати відблисків на моніторі. Використання антивідблискових засобів, таких як матові екрани або накладки, допоможе зменшити навантаження на очі.

Правильна позиція тіла є важливою для збереження здоров'я при роботі за комп'ютером. Сидіти потрібно так, щоб ноги стояли на підлозі або на підставці, а коліна були зігнуті під кутом приблизно 90 градусів. Це допомагає зберегти природну кривизну хребта. Руки повинні бути розташовані так, щоб зап'ястя залишалися в нейтральній позиції, а передпліччя були паралельні підлозі або трохи нахилені вниз.

Для підтримання високого рівня продуктивності та зниження навантаження на очі і тіло, важливо робити регулярні перерви. Короткі перерви кожні 20-30 хвилин дозволять розтягнути м'язи і відпочити очам. Мікроперерви, наприклад, короткі вправи для очей або рук кожні 10-15 хвилин, також є дуже корисними. Це можуть бути прості вправи, такі як переведення погляду з монітора на віддалений об'єкт або обертання зап'ясть.

Комфортні умови праці включають в себе не лише правильне розташування робочого місця, але й забезпечення належного мікроклімату. Температура в приміщенні повинна бути комфортною (20-24 °C), а повітря - свіжим завдяки регулярній вентиляції. Шумовий фон повинен бути мінімальним, оскільки надмірний шум може відволікати і знижувати продуктивність.

Ефективна організація робочого процесу також є важливим аспектом ергономіки. Завдання повинні бути чітко структуровані і розподілені, щоб уникнути перевантаження та стресу. Важливою є також соціальна складова: можливість для спілкування з колегами та підтримка командного духу сприяють зниженню рівня стресу та підвищенню задоволеності роботою.

Таким чином дотримання ергономічних вимог при організації робочого місця оператора ПК є важливим для підтримання здоров'я працівників та підвищення їх продуктивності. Впровадження цих рекомендацій допоможе створити комфортні умови роботи, знизити ризик професійних захворювань і сприятиме ефективній розробці інтернет-магазину "AirBaking".

Отже, забезпечення безпеки даних інтернет-магазину є вкрай важливим. За допомогою моделювання та прогнозування небезпечних ситуацій для інтернет-магазину "AirBaking" може проактивно вирішувати потенційні загрози та мінімізувати їхній вплив. Одночасно, впровадження ергономічних принципів на робочих місцях покращує комфорт і продуктивність співробітників, знижуючи ризик виробничих травм. Разом ці заходи створюють безпечне та ефективне робоче середовище, що в кінцевому підсумку сприяє загальному успіху та сталому розвитку інтернет-магазину "AirBaking".

ВИСНОВКИ

В кваліфікаційній роботі було здійснено розробку інтернет-магазину "Air Baking" засобами MERN-стеку. Виконані дослідження та практичні заходи дозволяють зробити наступні висновки:

У першому розділі проведено аналітичний огляд існуючих платформ та підходів для створення інтернет-магазинів. Це дозволило визначити основні переваги та недоліки різних технологій і обґрунтувати вибір MERN-стеку для реалізації проекту "Air Baking".У результаті сформовано детальне технічне завдання, яке включає область застосування, призначення розробки, функціональні та нефункціональні вимоги, вимоги до програмної документації, техніко-економічні показники, а також стадії та етапи розробки проекту. Це дозволило чітко окреслити цілі і задачі проекту.

В другому розділі кваліфікаційної роботи обґрунтовано вибір архітектури проекту та розроблено структуру бази даних, серверної та клієнтської частини. Використання MERN-стеку дозволило забезпечити гнучкість, масштабованість та ефективність розробки. Тут реалізовано всі основні компоненти проекту: серверну частину на базі Node.js та Express.js, клієнтську частину за допомогою React.js, а також базу даних на MongoDB. Проведено функціональне тестування всіх компонентів, що дозволило виявити та усунути помилки, забезпечивши стабільну роботу системи.

