

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

«На правах рукопису»

УДК 004.58

До захисту допущено:

В.о. Завідувача кафедри

Михайло НОВАТАРСЬКИЙ

_____ 2025 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інженерія програмного
забезпечення комп'ютерних систем» зі спеціальності 121 “Інженерія
програмного забезпечення” на тему: “ Система автоматизації закупівель у
ресторанному бізнесі”

Виконав:

студент VI курсу, групи ІМ-42мп

Никоненко Олег Олексійович

Науковий керівник:

ас., Ph.D.,

Коренко Дмитро Володимирович

Консультант з нормоконтролю:

проф., д.т.н.,

Жабін Валерій Іванович

Рецензент:

Ст.викладач кафедри СПіСКС.,

Радченко Костянтин Олександрович

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2025 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Кафедра обчислювальної техніки

(повна назва кафедри)

Рівень вищої – освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма – «Інженерія програмного забезпечення комп'ютерних систем»

До захисту допущено: В.о.

Завідувача кафедри

Михайло НОВАТАРСЬКИЙ

(підпис) (ім'я, прізвище)

«_____» _____ 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Никоненко Олегу Олексійовичу

1. Тема проекту “ Система автоматизації закупівель у ресторанному бізнесі ”, керівник дисертації Коренко Дмитро Володимирович, Ph.D., ас., затверджені наказом по університету від « 07 » 11 _____ 2025р.
2. № 4841-с
3. Термін подання студентом проекту «15» 12 2025.
4. Вихідні дані до проекту: технічне завдання, теоретичні дані, наукові публікації.
5. Зміст пояснювальної записки: огляд існуючих веб-сервісів, розгляд схем їх реалізації, а також принципів роботи, пошук та аналіз їх основних переваг
6. та недоліків, проектування та розробка програмного забезпечення, тестування програмного модулю та аналіз отриманих результатів.
7. Перелік завдань, які потрібно розробити: огляд існуючих веб-сервісів, розгл

схем їх реалізації, а також принципів роботи, проектування та розробка програмного забезпечення, тестування програмного модулю та аналіз отриманих результатів Никоненко Олегу Олексійовичу

8. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1.			
2.			
3.			
4.			

9. Дата видачі завдання 01.09.2025 року

КАЛЕНДАРНИЙ ПЛАН

п /п	Найменування етапів дипломного проекту (роботи)	Строк виконання етапів проекту(роботи)	Примітки
1	Обговорення теми дисертації. Визначення предмету дослідження	1.09.25-7.09.25	
2	Дослідження існуючих рішень та проблем у досліджуваній області	22.09.25-28.09.25	
3	Побудова архітектурного рішення	29.09.25-05.10.25	
4	Програмна реалізація створеного алгоритму	06.10.25-10.10.25	
5	Програмування та загальне налагодження системи.	10.10.25-19.10.25	
6	Розробка стартам-проекту	14.11.25-18.11.25	
7	Оформлення магістерської дисертації.	19.11.25-24.11.25	

Студент _____
(підпис)

Олег НИКОНЕНКО

Керівник _____
(підпис)

Дмитро КОРЕНКО

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Система автоматизації закупівель у ресторанному бізнесі
студентом: Никоненком Олегом Олексійовичем

Робота складається із вступу та чотирьох розділів. Загальний обсяг роботи: 77 аркуш основного тексту, 36 ілюстрації, 21 таблиці. При підготовці використовувалася література з 40 різних джерел.

Актуальність У сучасних умовах динамічного розвитку ресторанного бізнесу та високої конкуренції питання ефективного управління ресурсами набуває особливої актуальності. Одним із ключових аспектів стабільного функціонування закладів громадського харчування є процес закупівель, який безпосередньо впливає на собівартість страв, якість обслуговування та фінансові результати підприємства.

Традиційні методи управління закупівлями, що базуються на ручному опрацюванні замовлень, веденні документації в електронних таблицях або на папері, вже не відповідають сучасним вимогам ринку. Вони створюють ризики людських помилок, затримок у постачанні, неефективного використання запасів і фінансових ресурсів. Саме тому автоматизація процесів закупівель стає необхідною умовою для підвищення ефективності та конкурентоспроможності ресторанного бізнесу.

Розробка системи автоматизації закупівель у ресторанному бізнесі є важливим кроком до цифрової трансформації галузі. Така система дозволяє оптимізувати процес взаємодії з постачальниками, контролювати залишки на складах, формувати замовлення відповідно до фактичних потреб закладу, а також зменшувати витрати часу та ресурсів на адміністративні операції.

Особливу актуальність ця тема має в умовах нестабільної економічної ситуації, коливань цін на продукти та зростання вартості логістики. Автоматизація закупівель дозволяє оперативно реагувати на зміни ринку, аналізувати дані про споживання та забезпечувати стабільність бізнес-процесів навіть за несприятливих умов.

Таким чином, створення ефективної системи автоматизації закупівель

не лише сприяє підвищенню продуктивності ресторанного бізнесу, а й забезпечує прозорість, контроль і стійкість функціонування підприємств у сучасному конкурентному середовищі.

Мета і завдання дослідження. Метою даного дослідження є розробка та впровадження веб-системи автоматизації закупівель у ресторанному бізнесі, яка дозволить підвищити ефективність управління ресурсами, оптимізувати взаємодію з постачальниками та забезпечити прозорість процесів постачання. Дослідження спрямоване на створення інноваційного інструменту, що сприятиме зниженню операційних витрат, мінімізації людського фактору та підвищенню конкурентоспроможності закладів громадського харчування.

Об'єкт дослідження - веб-сервіс для автоматизації закупівель у ресторанному бізнесі.

Предмет дослідження - процес створення та функціонування веб-сервісу для автоматизації закупівель у ресторанному бізнесі.

Методи досліджень. Для досягнення поставлених в магістерській роботі задач використано методи теорії графів, методи імітаційного моделювання, методи роботи з великими даними.

Особистий внесок здобувача. Магістерське дослідження є самостійно виконаною роботою, в якій відображено особистий авторський підхід та особисто отримані теоретичні та прикладні результати, що відносяться до розробки веб-сервісу, спрямованого на покращення якості. Формулювання мети та завдань дослідження проводилось спільно з науковим керівником.

Ключові слова: веб-система, Java, Spring, Angular, Docker, REST, RESTful, Client-server, .

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1.....	10
ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ ВЕБ-СЕРВІСІВ ДЛЯ ЗАКУПІВЕЛЬ У РЕСТОРАННОМУ БІЗНЕСІ	10
1.1. Актуальність проблеми закупівель у ресторанному бізнесі ...	10
1.2. Огляд існуючих рішень	12
1.2.1. ChefHero	13
1.2.2. MarketMan	13
1.2.3. Chefsheet	14
1.2.4. Orderly	16
1.2.5. PurchasePlus	17
1.2.6. Transgourmet / Prodega	18
1.3. Аналіз існуючих рішень	20
Висновки до розділу 1.....	22
РОЗДІЛ 2.....	23
ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ	23
2.1. Back-end.....	23
2.1.1. Spring Framework	23
2.1.2. FastAPI.....	25
2.2. Front-end	26
2.2.1. React.....	26
2.2.2. Angular.....	28
2.2.3. Vue	29
2.3. DB	31
2.3.1. PostgreSQL.....	31
2.3.2. MongoDB.....	32
2.4. Deploy	33

2.4.1. AWS	34
2.4.2. Azure	35
2.4.3. On-Premises	36
Висновок до розділу 2.....	38
РОЗДІЛ 3.....	40
РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ АВТОМАТИЗАЦІЇ ЗАКУПІВЕЛЬ У РЕСТОРАННОМУ БІЗНЕСІ	40
3.1. База даних	40
3.2. Back-end	48
3.3. Frontend	55
Висновки до розділу 3.....	63
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ НА БАЗІ ПЛАТФОРМИ	64
4.1. Опис ідеї стартапу та її бізнес-модель.....	64
4.2. Технологічний аудит ідеї.....	66
4.3. Оцінка ринкових перспектив	68
4.4. Розробка ринкової стратегії стартап-проекту	77
4.5. Розробка маркетингової програми стартап-проекту.....	80
Висновки до розділу 4.....	84
ВИСНОВКИ	86
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	89

ВСТУП

У сучасних умовах розвитку економіки ресторанний бізнес переживає суттєві трансформації, пов'язані з підвищенням конкуренції, зростанням вимог споживачів та нестабільністю поставок. Зовнішні економічні й соціальні чинники, зокрема воєнний стан, інфляційні коливання та порушення логістичних ланцюгів, істотно впливають на діяльність підприємств громадського харчування. Це зумовлює потребу у впровадженні інноваційних технологій управління, які забезпечують ефективне використання ресурсів, стабільність бізнес-процесів і гнучкість у прийнятті управлінських рішень.

Процес закупівель є важливою частиною для підтримки рентабельності бізнесу, оскільки саме він визначає якість продукції, витрати на виробництво та загальну фінансову ефективність підприємства. Традиційні методи управління закупівлями, що передбачають ручну роботу з постачальниками, створення замовлень та контроль залишків, втрачають ефективність в умовах швидкої зміни ринку. Автоматизація цих процесів стає необхідною умовою для забезпечення безперервності роботи, зменшення впливу людського фактору та оптимізації логістичних операцій.

Метою даного дослідження є розробка та впровадження веб-системи автоматизації закупівель у ресторанному бізнесі, яка дозволить підвищити ефективність управління матеріальними ресурсами, забезпечити прозорість взаємодії з постачальниками та оптимізувати витрати. Дослідження спрямоване на створення інноваційного інструменту, здатного інтегрувати процеси поставання, контролю запасів і фінансового обліку в єдине інформаційне середовище.

Враховуючи сучасні виклики та тенденції цифровізації ресторанного бізнесу, впровадження системи автоматизації закупівель має потенціал суттєво покращити ефективність функціонування підприємств громадського харчування, сприяти стабільності їх діяльності та зміцненню позицій на конкурентному ринку.

У цьому контексті дана робота присвячена аналізу, розробці та впровадженню веб-системи автоматизації закупівель у ресторанному бізнесі, яка покликана підвищити рівень управління ресурсами, зменшити витрати та забезпечити сталий розвиток галузі у складних економічних умовах.

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ ВЕБ-СЕРВІСІВ ДЛЯ ЗАКУПІВЕЛЬ У РЕСТОРАННОМУ БІЗНЕСІ

1.1. Актуальність проблеми закупівель у ресторанному бізнесі

У сучасних умовах розвитку економіки та суспільства ресторанний бізнес перебуває у стані суттєвих трансформацій, зумовлених як внутрішніми, так і зовнішніми чинниками: посилення конкуренції, зміна споживацьких уподобань, логістичні виклики, інфляційні процеси, сезонні коливання та зміни у ланцюгах постачання. Тож питання ефективного управління закупівлями набуває особливої актуальності.

Перш за все, закупівельна діяльність є одним із ключових елементів функціонування закладу громадського харчування, адже саме вона безпосередньо впливає на якість продукції, собівартість страв, резерви запасів і оперативну здатність кухні. Як зазначають експерти, “ресторанний ланцюг постачання - це не лише про вартість, але й про різноманітність, швидкопсувність продуктів, сезонні коливання і надзвичайну важливість своєчасності постачання” [1].

Додатково, традиційні методи закупівель - ручне оформлення замовлень, використання електронних таблиць чи паперових форм, відсутність централізованого контролю - дедалі частіше виявляються неадекватними до викликів сучасного ринку. Наприклад, у гостинному секторі звертають увагу, що відсутність уніфікованої стратегії закупівель і залежність від фрагментарних постачальників призводить до нестабільності, зайвих витрат і ризиків: «без структурованої системи закупівель процес може бути хаотичним, призводити до надлишкових або недостатніх запасів» [2].

Крім того, сучасні виклики логістики, глобальної нестабільності та

волатильності цін накладають додаткове навантаження на сферу закупівель. Наприклад, «нестабільні ціни, транспортні затримки, фрагментація даних між системами, недостатня видимість постачальників - усе це суттєво ускладнює управління ланцюжками постачання у ресторанах» [3].

Особливо актуальною є потреба в цифровізації закупівельних процесів. В умовах, коли «виробники, дистриб'ютори та ресторани працюють із багатьма постачальниками, мають справу із коливаннями попиту та обмеженою видимістю даних», використання сучасних ІТ-інструментів дозволяє підвищити ефективність, скоротити витрати й зменшити кількість операційних помилок. [4].

З огляду на все вищесказане, постає очевидною потреба в розробці спеціалізованої системи автоматизації закупівель, яка б враховувала специфіку ресторанного бізнесу: швидкопсувність продуктів, великі обсяги дрібних замовлень, множинність постачальників, сезонність, а також вимоги щодо якості, стандартів та прозорості. Така система сприятиме: оптимізації взаємодії з постачальниками, контролю за залишками і формування замовлень відповідно до фактичного споживання, зменшенню часу й ресурсів, витрачених на адміністративні операції, підвищенню конкурентоспроможності закладу через зменшення витрат і покращення якості обслуговування.

Отже, тема автоматизації закупівель у ресторанному бізнесі має високу науково-практичну значущість, оскільки її вирішення дозволить закладам громадського харчування адаптуватися до динамічних сучасних умов, підвищити стійкість бізнесу, забезпечити ефективне управління ресурсами та зменшити ризики, пов'язані із традиційними підходами до закупівель.

1.2. Огляд існуючих рішень

У сучасних умовах стрімкої цифровізації ресторанного бізнесу ефективне управління закупівлями стає одним із ключових чинників успіху закладів громадського харчування. Потреба у зменшенні витрат, оптимізації часу на обробку замовлень, контролі за залишками продуктів та взаємодії з постачальниками спонукає ресторани переходити від ручних або напівавтоматизованих методів до повноцінних систем автоматизації закупівель. Такі системи забезпечують прозорість усіх етапів постачання - від формування замовлення до аналітики витрат і управління постачальницькими контрактами.

На міжнародному ринку вже існує низка потужних веб-рішень, орієнтованих на оптимізацію закупівельного процесу для ресторанів, кав'ярень та готельно-ресторанних комплексів. Вони пропонують широкий спектр функціональних можливостей: від автоматичного створення замовлень і синхронізації з POS-системами до формування фінансової звітності, контролю витрат і прогнозування потреб у продуктах.

Огляд і порівняльний аналіз цих рішень дозволяє не лише визначити їхні сильні сторони, а й виявити існуючі прогалини, які можна врахувати під час розробки нової системи, адаптованої до потреб українського ресторанного бізнесу. Особлива увага в цьому контексті приділяється аспектам локалізації, вартості впровадження, інтеграційних можливостей та зручності користування.

Таким чином, проведення детального аналізу існуючих платформ є необхідним кроком для обґрунтування доцільності створення нового веб-сервісу автоматизації закупівель, який би поєднав найкращі міжнародні практики з урахуванням реалій вітчизняного ринку.

1.2.1. ChefHero

ChefHero - це комерційна платформа (marketplace + інструмент закупівель) для сегменту HoReCa (ресторани, кафе), яка дозволяє клієнтам переглядати каталог постачальників, порівнювати ціни, формувати замовлення й керувати інвойсами в одному місці. [5].



Рис. 1.1. Chefhero

Однак варто зазначити, що з часом платформа зазнала ребрендингу: у 2021 році ChefHero офіційно стала Notch, що означало зміну позиціонування - не просто як маркетплейс для ресторанів, а як повноцінна система, що охоплює весь ланцюг постачання між дистриб'юторами та ресторанами. [6].

У момент ребрендингу менеджмент зазначав, що діючий бізнес моделі, орієнтований виключно на ресторани, виявився вразливим під час пандемії, тому для стабілізації потрібно було трансформуватися, залучивши дистриб'юторів та поглибивши функціональність [7].

1.2.2. MarketMan

MarketMan - це сучасна хмарна платформа для управління закупівлями та інвентаризацією у ресторанному бізнесі. Вона допомагає ресторанам, кафе, барам та кейтеринговим компаніям оптимізувати процес постачання продуктів, контролювати витрати, уникати надлишків або дефіциту інгредієнтів і підвищувати загальну ефективність операційної діяльності. [8].



Рис. 1.2. Платформа MarketMan

MarketMan надає користувачам повний набір інструментів для автоматизації закупівельного процесу - від формування замовлень до відстеження постачання та аналізу витрат.[9] Основні можливості платформи включають:

Управління постачальниками та замовленнями

Користувачі можуть створювати, надсилати та контролювати замовлення безпосередньо з інтерфейсу системи. MarketMan дозволяє інтегруватися з численними постачальниками, автоматично отримуючи оновлені прайси та наявність товарів. Система відстежує залишки інгредієнтів на складі у режимі реального часу, допомагаючи уникати перевитрат і точно планувати наступні закупівлі. MarketMan автоматично обчислює вартість кожної страви на основі актуальних цін постачальників, дозволяючи шеф-кухарям і менеджерам ефективно керувати ціноутворенням меню. Платформа підтримує інтеграції з популярними POS-рішеннями (наприклад, Toast, Square, Lightspeed), а також із бухгалтерськими системами, такими як QuickBooks або Xero. Це забезпечує повну синхронізацію фінансових даних між фронт-офісом і бек-офісом. MarketMan пропонує детальні аналітичні звіти - про витрати, відхилення між замовленнями й поставками, списання продуктів, рівень запасів тощо. Це дозволяє приймати обґрунтовані рішення на основі даних.

1.2.3. Chefsheet

Chefsheet - це інноваційна хмарна платформа, створена для автоматизації процесів управління запасами, закупівлями та аналізом витрат

у ресторанному бізнесі. Система орієнтована на малий та середній бізнес у сфері HoReCa, пропонуючи зручний і простий інтерфейс для щоденного контролю продуктів, інгредієнтів та витрат.[10].