У третьому розділі проаналізовано та розроблено інструкції з розміщення, обслуговування та наповнення інтернет-магазину "Air Baking", а також його популяризації та підтримки. Це забезпечує безперервну роботу сайту та його розвиток.

У четвертому розділі розглянуто моделювання та прогнозування небезпечних ситуацій, пов'язаних з роботою інтернет-магазину, а також вимоги ергономіки до організації робочого місця оператора ПК. Це дозволяє забезпечити безпеку даних та комфортні умови праці для працівників.

Виконання даної кваліфікаційної роботи продемонструвало можливості та

переваги використання MERN-стеку для розробки сучасних інтернет-магазинів. Отримані результати можуть бути використані для подальшого розвитку проекту "Air Baking" та інших аналогічних проектів у сфері електронної комерції.

ПЕРЕЛІК ДЖЕРЕЛ

1. Мельник, А.М. Архітектура сучасних веб-додатків. Київ : Видавничий дім «Києво-Могилянська академія», 2021, 55.

2. Воробйов, О.О. Розробка сучасних веб-додатків: основи та приклади. Дніпро : Журфонд, 2019, 157-158.

3. Dyer, R. Practical Node.js: Building Real-World Scalable Web Apps. Apress, 2019, 230.

4. Carver, T. Shopify: Beginner to Pro Guide: The Comprehensive Guide. Independently Published, 2021, 124.

5. Jones, S. Building E-commerce Sites with Shopify. Packt Publishing, 2018, 167.

6. Edmonds, R. WooCommerce Explained: Your Step-by-Step Guide to WooCommerce. OSTraining, 2020, 244-245.

7. Young, M. WooCommerce: Beginner to Pro Guide. Independently Published, 2021, 165.

8. Brown, L. WooCommerce Development. Independently Published, 2020, 123.

9. Meloni, J. Magento 2 for Beginners. Que Publishing, 2020, 204.

10. Fitzpatrick, B. Magento 2 Development Cookbook. Packt Publishing, 2020, 317.

11. Turner, T. BigCommerce Essentials. Packt Publishing, 2020, 102.

12. Duda O., Kunanets N., Matsiuk O., Pasichnyk V., Cloud-based IT Infrastructure for “Smart City” Projects, in Dependable IoT for Human and Industry: Modeling, Architecting, Implementation. River Publishers, 2018. P. 389-410.

13. Wieclaw L., Pasichnyk V., Kunanets N., Duda O., Matsiuk O., Falat P. Cloud computing technologies in “smart city” projects. In Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Bucharest: Romania, 21–23 September 2017. pp. 339–342.

14. "Duda O., Kunanets N., Matsiuk O., Pasichnyk V. Information-Communication Technologies of IoT in the ""Smart Cities"" Projects"", CEUR Workshop Proceedings. 2018. Vol. 2105. P. 317- 330."

15. Nixon, R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. O'Reilly Media, 2018, 456-457.

16. Mele, A. Django 3 By Example. Packt Publishing, 2020. 314.

17. Vincent, W. S. Django for Beginners: Build websites with Python and Django. Independently published, 2019, 54.

18. Hartl, M. Ruby on Rails Tutorial: Learn Web Development with Rails. Addison-Wesley Professional, 2019, 413.

19. Turvey, D. Mastering MERN: Building Modern Web Applications with Mongo, Express, React, and Node. Packt Publishing, 2020, 117.

20. Eisenberg, S. Full-Stack React Projects. Packt Publishing, 2020, 204.

21. Кузьменко, П.Т. MongoDB: від новачка до професіонала. К.: Вища школа, 2018, 141.

22. Сидоренко, І.О. MERN-стек: розробка повного циклу веб-застосунків Харків: Основа, 2020, 104.

23. Кравчук, Н.П. Практика програмування на React. К.: Вид. група «КМ-Букс», 2021, 54 .

24. Половий, Д.В. Архітектура REST API: принципи та практики. Дніпро: Видавництво Дніпро, 2021, 117.

25. Walker, A. Building Progressive Web Apps with React. Packt Publishing, 2019, 280 p.

26. Varma, S. Express in Action: Writing, building, and testing Node.js applications. Manning Publications, 2021, 165.

27. Murphy, T. High Performance Browser Networking. O'Reilly Media, 2020, 221.

28. Watson, M. Full Stack JavaScript Development with MEAN. O'Reilly Media, 2020, 154.

29. Newman, S. Building Microservices. O'Reilly Media, 2021, 305.

30. Jamison, C. REST API Development with Node.js. Packt Publishing, 2021, 104.

31. Матеріали VI Міжнародної студентської науково-технічної конференції. Тернопіль: Тернопільський національний технічний університет ім. І.Пулюя (м. Тернопіль, 27-28 квітня 2023 р.), 2023.- сс. 131, 175

32. Butcher, P. MongoDB: The Definitive Guide. O'Reilly Media, 2021, 57.

33. Griswold, B. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node. Apress, 2018, 241.

34. Lamm, M. Hands-On Full Stack Development with Node.js and MongoDB. Packt Publishing, 2019, 305.

35. Rubin, R. Pro Express.js: Master Express.js: The Node.js Framework For Your Web Development, Apress, 2019, 107-108.

36. Mackenzie, K. Advanced Node.js Development. Apress, 2021, 203.

37. Pena, W. SEO Secrets: Discover Why SEO Is by Far the Most Important Skill to Learn in 2022. Independently published, 2022, 104.

38. Актуальні задачі сучасних технологій : зб. тез доповідей XI міжнар. наук.-практ. конф. Молодих учених та студентів, (Тернопіль, 7-8 грудня 2022) / М-во освіти і науки України, Тернопільський національний технічний університет імені Івана Пулюя [та ін.]. Тернопіль: ФОП Паляниця В. А., 2022, с. 169

39. Duda O., Kochan V., Kunanets N., Matsiuk O., Pasichnyk V., Sachenko A., Pytlenko T. Data processing in IoT for smart city systems. In Proceedings of the 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, 18–21 September 2019; Volume 1, pp. 96–99.

40. Матеріали X науково-технічної конфції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 7–8 грудня 2022 р.). Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2022. с. 30

41. Русанов, М. Г., Русанов, Н. В. Чернікова, І. М. Шепель. М. Г. Безпека

життєдіяльності: навч. посібник. Комунальний заклад «Харківська гуманітарнопедагогічна академія» Харківської обласної ради, Харків: 2019, 57-59.

42. Сучасні вимоги ергономіки робочого місця оператора комп’ютера. URL: https://oppb.com.ua/news/suchasni-vymogy-ergonomiky-robochogo-miscya-operatora-kompyutera (дата звернення: 30.05.2024)

ДОДАТКИ

Додаток А

Лістинг коду для оголошення схем та моделей бази даних інтернет-магазину "Air Baking"

```
// Імпортуємо бібліотеку Mongoose для роботи з MongoDB
import mongoose from "mongoose";

// Оголошуємо схему для колекції "food"
// Схема визначає структуру документів у колекції
const foodSchema = new mongoose.Schema({
    name: { type: String, required: true }, // Назва виробу
    description: { type: String, required: true }, // Опис виробу
    price: { type: Number, required: true }, // Ціна виробу
    image: { type: String, required: true }, // зображення
    category: { type: String, required: true } // категорія c});

// Створюємо або отримуємо модель для колекції "food"
// Перевіряємо, чи модель "food" вже існує в mongoose.models
// Якщо існує, то використовуємо її, якщо ні — створюємо нову модель
на основі схеми foodSchema
const foodModel = mongoose.models.food || mongoose.model("food",
foodSchema);

// Експортуємо модель для подальшого використання в інших частинах
додатку
export default foodModel;
```