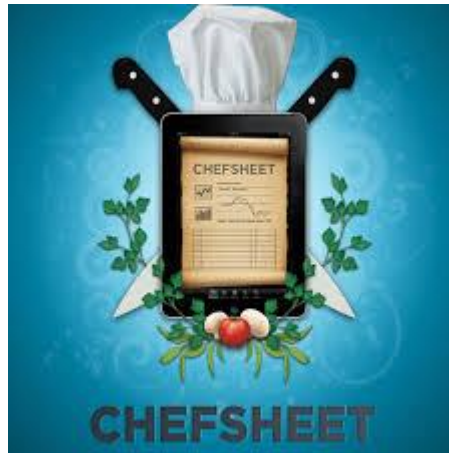


Рис. 1.3. Платформа Chefsheet

Chefsheet надає користувачам гнучкий набір інструментів для управління інвентаризацією, закупівлями та плануванням меню, що дозволяє значно скоротити час на адміністративні операції та уникнути надлишкових витрат. Основні функції системи включають:

Платформа дозволяє легко відстежувати кількість продуктів і напівфабрикатів на складі, вести історію закупівель, а також прогнозувати потреби на основі попередніх періодів. Інтерфейс оптимізований для мобільних пристроїв, що дає змогу персоналу кухні швидко оновлювати дані просто зі смартфона чи планшету. Chefsheet дає змогу формувати автоматичні замовлення постачальникам, відстежувати статус замовлень і порівнювати ціни між різними постачальниками. Це дозволяє обирати вигідніші пропозиції та зменшувати витрати на закупівлі. Система розраховує реальну собівартість кожної страви з урахуванням цін на інгредієнти та змін у постачанні. Це допомагає власникам ресторанів оперативно коригувати ціни меню відповідно до ринкових коливань. Платформа пропонує звіти в реальному часі щодо витрат, закупівель, залишків, а також втрат або списань. Аналітичні інструменти дозволяють виявляти проблемні ділянки - наприклад, перевитрати продуктів або низьку

рентабельність певних страв, також Chefsheet підтримує багатокористувацький доступ, що дозволяє менеджерам, кухарям і адміністраторам працювати в одній системі з різними правами доступу. Це забезпечує узгодженість дій та прозорість у процесі прийняття рішень.

1.2.4. Orderly

Orderly - це сучасна платформа для автоматизації процесів постачання та закупівель у ресторанному бізнесі, розроблена з метою мінімізації витрат часу й ресурсів на управління ланцюгами поставок. Вона орієнтована на ресторани, кав'ярні, готелі та інші заклади громадського харчування, які прагнуть оптимізувати процес замовлення продуктів, інвентаризацію та взаємодію з постачальниками[11].



Рис. 1.4. Платформа ¹Orderly

Orderly створена як “розумна” система управління закупівлями, що дозволяє автоматично відстежувати потреби ресторану в продуктах, порівнювати ціни від різних постачальників і формувати замовлення в один клік. Її головна мета - усунути людський фактор із рутинних процесів і забезпечити прозорість у витратах.[12].

Платформа об'єднує три ключові напрями управління, а саме контроль витрат і фінансових потоків, автоматизація замовлень, аналітика ефективності постачань. Orderly аналізує історію покупок, поточні залишки на складі та прогнозовані потреби, створюючи оптимальні списки закупівель. Це дозволяє менеджерам ресторанів економити до 30% часу на адміністративних завданнях. Система автоматично зчитує чеки, інвойси та

накладні (зокрема через фото або PDF-файли), переводячи їх у цифрову форму. Це усуває необхідність ручного введення даних і знижує ризик помилок у фінансовому обліку. Завдяки базі постачальників користувач може швидко визначити, у кого придбати інгредієнти за кращою ціною. Orderly підтримує динамічне оновлення прайсів, що особливо важливо в умовах інфляції або сезонних коливань. Платформа має мобільну версію, яка дозволяє персоналу створювати та відправляти замовлення безпосередньо з кухні. Це підвищує зручність роботи та прискорює процес закупівель. Orderly легко інтегрується з популярними системами обліку (QuickBooks, Xero, Toast, Square POS), що дозволяє забезпечити повну синхронізацію між закупівлями, бухгалтерією та POS-системами ресторану.

1.2.5. PurchasePlus

PurchasePlus - це потужна хмарна платформа для управління закупівлями, фінансами та постачанням у ресторанному та готельному бізнесі. Вона розроблена для оптимізації процесів взаємодії між ресторанами, постачальниками та адміністративними підрозділами, забезпечуючи повний контроль за всіма етапами закупівель - від створення замовлення до оплати рахунків.[13].



Рис. 1.5. Платформа PurchasePlus

PurchasePlus - це інтегрована система автоматизації закупівель, яка дозволяє бізнесам у сфері HoReCa підвищити ефективність управління ресурсами, зменшити витрати та забезпечити прозорість фінансових операцій. Платформа широко використовується у ресторанах, готелях, кейтерингових компаніях і мережах громадського харчування в Австралії,

Новій Зеландії, Європі та США.

Її головна мета - зробити процес закупівель максимально автоматизованим, аналітичним і прозорим, щоб менеджери могли ухвалювати обґрунтовані рішення на основі реальних даних. Система дозволяє створювати, редагувати та відстежувати замовлення в онлайн-режимі. Завдяки інтеграції з постачальниками всі ціни оновлюються автоматично, що усуває необхідність ручного введення даних. PurchasePlus надає користувачам можливість порівнювати пропозиції від кількох постачальників за критеріями ціни, якості, строків доставки тощо. Це сприяє ухваленню вигідніших комерційних рішень. У системі реалізований модуль управління рахунками-фактурами, бюджетами та платежами.

Автоматизований облік витрат дозволяє уникнути помилок, пов'язаних із ручними операціями, та спрощує підготовку фінансової звітності.

PurchasePlus надає повний огляд запасів у реальному часі, допомагаючи уникнути як надлишкових закупівель, так і нестачі продуктів. Функція прогнозування попиту дозволяє розраховувати оптимальний рівень запасів відповідно до динаміки продажів. Платформа сумісна з популярними рішеннями для бухгалтерського обліку (Xero, MYOB, QuickBooks), POS-системами (Micros, Square, Lightspeed) і ERP-рішеннями. Це забезпечує єдину екосистему управління для бізнесу будь-якого масштабу.

1.2.6. Transgourmet / Prodega

Transgourmet / Prodega - це одна з провідних європейських систем постачання та закупівель для ресторанного і готельного бізнесу, яка поєднує в собі елементи B2B-комерції, логістики, аналітики та автоматизації процесів закупівель. Компанія є частиною групи Transgourmet Group, що входить до складу COOP Schweiz [14].



Рис. 1.6. Платформа TransGourmet / Prodega

Transgourmet / Prodega - це повноцінна екосистема для професійного постачання HoReCa-сектору, що забезпечує ресторани, готелі, кейтерингові компанії, лікарні, школи та інші установи продуктами харчування, напоями, інвентарем і господарськими товарами. [15].

Платформа надає можливість клієнтам автоматизувати процеси закупівель, керувати асортиментом, планувати витрати та оптимізувати ланцюг постачань.

Відмінною рисою системи є її поєднання онлайн-замовлення, фізичних cash&carry складів і розвиненої логістичної мережі. Це дозволяє клієнтам обирати зручний формат співпраці - від самостійного відвідування складу до повністю дистанційних поставок через електронний портал. [16]. Система дозволяє користувачам формувати та відправляти замовлення через персоналізований онлайн-кабінет. Кожен клієнт отримує доступ до каталогу з понад 25 000 найменувань товарів, які постійно оновлюються. Замовлення інтегруються з внутрішніми ERP- або POS-системами ресторану, що дозволяє синхронізувати облік закупівель і витрат. Це зменшує адміністративне навантаження на персонал і знижує ризики помилок у плануванні запасів. Transgourmet використовує гнучку систему ціноутворення: кожен клієнт отримує індивідуальні ціни, знижки та акційні пропозиції відповідно до обсягів закупівель і частоти замовлень. Платформа дозволяє відстежувати статус замовлень у реальному часі, планувати графік поставок і керувати кількома точками доставки. Розвинена логістична

мережа охоплює всю територію Швейцарії, Австрії, Німеччини та Франції. Компанія активно впроваджує екологічні стандарти: скорочення харчових відходів, оптимізація маршрутів доставки, використання електровантажівок і постачання сертифікованої продукції (Fairtrade, Bio Suisse).

1.3. Аналіз існуючих рішень

У цьому розділі виконується порівняльний аналіз розглянутих вище веб-сервісів та програмних рішень, що використовуються у сфері ресторанного бізнесу для управління закупівлями, запасами та постачанням. Даний аналіз є важливим етапом у контексті визначення ключових проблем галузі, а також виявлення можливостей для створення нової, більш ефективної системи автоматизації закупівель, адаптованої до сучасних потреб ринку громадського харчування.

Метою аналізу є визначення переваг і недоліків кожного із досліджуваних рішень, а також окреслення їхньої ролі у підвищенні ефективності операційних процесів ресторанів. Особлива увага приділяється таким аспектам, як:

- функціональність системи (управління замовленнями, складський облік, фінансовий контроль тощо);

- зручність користувацького інтерфейсу;

- інтеграції з POS-системами, ERP-рішеннями та постачальниками;

- цінова політика та сегмент ринку;

- можливість адаптації до локальних умов і специфіки національного ринку.

У цьому контексті будуть детально розглянуті рішення для ресторанного бізнесу, зокрема ChefHero, MarketMan, Chefsheet, Orderly, PurchasePlus та Transgourmet / Prodega.

Кожен із цих сервісів займає певну нішу на ринку та вирішує схожі проблеми, але з різним рівнем автоматизації, інтеграцій та користувацьких можливостей:

ChefHero - це платформа для онлайн-замовлення продуктів безпосередньо у постачальників, що дозволяє ресторанам централізувати закупівельні процеси та контролювати витрати.