Додаток Б

Лістинг коду файлу server.js

```
import express from "express"; // Імпортуємо фреймворк Express для створення веб-сервера
import cors from 'cors'; // Імпортуємо модуль cors для налаштування політики CORS
import { connectDB } from "./config/db.js"; // Імпортуємо функцію для підключення до бази даних
import userRouter from "./routes/userRoute.js"; // Імпортуємо маршрути для користувачів
import foodRouter from "./routes/foodRoute.js"; // Імпортуємо маршрути для продуктів
import 'dotenv/config'; // Імпортуємо та завантажуємо змінні середовища з файлу .env
import cartRouter from "./routes/cartRoute.js"; // Імпортуємо маршрути для корзин
import orderRouter from "./routes/orderRoute.js"; // Імпортуємо маршрути для замовлень
// Конфігурація додатку
const app = express(); // Створюємо екземпляр додатку Express
const port = process.env.PORT || 4000; // Визначаємо порт для сервера, використовуючи змінну середовища або порт 4000 за замовчуванням
// Підключення проміжного програмного забезпечення (middlewares)
app.use(express.json()); // Додаємо middleware для парсингу JSON в запитах
app.use(cors()); // Додаємо middleware для налаштування CORS
// Підключення до бази даних
connectDB(); // Викликаємо функцію для підключення до бази даних
// Визначення маршрутів API
app.use("/api/user", userRouter); // Використовуємо маршрути для користувачів
app.use("/api/food", foodRouter); // Використовуємо маршрути для продуктів
app.use("/images", express.static('uploads')); // Додаємо статичний маршрут для завантажених зображень
app.use("/api/cart", cartRouter); // Використовуємо маршрути для корзин
app.use("/api/order", orderRouter); // Використовуємо маршрути для замовлень
// Основна точка доступу
app.get("/", (req, res) => {
    res.send("API Working"); // Відправляємо відповідь "API Working" при доступі до кореневого маршруту
});
// Запуск сервера
app.listen(port, () => console.log(`Server started on http://localhost:${port}`)); // Запускаємо сервер і виводимо повідомлення про успішний старт
```

Додаток В

Лістинг коду файлу frontend\src\App.jsx

```
import React, { useState } from 'react' // Імпорт React та хука useState
import Home from './pages/Home/Home' // Імпорт компонента для головної сторінки
import Footer from './components/Footer/Footer' // Імпорт компонента для футера
import Navbar from './components/Navbar/Navbar' // Імпорт компонента для навігаційної панелі
import { Route, Routes } from 'react-router-dom' // Імпорт компонентів для маршрутизації
import Cart from './pages/Cart/Cart' // Імпорт компонента для сторінки корзини
import LoginPopup from './components/LoginPopup/LoginPopup' // Імпорт компонента для вікна входу
import PlaceOrder from './pages/PlaceOrder/PlaceOrder' // Імпорт компонента для сторінки оформлення замовлення
import MyOrders from './pages/MyOrders/MyOrders' // Імпорт компонента для сторінки моїх замовлень
import { ToastContainer } from 'react-toastify'; // Імпорт контейнера для повідомлень
import 'react-toastify/dist/ReactToastify.css'; // Імпорт стилів для повідомлень
import Verify from './pages/Verify/Verify' // Імпорт компонента для сторінки верифікації
const App = () => {
  const [showLogin, setShowLogin] = useState(false); // Створення стану для відображення вікна входу
  return (
    <>
      <ToastContainer /> {/* Контейнер для повідомлень */}
      {showLogin ? <LoginPopup setShowLogin={setShowLogin} /> : null} {/* Відображення вікна входу за потреби */}
      <div className='app'>
        <Navbar setShowLogin={setShowLogin} /> {/* Компонент навігаційної панелі з передачею функції для відкриття вікна входу */}
        <Routes> {/* Налаштування маршрутів */}
          <Route path='/' element={<Home />} /> {/* Маршрут для головної сторінки */}
          <Route path='/cart' element={<Cart />} /> {/* Маршрут для сторінки корзини */}
          <Route path='/order' element={<PlaceOrder />} /> {/* Маршрут для сторінки оформлення замовлення */}
          <Route path='/myorders' element={<MyOrders />} /> {/* Маршрут для сторінки моїх замовлень */}
          <Route path='/verify' element={<Verify />} /> {/* Маршрут для сторінки верифікації */}
```

```
        </Routes>
      </div>
      <Footer /> {/* Компонент футера */}
    </>
  )
}
export default App // Експорт компонента App
```