MarketMan - комплексна система управління інвентарем і закупівлями, яка надає аналітику витрат, планування меню та фінансовий контроль.

Chefsheet - орієнтована на малий та середній бізнес платформа, що спрощує облік запасів і формування замовлень без потреби у складних ERP-інтеграціях.

Orderly - мобільний застосунок, який автоматизує облік витрат і дозволяє легко порівнювати ціни від постачальників, роблячи процес закупівель прозорішим.

PurchasePlus - хмарна система, яка поєднує управління закупівлями, рахунками та взаємодію з постачальниками, орієнтована на середній та великий бізнес.

Transgourmet / Prodega - крупний європейський постачальник продуктів харчування для ресторанного сектору, що надає власний онлайн-портал для оформлення замовлень, відстеження поставок і керування рахунками.

Аналіз зазначених рішень дозволить виявити не лише їхні функціональні сильні сторони, а й ті аспекти, де існують обмеження - наприклад, недостатня локалізація для українського ринку, складність інтеграції з місцевими POS-системами або висока вартість впровадження.

Таким чином, проведений аналіз є ключовим для формування концепції власної системи автоматизації закупівель, яка б поєднувала зручність, доступність і функціональність, орієнтовану на реальні потреби ресторанів малого та середнього бізнесу. Отримані результати стануть основою для розробки архітектури майбутнього веб-сервісу, спрямованого на оптимізацію закупівельних процесів у ресторанному секторі.

Висновки до розділу 1

У контексті сучасних викликів ресторанного бізнесу, таких як зростання конкуренції, коливання цін на продукти, дефіцит кваліфікованого персоналу та необхідність ефективного управління витратами, створення системи автоматизації закупівель у ресторанному бізнесі стає надзвичайно актуальним завданням. Важливість цього завдання полягає у тому, щоб забезпечити підприємства громадського харчування сучасними інструментами для оптимізації процесів постачання, підвищення прозорості фінансових операцій, а також зменшення впливу людського фактора при формуванні замовлень та контролі запасів.

Сфера закупівель у ресторанному бізнесі є однією з найскладніших, оскільки потребує щоденного контролю великої кількості позицій, від продуктів харчування з обмеженим терміном придатності до технічних та господарських матеріалів. У зв'язку з цим, традиційні методи управління закупівлями - за допомогою Excel-таблиць або ручного контролю - вже не відповідають вимогам сучасного ринку.

Огляд існуючих цифрових рішень, таких як ChefHero, MarketMan, Chefsheet, Orderly, PurchasePlus, Transgourmet/Prodega та інших, показує, що вже існують інструменти, які частково або повністю автоматизують процеси закупівель. Ці сервіси дозволяють створювати замовлення онлайн, вести аналітику витрат, контролювати рівень запасів, інтегруватися з POS- та ERP-системами, що суттєво підвищує ефективність управління рестораном.

Проте, незважаючи на наявність таких систем, багато з них орієнтовані переважно на великі ресторанні мережі або міжнародні компанії, а не на малі та середні підприємства, які складають основну частину українського ресторанного ринку. Крім того, більшість доступних іноземних платформ мають високу вартість ліцензії, відсутність локалізації українською мовою та обмежену інтеграцію з місцевими постачальниками.

РОЗДІЛ 2

ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ

2.1. Back-end

У рамках проєкту планується використовувати клієнт-серверну архітектуру, де взаємодія здійснюється між фронтендом (клієнтом) та серверною частиною (бекенд), яка обробляє бізнес-логіку й працює з базою даних. У такій архітектурі клієнт надсилає HTTP/HTTPS запити до сервера, сервер обробляє їх, виконує необхідні операції з даними й повертає відповідь клієнту - зазвичай у форматі JSON. Це базова модель веб-додатків і API, які використовуються практично у всіх сучасних системах[17].

Клієнт-серверна архітектура дозволяє ефективно розділити відповідальність. Front-end (клієнт) відповідає за показ інтерфейсу користувачеві, валідацію введених даних та відправку запитів до API. Back-end (сервер) реалізує логіку застосунку, маршрутизацію запитів, обробку даних, взаємодію з базою даних, авторизацію/аутентифікацію та інші серверні задачі.

2.1.1. Spring Framework

Spring Framework - це один із найпопулярніших фреймворків для розробки корпоративних застосунків на мові Java. Його головна мета - спростити процес створення, тестування та розгортання серверних додатків, забезпечуючи високу гнучкість та масштабованість. Завдяки модульній структурі Spring надає розробникам можливість використовувати лише необхідні компоненти, що робить його універсальним інструментом для побудови бекенд-систем будь-якої складності [18].



Spring Framework Runtime

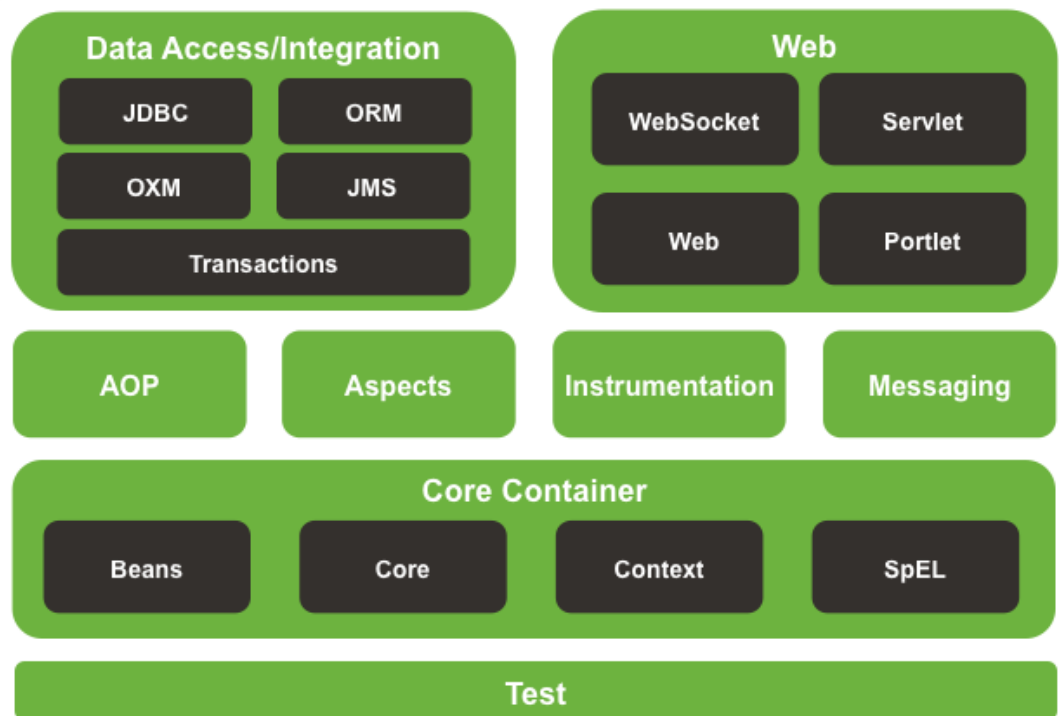


Рис. 2.1. Модулі Spring Framework

Ключовим елементом архітектури Spring є Inversion of Control (IoC) - принцип, який передбачає передачу управління створенням і життєвим циклом об'єктів спеціальному контейнеру. Це дозволяє досягти слабкого зв'язку між компонентами, що підвищує гнучкість системи та спрощує тестування. Іншим важливим механізмом є Aspect-Oriented Programming (AOP), який допомагає відокремити технічну логіку (наприклад, логування, безпеку, транзакції) від бізнес-логіки [19].

Одним із найпоширеніших модулів є Spring Boot, який спрощує створення автономних веб-застосунків завдяки автоматичній конфігурації та вбудованому веб-серверу (Tomcat, Jetty або Undertow). Це дозволяє швидко запускати RESTful-сервіси без складних налаштувань. Крім того, Spring Boot забезпечує інтеграцію з базами даних (через Spring Data JPA), системами безпеки (Spring Security) та мікросервісною архітектурою (Spring Cloud), що

робить його ідеальним вибором для розробки сучасних веб-платформ.

Використання Spring у межах даного проєкту є доцільним, оскільки фреймворк поєднує високу продуктивність, надійність і підтримку великої кількості технологій. Завдяки цьому Back-end частина системи зможе ефективно обробляти запити, забезпечувати безпеку даних і масштабуватися відповідно до зростання навантаження.

2.1.2. FastAPI

FastAPI - це сучасний, високопродуктивний фреймворк для створення API та бекенд-застосунків мовою програмування Python. Його було розроблено Себастьяном Раміресом у 2018 році з метою забезпечення максимальної швидкості розробки та виконання, використовуючи стандарт ASGI (Asynchronous Server Gateway Interface) для асинхронної обробки запитів [20].



Рис. 2.2. FastAPI Framework

Однією з ключових особливостей FastAPI є автоматична генерація документації API. Завдяки інтеграції зі стандартами OpenAPI та Swagger UI, розробники отримують зручний інтерфейс для тестування та перегляду ендпоінтів без додаткових зусиль. Це значно полегшує підтримку та розвиток великих проєктів.

FastAPI також базується на типізації Python (type hints), що забезпечує

виявлення помилок ще на етапі розробки. Система автоматично перевіряє правильність типів даних у запитах і відповідях, що підвищує надійність та стабільність коду. Вбудована підтримка Pydantic - бібліотеки для валідації даних - дозволяє легко створювати моделі запитів і відповідей, які гарантують узгодженість структури між клієнтом і сервером [21].

Ще однією важливою перевагою FastAPI є висока продуктивність, яка наближається до таких фреймворків, як Node.js та Go. Завдяки асинхронній обробці запитів (через `async/await`) FastAPI здатний обслуговувати тисячі одночасних підключень, що робить його придатним для створення масштабованих мікросервісів і хмарних рішень [22]

FastAPI чудово підходить для реалізації RESTful-сервісів, інтеграції з базами даних (через SQLAlchemy або Tortoise ORM), а також для побудови мікросервісної архітектури. У межах цього проєкту його можна використовувати як альтернативу Spring Framework для Back-end частини системи, якщо пріоритетом є швидкість розробки, простота коду та зручна інтеграція з Python-екосистемою.

2.2. Front-end

2.2.1. React

React - це сучасна JavaScript-бібліотека для побудови інтерфейсів користувача (UI), створена компанією Meta (Facebook) у 2013 році. Вона широко використовується для розробки односторінкових вебзастосунків (SPA) та динамічних інтерфейсів, що забезпечують високу швидкодію та інтерактивність. [23]

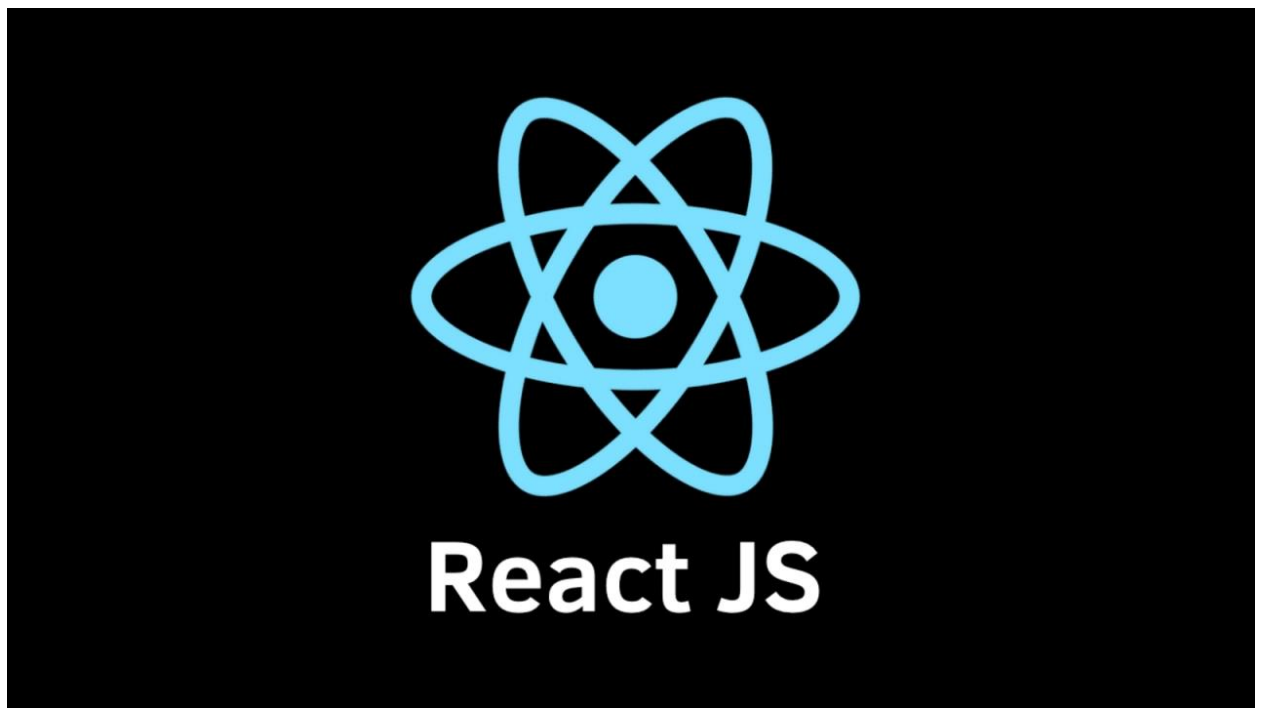


Рис. 2.3. React Framework

Основна ідея React полягає у використанні компонентного підходу - розбитті інтерфейсу на незалежні частини (компоненти), які можуть бути повторно використані в різних частинах програми. Такий підхід полегшує підтримку, розширення та тестування коду, а також сприяє модульності системи. Компоненти можуть мати власний стан (**state**) і властивості (**props**), що дозволяє створювати гнучкі та динамічні інтерфейси.

React використовує концепцію віртуального DOM (Virtual DOM), яка значно підвищує продуктивність. Замість того, щоб оновлювати реальний DOM щоразу, коли змінюються дані, React оновлює лише змінені частини віртуального представлення сторінки. Це забезпечує швидшу реакцію користувацького інтерфейсу та зменшує навантаження на браузер [24].

Ще однією сильною стороною React є JSX (JavaScript XML) - синтаксичне розширення, яке дозволяє розробникам писати HTML-подібний код безпосередньо в JavaScript. Це спрощує розробку та робить код більш читабельним. React також підтримує роботу з численними бібліотеками для керування станом, такими як **Redux** або **Context API**, що допомагає ефективно працювати з великими обсягами даних і складною логікою

користувацької взаємодії [25].

React активно використовується в корпоративних системах, стартапах та веб-платформах завдяки своїй гнучкості, масштабованості та широкій спільноті. Для проєкту автоматизації закупівель у ресторанному бізнесі React є доцільним вибором, оскільки дозволяє створювати інтуїтивно зрозумілий та швидкий інтерфейс користувача, який може легко взаємодіяти з бекендом через REST API.

2.2.2. Angular

Angular - це потужний фреймворк для створення вебзастосунків, розроблений компанією Google. Перша версія (AngularJS) з'явилася у 2010 році, але сучасна версія Angular (починаючи з Angular 2) повністю переписана на TypeScript і орієнтована на розробку динамічних односторінкових застосунків (SPA) високого рівня складності [26].



Рис. 2.4. Angular Framework

Angular забезпечує структурований підхід до розробки, який базується на архітектурі Model-View-Controller (MVC). Завдяки цьому розробники можуть чітко розділяти логіку, представлення та дані, що робить код більш зрозумілим і зручним для масштабування. Основними складовими Angular є модулі, компоненти, сервіси та директиви, які спільно утворюють гнучку систему управління інтерфейсом і логікою програми.

Однією з ключових переваг Angular є двостороннє зв'язування даних (two-way data binding), яке дозволяє автоматично синхронізувати зміни між моделлю даних і відображенням інтерфейсу. Це спрощує роботу з динамічними формами, списками, таблицями та іншими елементами користувацького інтерфейсу [27].

Angular також має вбудовану підтримку Dependency Injection (DI) - механізму, який дозволяє ефективно управляти залежностями між компонентами, що покращує тестування, повторне використання коду та загальну архітектурну гнучкість застосунку. Крім того, Angular інтегровано з потужними інструментами для розробки, такими як Angular CLI (Command Line Interface), який автоматизує створення, налаштування та тестування проєктів [28].

Завдяки суворій типізації, підтримці реактивного програмування через RxJS, а також високій продуктивності рендерингу, Angular є ідеальним вибором для великомасштабних корпоративних систем. Для системи автоматизації закупівель у ресторанному бізнесі цей фреймворк може бути використаний для створення адміністративних панелей, інтерактивних інтерфейсів для замовлення товарів, а також для ефективної взаємодії з бекендом через REST API або GraphQL.

2.2.3. Vue

Vue.js - це прогресивний JavaScript-фреймворк для створення користувацьких інтерфейсів та односторінкових вебзастосунків (SPA). Його розробив Еван Ю (Evan You) у 2014 році, і відтоді Vue.js став одним із найпопулярніших інструментів фронтенд-розробки завдяки своїй простоті, гнучкості та легкій інтеграції з іншими технологіями [29].



Рис. 2.5. Vue.js Framework

На відміну від великих фреймворків, таких як Angular, Vue має модульну структуру, що дозволяє поступово впроваджувати його у вже наявні проєкти. Розробники можуть використовувати Vue як просту бібліотеку для відображення даних або як повноцінний фреймворк для побудови складних SPA-застосунків. Такий підхід робить Vue особливо привабливим для малих і середніх проєктів, які поступово розвиваються в масштабні системи.

Однією з головних переваг Vue.js є реактивність - система автоматичного оновлення інтерфейсу при зміні даних. Це досягається завдяки механізму data binding, який забезпечує синхронізацію між моделлю даних і візуальним відображенням. Крім того, Vue використовує віртуальний DOM (Virtual DOM), що забезпечує високу продуктивність і швидкість рендерингу сторінок [30].

Фреймворк має зручну структуру на основі компонентів, які містять у собі HTML, CSS і JavaScript, що спрощує розробку, тестування й повторне використання елементів інтерфейсу. Vue також надає офіційні бібліотеки для роботи з маршрутизацією (Vue Router) і керування станом (Vuex), що дозволяє будувати масштабовані вебсистеми з чіткою архітектурою. [31].