Додаток Г

Тези VI Міжнародної студентської науково - технічної конференції

"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"

Міністерство освіти і науки України,
Тернопільський національний технічний університет
імені Івана Пулюя
Маріборський університет (Словенія)
Технічний університет в Кошице (Словаччина)
Каунаський технологічний університет (Литва)
Львівський національний університет
імені Івана Франка,
Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
Луцький національний технічний університет,
Чернівецький національний університет
імені Юрія Федьковича,
Вроцлавський економічний університет (Польща)
Університет технологій та економіки
імені Хелени Ходковської (Польща)
Донбаська державна машинобудівна академія

Студентське наукове товариство

VI МІЖНАРОДНА

студентська науково - технічна конференція

"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ.

АКТУАЛЬНІ ПИТАННЯ"

27-28 квітня 2023 р.

(збірник тез конференції)

Тернопіль 2023

ББК 72+34 (Укр)
М34

Матеріали VI Міжнародної студентської науково - технічної конференції / Тернопіль: Тернопільський національний технічний університет ім. І.Пулюя (м. Тернопіль, 27-28 квітня 2023 р.), 2023.- 348 с.

VI Міжнародна студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"

УДК 004.9

Гарматюк Н. – асп., Лісовий Н. – ст. гр. СНс-32, Дуда В. – ст. гр. СН-11
Тернопільський національний технічний університет імені Івана Пулюя

РОЛЬ ІННОВАЦІЙНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРИ ФОРМУВАННІ «РОЗУМНИХ» МІСТ

Науковий керівник: к.т.н., доц. Дуда О.М.

Harmatiuk N., Lisovyi N., Duda V.
Ternopil Ivan Puluj National Technical University

INNOVATIVE INFORMATION TECHNOLOGIES ROLE IN THE SMART CITIES FORMATION

Supervisor: Ph.D., Dr. Duda O.M.

Ключові слова: інновації, інформаційні технології, розумне місто.
Key words: innovations, information technologies, smart city.

Очікується, що до 2050 року населення світу зросте приблизно до 9,8 мільярда людей. При цьому прогнозується, що приблизно 6,7 мільярда громадян будуть жити в містах [1]. За прогнозами ООН, очікується, що до 2030 року буде понад сорок мегаполісів з населенням понад десять мільйонів жителів і вони споживатимуть понад 81% усіх світових ресурсів [2].

Тенденція розвитку сучасних міст – це перехід до більшого рівня «розумності». Вона проявляється розширенні процесів впровадження та інтеграції обширної множити інформаційних та комунікаційних технологій. Зокрема, хмарні обчислення, Інтернет речей (IoT) та інші інформаційно-технологічні інструменти, призводять до підвищення рівня «розумності» сучасних міст. Хоча опубліковано обширний перелік результатів досліджень, на даний час немає усталеного підходу до формування інформаційно-технологічного та соціо-комунікаційного концепту «розумного» міста. Тому наукові розвідки в царині «розумних» міст є актуальним напрямком сучасних досліджень.

Інновації в сучасних містах можна умовно розділити на три основні категорії: «розумні» цифрові технології, покращення умов життя та підвищення екологічних характеристик довкілля. Найпопулярніша серед цих категорій – «розумні» цифрових технології. Водночас «екологічні» характеристики довкілля досліджуються відносно рідше. Згідно досліджень [3] лише близько половини критеріїв «розумності» міст, які використовуються в емпіричних дослідженнях, стосуються потреб міських жителів, а решта є загальними технологічними характеристиками. Автори наголошують, що *«Розумне місто – це місто, в якому збалансовано економічні, екологічні та суспільні досягнення для покращення добробуту мешканців шляхом широкого впровадження інформаційних, комунікаційних та інших технологічних інструментів»* [3]. Згідно з цим визначенням, певна міська інноваційна сутність наприклад, інтелектуальні давачі або покращений моніторинг процесів, сприяє підвищенню «розумності» міста, лише якщо вона покращує якість життя (англ. Quality of Life, QoL) громадян. Концепція сталого розвитку потребує збалансованості соціальних, економічних та екологічних цілей розвитку.