Завдяки своїй легкості, інтуїтивності та широкій спільноті Vue.js часто обирають для створення високодинамічних вебінтерфейсів, панелей

адміністрування, каталогів товарів і CRM-систем. У межах розробки системи автоматизації закупівель у ресторанному бізнесі Vue.js може бути використаний для реалізації фронтенду, який забезпечить зручну взаємодію користувача з базою даних, динамічне оновлення інформації про товари, а також швидкий доступ до аналітики та замовлень.

2.3. DB

2.3.1. PostgreSQL

PostgreSQL - це потужна об'єктно-реляційна система керування базами даних (СУБД) з відкритим вихідним кодом. Вона підтримує стандарт SQL і водночас надає розширені можливості, які виходять за межі традиційних реляційних моделей. Розробка PostgreSQL розпочалася ще в 1986 році в Каліфорнійському університеті Берклі, і з того часу система здобула репутацію однієї з найнадійніших СУБД у світі [32].



Рис. 2.6. PostgreSQL

Однією з ключових переваг PostgreSQL є її підтримка транзакційності відповідно до принципів ACID (Atomicity, Consistency, Isolation, Durability), що гарантує надійність збереження даних навіть у випадку збоїв. Крім того, PostgreSQL підтримує паралельні запити, індексацію, тригери, збережені

процедури та оголошення користувацьких типів даних, що робить її універсальним рішенням для складних корпоративних застосунків.

PostgreSQL відома своєю розширюваністю: користувач може створювати власні функції на різних мовах програмування (наприклад, Python, Java, C), а також підключати додаткові модулі, як-от PostGIS для роботи з географічними даними. Це робить PostgreSQL придатною для аналітичних систем, ERP, CRM та фінансових платформ [33].

Для системи автоматизації закупівель у ресторанному бізнесі PostgreSQL є доцільним вибором, оскільки забезпечує високу надійність, масштабованість і захист даних. Завдяки підтримці складних зв'язків між таблицями можна ефективно зберігати інформацію про постачальників, замовлення, інвентар та транзакції, гарантуючи цілісність усіх бізнес-процесів.

2.3.2. MongoDB

MongoDB - це нереляційна документоорієнтована система керування базами даних (NoSQL), створена у 2009 році компанією MongoDB Inc. Вона орієнтована на роботу з великими обсягами неструктурованих або напівструктурованих даних, що робить її одним із найпопулярніших рішень у сучасних вебзастосунках [34]



Рис. 2.7. MongoDB

Основною особливістю MongoDB є використання формату BSON (Binary JSON) для зберігання даних. Це забезпечує гнучкість структури - на

відміну від реляційних баз, у MongoDB не потрібно заздалегідь визначати схему таблиці. Кожен запис зберігається у вигляді документа з полями, що можуть відрізнятися за структурою від інших документів у колекції. Такий підхід спрощує розробку та дозволяє швидко змінювати модель даних без складних міграцій [35].

MongoDB підтримує горизонтальне масштабування через шардінг, а також високу доступність за допомогою механізму реплікації (Replica Set). Завдяки цьому база даних здатна ефективно працювати з великими навантаженнями та обробляти тисячі запитів у реальному часі.

MongoDB часто використовується у системах електронної комерції, аналітичних платформах, реальному моніторингу даних та хмарних застосунках. У межах системи автоматизації закупівель у ресторанному бізнесі MongoDB може використовуватись для зберігання нормативних документів, каталогів постачальників, товарних карток, історії замовлень або для роботи з аналітикою в реальному часі.

2.4. Deploy

У процесі розробки сучасних веб-застосунків важливу роль відіграє етап деплою (розгортання) - перенесення програмного забезпечення з середовища розробки у виробниче (production). Одним із ключових підходів до спрощення розгортання є контейнеризація.

Контейнеризація - це технологія, що дозволяє ізолювати застосунок та всі його залежності у спеціальний контейнер, який можна легко переносити між різними середовищами. На відміну від віртуальних машин, контейнери не потребують власної операційної системи - вони використовують ядро ОС хост-машини, що робить їх легкими та швидкими.

Найпоширенішим інструментом для контейнеризації є Docker - платформа з відкритим кодом, створена у 2013 році. Docker дозволяє розробникам “упаковувати” застосунок разом із усіма необхідними

бібліотеками, конфігураціями та системними залежностями у стандартизований контейнер. Це гарантує стабільну роботу системи на будь-якому сервері, незалежно від середовища [36].



Рис. 2.8. Docker

Використання Docker має такі переваги:

Портативність - контейнер можна запустити на будь-якому сервері або хмарній платформі.

Швидке розгортання - запуск контейнера займає секунди, що скорочує час CI/CD-процесів.

Ізоляція середовищ - кожен контейнер працює незалежно, що запобігає конфліктам між залежностями.

Масштабованість - Docker легко інтегрується з оркестраторами, такими як Kubernetes, що дозволяє автоматично розподіляти навантаження між контейнерами.

2.4.1. AWS

Amazon Web Services (AWS) - це провідна хмарна платформа, створена компанією Amazon у 2006 році. Вона надає широкий спектр сервісів для розробки, розгортання та масштабування застосунків у хмарі, включаючи обчислювальні ресурси (EC2), бази даних (RDS, DynamoDB), сховище даних (S3) та інструменти для CI/CD [37].

AWS активно підтримує контейнеризацію через сервіси Amazon Elastic

Container Service (ECS), Elastic Kubernetes Service (EKS) та AWS Fargate. Вони дозволяють розгортати Docker-контейнери без необхідності вручну керувати серверами. AWS забезпечує автоматичне масштабування, балансування навантаження та моніторинг роботи контейнерів.



Рис. 2.8. AWS

Основні переваги AWS:

Глобальна інфраструктура - дата-центри розташовані по всьому світу, що зменшує затримку при доступі до даних.

Інтеграція з CI/CD-пайплайнами - AWS підтримує інструменти на кшталт AWS CodePipeline, CodeBuild та CodeDeploy, що спрощують автоматизацію розгортання.

Висока безпека - багаторівневе шифрування даних і система доступу IAM.

Для проєкту автоматизації закупівель AWS є ефективним рішенням, оскільки дозволяє швидко масштабувати систему залежно від навантаження (наприклад, у пікові години замовлень), а також зберігати дані в надійному середовищі з високим рівнем доступності (uptime понад 99.99%).

2.4.2. Azure

Microsoft Azure - це хмарна платформа, розроблена компанією Microsoft, що надає понад 200 сервісів для обчислень, аналітики, баз даних і DevOps. Azure дозволяє розробникам створювати, тестувати та розгортати застосунки у глобально розподіленому середовищі [38].



Рис. 2.9. Microsoft Azure

Azure має потужну підтримку Docker і Kubernetes через сервіси Azure Container Instances (ACI) та Azure Kubernetes Service (AKS). Вони дозволяють автоматизувати деплой контейнеризованих застосунків і легко інтегрувати їх із системами CI/CD, такими як GitHub Actions або Azure DevOps Pipelines.

Основні переваги Azure:

Глибока інтеграція з екосистемою Microsoft - зручна робота з Visual Studio, Active Directory, Office 365 тощо.

Гнучке керування контейнерами - можливість швидкого масштабування, оновлення та відновлення контейнерів.

Безпека і відповідність стандартам - Azure відповідає ISO, GDPR, HIPAA та іншим міжнародним нормам безпеки.

Для системи автоматизації закупівель Azure є оптимальним вибором, якщо компанія вже використовує продукти Microsoft. Azure дозволяє розміщувати бекенд, базу даних та аналітику у єдиному середовищі, забезпечуючи безперебійну роботу веб-сервісу для ресторанного бізнесу.

2.4.3. On-Premises

On-Premises (або локальне розгортання) означає встановлення та експлуатацію програмного забезпечення на власних серверах підприємства, без використання зовнішніх хмарних сервісів. Такий підхід забезпечує повний контроль над інфраструктурою, даними та безпекою, але вимагає більше ресурсів для обслуговування. [39].

При використанні Docker контейнери можуть розгортатися безпосередньо на локальних серверах або в приватних дата-центрах. Це дозволяє отримати переваги контейнеризації - швидке оновлення застосунку, просте масштабування та ізоляцію середовищ - без передачі даних у зовнішні хмари. Для оркестрації контейнерів часто застосовуються інструменти Kubernetes, Docker Swarm або OpenShift [40].

Переваги On-Premises підходу:

Повний контроль над даними - важливо для компаній, які працюють із конфіденційною або фінансовою інформацією.

Гнучке налаштування інфраструктури - система адаптується під специфіку бізнесу.

Відсутність абонентської плати - на відміну від хмарних сервісів, немає регулярних витрат за використання ресурсів.

У випадку системи автоматизації закупівель у ресторанному бізнесі On-Premises деплой може бути виправданим для мереж ресторанів, що мають власні IT-відділи та бажають забезпечити максимальний контроль над бізнес-даними.

Висновок до розділу 2

У другому розділі було детально розглянуто архітектурні, технологічні та інфраструктурні аспекти розробки системи автоматизації закупівель у ресторанному бізнесі. Основну увагу приділено аналізу клієнт-серверної архітектури, де взаємодія між користувачем і системою реалізується через веб-інтерфейс (Front-end) та серверну частину (Back-end), що обробляє запити й забезпечує доступ до бази даних.

Було розглянуто сучасні технологічні підходи, які забезпечують ефективну, масштабовану та безпечну роботу веб-сервісу. Зокрема, проаналізовано кілька можливих рішень для кожного з основних компонентів системи - бекенду, фронтенду, бази даних та середовища розгортання.

За результатами аналізу технологій було обрано оптимальний стек для реалізації проєкту. Back-end: фреймворк Spring - завдяки своїй модульності, високій продуктивності та підтримці інтеграцій із сучасними базами даних і хмарними сервісами. Spring забезпечує стабільну роботу бізнес-логіки та безпечну обробку запитів користувачів. Front-end: фреймворк Angular, який надає широкі можливості для створення динамічного інтерфейсу користувача, підтримує двосторонню прив'язку даних (data binding) та має потужну систему компонентів. Це робить його ефективним інструментом для побудови адаптивного й інтуїтивно зрозумілого інтерфейсу.

База даних (DB): PostgreSQL, що поєднує потужність реляційної моделі з гнучкістю сучасних механізмів збереження даних. PostgreSQL забезпечує високу надійність, транзакційність і масштабованість, що особливо важливо для систем із великими обсягами даних.

Deploy: для розгортання системи було обрано локальне розгортання системи. Окремі компоненти системи можуть бути підняті за допомогою Docker.

Таким чином, у межах цього розділу було сформовано технічну основу системи, яка відповідає сучасним вимогам до веб-рішень: надійність, безпека, масштабованість та зручність розгортання. Обраний стек технологій забезпечує узгоджену взаємодію всіх компонентів системи та створює стабільну платформу для подальшої реалізації й впровадження програмного продукту в ресторанному бізнесі.

РОЗДІЛ 3

РОЗРОБКА ВЕБ-ПЛАТФОРМИ ДЛЯ АВТОМАТИЗАЦІЇ ЗАКУПІВЕЛЬ У РЕСТОРАННОМУ БІЗНЕСІ

3.1. База даних

Основні таблиці бази даних представляють ключові сутності, що функціонують у межах B2B платформи торгівлі продуктами харчування. До них відносяться:

Ролі (Roles): Встановлюють рівні доступу користувачів до функціоналу системи.

Користувачі (Users): Зберігають дані про зареєстрованих учасників платформи, включаючи персональну інформацію та контактні реквізити.

Категорії (Categories): Містять класифікацію інгредієнтів за типами продукції.

Інгредієнти (Ingredients): Описують товарні позиції, доступні для торгівлі на платформі.

Постачальники (Suppliers): Зберігають інформацію про компанії-постачальники та їхні реквізити.

Пропозиції (Offers): Фіксують комерційні пропозиції від постачальників щодо конкретних інгредієнтів.

Замовлення (Orders): Реєструють транзакції між покупцями та постачальниками.

Елементи замовлення (Order_Items): Деталізують склад кожного замовлення з конкретними товарними позиціями.

Відгуки (Reviews): Зберігають оцінки та коментарі покупців про постачальників.

Таблиця Roles містить інформацію про ролі користувачів у системі.

Основні поля включають:

id (Primary Key): Унікальний ідентифікатор ролі, генерується автоматично.

name: Назва ролі (ADMIN, SUPPLIER, BUYER), обмежена 50 символами, має унікальне значення.

description: Текстовий опис призначення та повноважень ролі.

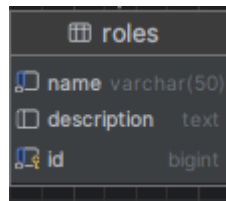


Рис. 3.1-Таблиця Role

Таблиця містить три основні ролі: адміністратор для управління системою, постачальник для створення пропозицій та покупець для здійснення замовлень.

Таблиця Users зберігає персональні дані користувачів платформи.

Основні поля включають:

id (Primary Key): Унікальний ідентифікатор користувача.

email: Електронна адреса для входу в систему, має унікальне значення.

password_hash: Хешований пароль для забезпечення безпеки авторизації.

first_name, last_name: Ім'я та прізвище користувача.

phone: Контактний телефонний номер.

is_active: Булеве поле, що визначає активність облікового запису (за замовчуванням true).

role_id (Foreign Key): Посилання на таблицю Roles, визначає роль користувача.

created_at: Часова мітка створення облікового запису.

updated_at: Часова мітка останнього оновлення даних.

users	
email	varchar(255)
password_hash	varchar(255)
first_name	varchar(255)
last_name	varchar(255)
phone	varchar(50)
is_active	boolean
role_id	bigint
created_at	timestamp
updated_at	timestamp
id	bigint

Рис. 3.2-Таблиця Users

Для забезпечення цілісності даних встановлено зовнішній ключ на таблицю ролей з обмеженням ON DELETE RESTRICT, що запобігає видаленню ролі, яка використовується.

Таблиця Categories організовує класифікацію продуктів харчування. Основні поля включають:

id (Primary Key): Унікальний ідентифікатор категорії.

name: Назва категорії продуктів.

description: Детальний опис типу продукції, що входить до категорії.

categories	
name	varchar(255)
description	text
id	bigint

Рис. 3.3-Таблиця Categories

Категорії дозволяють структурувати асортимент та полегшують пошук необхідних інгредієнтів для покупців.

Таблиця Ingredients містить відомості про товарні позиції. Основні поля включають:

id (Primary Key): Унікальний ідентифікатор інгредієнта.

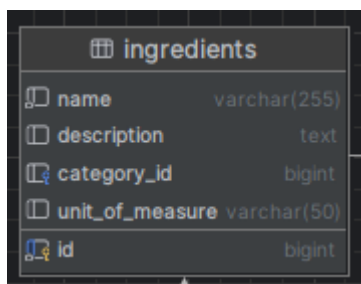
name: Найменування товару.

description: Докладний опис характеристик продукту.

category_id (Foreign Key): Посилання на категорію, до якої належить інгредієнт.

unit_of_measure: Одиниця виміру (кілограм, літр, штука тощо).

Зовнішній ключ category_id з налаштуванням ON DELETE SET NULL гарантує, що при видаленні категорії інгредієнт не буде видалений, але втратить прив'язку до категорії.



ingredients	
name	varchar(255)
description	text
category_id	bigint
unit_of_measure	varchar(50)
id	bigint

Рис. 3.4-Таблиця Ingredients

Таблиця Suppliers зберігає інформацію про компанії-постачальники. Основні поля включають:

id (Primary Key): Унікальний ідентифікатор постачальника.

user_id (Foreign Key, Unique): Унікальне посилання на користувача з роллю SUPPLIER.

company_name: Офіційна назва компанії.

address: Юридична та фізична адреса постачальника.

tax_id: Ідентифікаційний номер платника податків.

rating: Середній рейтинг постачальника від 0.00 до 5.00.

description: Опис діяльності компанії та асортименту продукції.

Зв'язок один-до-одного з таблицею Users через унікальний зовнішній ключ забезпечує, що кожен постачальник має власний обліковий запис користувача.

suppliers	
user_id	bigint
company_name	varchar(255)
address	text
tax_id	varchar(50)
rating	numeric(3,2)
description	text
id	bigint

Рис. 3.5-Таблиця Suppliers

Таблиця Offers реєструє комерційні пропозиції від постачальників. Основні поля включають:

id (Primary Key): Унікальний ідентифікатор пропозиції.

supplier_id (Foreign Key): Посилання на постачальника, що створив пропозицію.

ingredient_id (Foreign Key): Посилання на інгредієнт, який пропонується.

price: Ціна за одиницю товару.

available_quantity: Доступна кількість товару на складі.

min_order_quantity: Мінімальна партія для замовлення.

is_active: Статус активності пропозиції.

valid_from, valid_until: Період дієвості комерційної пропозиції.

created_at, updated_at: Часові мітки створення та оновлення.

Каскадне видалення (ON DELETE CASCADE) гарантує автоматичне видалення пропозицій при видаленні постачальника або інгредієнта.

offers	
supplier_id	bigint
ingredient_id	bigint
price	numeric(10,2)
available_quantity	numeric(10,2)
min_order_quantity	numeric(10,2)
is_active	boolean
valid_from	date
valid_until	date
created_at	timestamp
updated_at	timestamp
id	bigint

Рис. 3.6-Таблиця Offers

Таблиця Orders фіксує замовлення покупців. Основні поля включають:

id (Primary Key): Унікальний ідентифікатор замовлення.

buyer_id (Foreign Key): Посилання на користувача-покупця.

total_amount: Загальна вартість замовлення.

status: Поточний статус замовлення (PENDING, CONFIRMED, SHIPPED, DELIVERED, CANCELLED).

delivery_address: Адреса доставки товару.

created_at, updated_at: Часові мітки створення та модифікації замовлення.

Обмеження CHECK забезпечує, що статус може приймати лише визначені значення, що відповідають життєвому циклу замовлення.

orders	
buyer_id	bigint
total_amount	numeric(12,2)
status	varchar(20)
delivery_address	text
created_at	timestamp
updated_at	timestamp
id	bigint

Рис. 3.7-Таблиця Orders

Таблиця Order_Items деталізує склад кожного замовлення. Основні

поля включають:

id (Primary Key): Унікальний ідентифікатор позиції замовлення.

order_id (Foreign Key): Посилання на замовлення.