Література

[1] Ritchie, H.; Roser, M. Urbanization. Our World in Data. 2018. URL: https://ourworldindata.org/urbanization.
[2] Population Division, Department of Economic and Social Affairs, United Nations. 2018 United Nations Report. URL: https://www.un.org/en/desa/around-25-billion-more-people-will-be-living-cities-2050-projects-new-un-report.
[3] Dashkevych, Oleg, and Boris A. Portnov. Criteria for Smart City Identification: A Systematic Literature Review. Sustainability 14.8 (2022): 4448.

УДК 004.9

Скалецький П. – аспірант, Лісовий Н. – ст. гр. СНс-32, Гіжовський А.
Тернопільський національний технічний університет імені Івана Пулюя

МОБІЛЬНІ ЗАСТОСУНКИ ТА РОЗВИТОК «РОЗУМНИХ» МІСТ

Науковий керівник: к.т.н., доц. Дуда О.М.

Skaletskyi P., Lisovyi N., Hizhovskyi A.
Ternopil Ivan Puluj National Technical University

MOBILE APPLICATIONS AND SMART CITIES DEVELOPMENT

Supervisor: Ph.D., Dr. Duda O.M.

Ключові слова: інформаційні технології, розумне місто, web 3.0.
Key words: information technologies, smart city, web 3.0.

На даний час практично всі інформаційні сутності взаємопов'язані через Інтернет. Завдяки широкосмуговому підключенню до Інтернет, інтерактивні послуги поширюються практично в усіх доменах людської діяльності. Водночас, завдяки портативності та обширному переліку доступних функцій стали популярними смартфони. А «розумні» лічильники енергії, «розумні» прилади та безпекові пристрої стали повсякденними. Це черговий етап процесу трансформації інноваційних інформаційно-технологічних проєктів «розумних» міст. Водночас неспроможність існуючої міської інфраструктури забезпечити прийнятні показники масштабованості, навколишнього середовища та безпеки зростають вимоги до критеріїв стійкості «розумних» міст [1]. Щоб покращити соціальну та економічну якість життя громадян, підвищити ефективність муніципальних служб та послуг потрібно будувати стійкіші «розумні» міста з використанням інноваційних інформаційних та комунікаційних технологій [2]. Відповідно, у процесі формування стійких «розумних» міст та для використання переваг та зручностей технологічного прогресу зростає потреба в розробці обширного переліку «розумних» пристроїв та застосунків.

Сучасні мобільні застосунки відіграють вагому роль в інформаційних та комунікаційних технологіях. Щоб задовольнити зростаючі потреби жителів «розумних» міст, розробникам доводиться створювати застосунки для різних мобільних платформ, зокрема, Android та iOS. Це вимагає значних затрат часу та зусиль. Щоб зменшити застрати часу, розробники почали створювати мобільні програми за допомогою стандартних веб-технологій, зокрема CSS, HTML5, JavaScript, завдяки зручному перенесенню інтерфейсів з однієї платформи на іншу. На даний час, активно розвиваються мобільні застосунки розроблені таким способом. Щоб не відставати від швидкого розвитку інформаційних та комунікаційних технологій для побудови стійких «розумних» міст та залищатись конкурентноспроможними багато розробників пришвидшують процеси розроблення мобільних застосунків. Водночас безпекові аспекти часто ігноруються. Це призводить до збільшення безпекових ризиків та можлифостей втрати конфіденційності мобільних застосунків [3].

Література

[1] Hammi, Badis, et al. "Is it really easy to detect sybil attacks in C-ITS environments: a position paper." IEEE Transactions on Intelligent Transportation Systems 23.10 (2022): 18273-18287.

[2] Chen, Yidan, and Fangfang Xu. "The optimization of ecological service function and planning control of territorial space planning for ecological protection and restoration." Sustainable Computing: Informatics and Systems 35 (2022): 100748.

[3] Kuppa, Koundinya, et al. "ConvXSS: A deep learning-based smart ICT framework against code injection attacks for HTML5 web applications in sustainable smart city infrastructure." Sustainable Cities and Society 80 (2022): 103765.

Додаток Д

Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів «АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ»

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Міжнародний університет цивільної авіації (Марокко)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей

XI Міжнародної науково-практичної конференції молодих учених та студентів
7-8 грудня 2022 року

УКРАЇНА
ТЕРНОПІЛЬ – 2022

Актуальні задачі сучасних технологій : зб. тез доповідей XI міжнар. наук.-практ. конф. Молодих учених та студентів, (Тернопіль, 7-8 грудня 2022) / М-во освіти і науки України, Терн. націон. техн. ун-т ім. І. Пулюя [та ін.]. – Тернопіль: ФОП Паляниця В. А., 2022. – 202.

ISBN 978-617-7875-49-8

Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів «АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ» – Тернопіль, 7-8 грудня 2022 року

УДК 004.9

Н.В. Лісовий [1], А.Р. Ставицька [2], А.В. Гіжовський [3]
[1]Тернопільський національний технічний університет імені Івана Пулюя, Україна
[2] Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
[3] Технічний коледж Тернопільського національного технічного університету імені Івана Пулюя

ХМАРНІ ІНФОРМАЦІЙНО-ТЕХНОЛОГІЧНІ ПЛАТФОРМИ АНАЛІТИЧНОГО ОПРАЦЮВАННЯ ДАНИХ

N.V. Lisovyi, A.R. Stavytska, A.V. Hizhovskyi

CLOUD INFORMATION TECHNOLOGY PLATFORMS FOR ANALYTICAL DATA PROCESSING

Зі стрімким розвитком інформаційних та комунікаційних технологій сучасний бізнес та підприємництво стають все більш керованим даними. Тому зростає потреба підтримки бізнесових та управлінських рішень сучасними методами аналітичного опрацювання даних. На даний час опубліковано обширний перелік результатів наукових досліджень щодо інформаційно-технологічних платформ аналізу даних у різноманітних господарських та виробничих секторах. Однак ключові характеристики, загальні висновки та рекомендації щодо використання цих платформ роззосереджені по різних дослідженнях [1]. В процесі наукових розвідок встановлено, що на даний ще не опубліковано інформації щодо спроб систематично агрегувати та синтезувати особливості використання та характеристики загальнодоступних хмарних інформаційно-технологічних платформ аналітичного опрацювання даних. Адже дослідження засобів аналітичного опрацювання даних у бізнесових та виробничих секторах є популярною темою наукових досліджень впродовж останнього періоду часу.

Щоб ефективно зрозуміти хмарні платформи аналітичного опрацювання даних, важливо знати, які функціональні набори підтримуються, які інформаційні технології прийняті до використання, які інформаційно-технологічні шаблони архітектури застосовувалися та з якими перешкодами зіткнулися користувачі та дослідники в процесі їх експлуатації [2]. Розробники хмарних інформаційно-технологічних платформ платформ стикаються з багатьма складнощами під час реалізації функцій та надання послуг в галузі аналітичного опрацювання даних. Окрім того, користувачі хмарних інформаціно-технологічних платформ при використанні інструментів аналітичного опрацювання даних стикаються з багатьма складнощами під час використання систем такого класу. Для цього доцільно провести поглиблений систематичний огляд літератури, який потрібно чітко зосередити на доменах хмарних інформаційно-технологічних платформ, зацікавлених сторонах, цілях, запроваджених та реалізованих інформаційних технологіях, властивостях даних і перешкодах щодо їх системного застосування. Виявлені особливості та складнощі використання хмарних аналітичних засобів допоможуть охарактеризувати різні інформаційно-технологічні платформи та прокладуть шлях для проведення подальших досліджень.

Література

1. Krisnawijaya, Ngakan Nyoman Kutha, et al. "Data analytics platforms for agricultural systems: A systematic literature review." Computers and Electronics in Agriculture 195 (2022): 106813.

2. . Varshney, Manasvi, Bharat Bhushan, and A. K. M. Haque. "Big Data Analytics and Data Mining for Healthcare Informatics (HCI)." Multimedia Technologies in the Internet of Things Environment, Volume 3. Springer, Singapore, 2022. 167-195.

Додаток Е

Матеріали X науково-технічної конференції «ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

7–8 грудня 2022 року

ТЕРНОПІЛЬ
2022

Матеріали X науково-технічної конфції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 7–8 грудня 2022 р.). – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2022. –162 с.

УДК 004.9

Н. Лісовий, А. Ставицька, А. Гіжовський
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)
(Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»)
(Технічний коледж Тернопільського національного технічного університету імені Івана Пулюя, Україна)

АНАЛІТИЧНЕ ОПРАЦЮВАННЯ ВЕЛИКИХ ЗА ОБСЯГОМ ДАНИХ

UDC 004.9

N. Lisovyi,| A. Stavytska, A. Hizhovskyi

LARGE DATA VOLUMES ANALYTICAL PROCESSING

В продовж останнього періоду часу експоненційно зростають обсяги цифрових даних, що створені засобами різнотипових IoT-пристроїв та інформаційно-технологічних платформ. Це зростання відбувається завдяки недавньому розвитку інформаційних та комунікаційних технологій та їх активному впровадженню у практично всі сфери людської діяльності. Зокрема, збільшенням кількості інтелектуальних пристроїв, що генерують дані з інтегрованими давачами та виконавчими механізмами, які підключені засобами глобальних загальнодоступних хмарних сервісів, збільшенням кількості користувачів Інтернету, активним запровадженням інформаційних технологій з елементами віртуальної та доповненої реальності, розвитком мобільнлого зв'язку 5G, популяризацією соціальних мереж, збільшення транзакцій електронної комерції тощо. Станом на грудень 2020 року обсяг щоденних цифрових даних у всьому світі становив 59 зетабайт. За прогнозами фахівців очікується, що в 2024 році він досягне 149 зетабайт [1], оскільки людство переходить в майбутнє, яке буде більше керуватися даними.

Інтелектуальні технології аналітичного опрацювання «Великих даних» активно використовуються при розбудові «розумних міст» з інтегрованими інформаційними системами керування дорожнім рухом, які автоматично оптимізують транспортні потоки, системами моніторингу характеристик навколишнього середовища, які оновлюють дані щодо його забруднення в режимі реального часу та прогнозують зміну якості повітря та води, раціоналізованим та автоматизованим збиранням сміття та побутових відходів, інтелектуальними системами паркування транспортних засобів у густонаселених містах [2]. Вибухове зростання обсягів накопичуваних даних сформувало обширний перелік задач, пов'язаних з оперативним збиранням даних, ефективним їх зберіганням, пошуком, обробкою та поданням через зростання характеристик обсягу, різноманітності та швидкості даних. Процеси видобування знань або виявлення корисних шаблонів у великих за обсягом наборах та колекціях даних на даний час потребують величезних обчислювальних ресурсів, розвиненої та розгалуженої інфрмаційно-технологічної інфраструктури та кваліфікованих фахівців в галузі аналітики даних та є актуальним напрямком сучасних наукових досліджень.

Література

1. Holst, A. «Amount of information globally 2010–2024.» Statista. URL: https://www. statista. com/statistics/871513/worldwide-datacreated/#:~: text= The% 20total% 20amount% 20of% 20data, ever% 2Dgrowing% 20global% 20data% 20sphere. Erişim Tarihi 27 (2020): 2020.
2. Khan, Shahbaz. «Barriers of big data analytics for smart cities development: a context of emerging economies.» International Journal of Management Science and Engineering Management 17.2 (2022): 123–131.