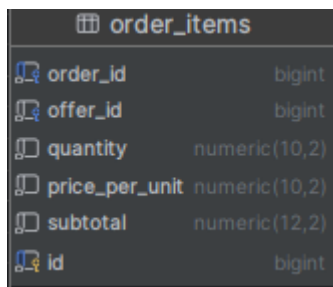
offer_id (Foreign Key): Посилання на конкретну пропозицію
постачальника.

quantity: Кількість одиниць товару в замовленні.

price_per_unit: Ціна за одиницю на момент створення замовлення.

subtotal: Проміжна сума для даної позиції ($\text{quantity} \times \text{price_per_unit}$).

Зв'язок з таблицями Orders та Offers через каскадне видалення
забезпечує цілісність даних при видаленні замовлення.



order_items	
order_id	bigint
offer_id	bigint
quantity	numeric(10,2)
price_per_unit	numeric(10,2)
subtotal	numeric(12,2)
id	bigint

Рис. 3.8-Таблиця Order_Items

Таблиця Reviews містить відгуки покупців про постачальників. Основні
поля включають:

id (Primary Key): Унікальний ідентифікатор відгуку.

order_id (Foreign Key): Посилання на замовлення, за яким залишено
відгук.

buyer_id (Foreign Key): Посилання на покупця-автора відгуку.

supplier_id (Foreign Key): Посилання на постачальника, якого
оцінюють.

rating: Числова оцінка від 1 до 5.

comment: Текстовий коментар з деталями враження від співпраці.

created_at: Часова мітка створення відгуку.

Обмеження CHECK ($\text{rating} \geq 1 \text{ AND } \text{rating} \leq 5$) гарантує коректність
оцінок в діапазоні п'ятибальної шкали.

Індекси для оптимізації продуктивності

Для прискорення виконання запитів створено індекси на найбільш використовувані поля:

Індекс на email користувачів для швидкого пошуку при авторизації

Індекси на зовнішні ключі для оптимізації JOIN операцій

Індекси на поля is_active та status для фільтрації даних

Індекси на buyer_id та supplier_id для звітності та аналітики

Для кожної сутності системи було створено структуру, що визначає будову відповідної таблиці бази даних.

Тригери автоматичного оновлення

Реалізовано механізм автоматичного оновлення поля updated_at через тригери:

Функція update_updated_at_column() автоматично встановлює поточний час при модифікації запису

Тригери застосовано до таблиць users, offers та orders для відстеження історії змін

Ця структура бази даних забезпечує ефективне зберігання та управління даними B2B платформи з дотриманням принципів нормалізації та цілісності даних.

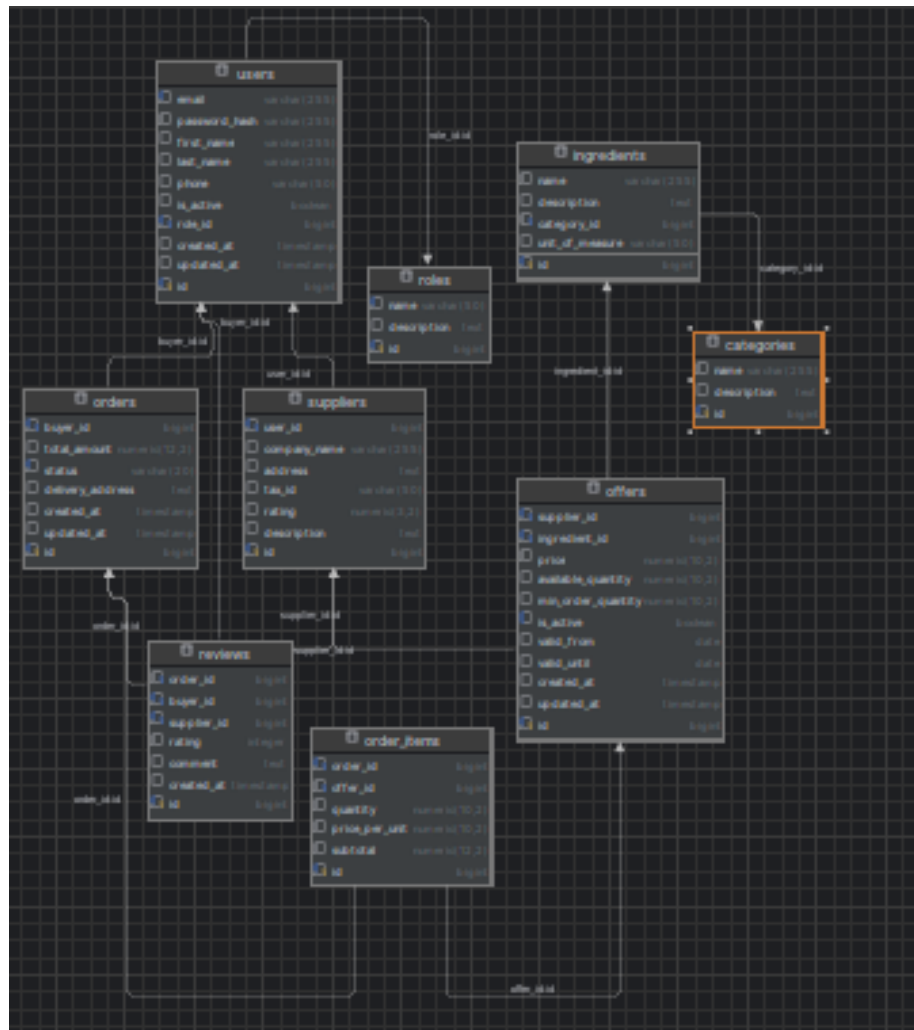


Рис. 3.9-ERD-діаграма

3.2. Back-end

Контролери є точкою входу для всіх HTTP запитів від клієнтських додатків. Контролери отримують запити, валідують вхідні дані, викликають відповідні методи бізнес-логіки з шару сервісів та формують HTTP відповіді. Цей шар реалізує REST API endpoints та обробляє маршрутизацію запитів.

AuthController - обробляє запити на реєстрацію, автентифікацію та вихід користувачів з системи.

```

1  package nykon.foodmarket.controller;
2
3  > import ...
12
13  < @RestController new *
14  @RequestMapping("/api/auth")
15  @RequiredArgsConstructor
16  @CrossOrigin(origins = "*", maxAge = 3600)
17  public class AuthController {
18
19      private final AuthService authService;
20
21
22      @PostMapping("/register") new *
23      public ResponseEntity<User> register(@RequestBody RegisterRequest request) {
24          User user = authService.register(request);
25          return ResponseEntity.status(HttpStatus.CREATED).body(user);
26      }
27
28      @PostMapping("/login") new *
29      public ResponseEntity<LoginResponse> login(@RequestBody LoginRequest request) {
30          LoginResponse response = authService.login(request);
31          return ResponseEntity.ok(response);
32      }

```

Рис. 3.10-Приклад класа-Контроллера

UserController - управляє інформацією про користувачів, дозволяє отримувати та оновлювати дані поточного користувача.

AdminController - надає функціонал для адміністративного управління системою, включаючи перегляд всіх користувачів, управління їх ролями та статусами.

CategoryController - забезпечує операції для категорій продукції.

IngredientController - реалізує управління інгредієнтами, включаючи створення, перегляд, оновлення та видалення.

SupplierController - управляє профілями постачальників, дозволяє переглядати інформацію про компанії та їхні пропозиції.

OfferController - обробляє операції з комерційними пропозиціями постачальників, включаючи створення, редагування, активацію/деактивацію

та пошук пропозицій.

OrderController - забезпечує функціонал замовлень, дозволяє покупцям створювати замовлення, переглядати історію, а також управляти елементами замовлення та змінювати статуси.

Сервіси містять бізнес-логіку додатку та є посередником між контролерами і репозиторіями. Сервіси інкапсулюють складні операції, виконують валідацію даних, координують роботу кількох репозиторіїв та забезпечують транзакційність операцій.

AuthService - реалізує логіку автентифікації та авторизації користувачів. Відповідає за реєстрацію нових користувачів, перевірку облікових даних, генерацію JWT tokenів та управління сесіями. Використовує BCrypt для хешування паролів та взаємодіє з JwtTokenProvider для роботи з токенами.

```
public class AuthService {  
  
    private final UserRepository userRepository;  
    private final RoleRepository roleRepository;  
    private final PasswordEncoder passwordEncoder;  
    private final AuthenticationManager authenticationManager;  
    private final JwtTokenProvider jwtTokenProvider;  
  
    public User register(RegisterRequest request) {  
        if (userRepository.findByEmail(request.getEmail()) != null)  
            throw new RuntimeException("Email already exists");  
        User user = new User();  
        user.setEmail(request.getEmail());  
        user.setPasswordHash(passwordEncoder.encode(request.getPassword()));  
        user.setFirstName(request.getFirstName());  
        user.setLastName(request.getLastName());  
        user.setPhone(request.getPhone());  
        user.setIsActive(true);  
  
        Role buyerRole = roleRepository.findByName("BUYER")  
            .orElseThrow(() -> new RuntimeException("Default role not found"));  
        user.setRole(buyerRole);  
  
        return userRepository.save(user);  
    }  
}
```

Рис. 3.11-Приклад класа-Сервіса

UserService - управляє даними користувачів, надає методи для отримання інформації про поточного користувача, оновлення профілю та зміни налаштувань. Забезпечує перевірку унікальності email при реєстрації.

AdminService - реалізує адміністративний функціонал, включаючи отримання списків всіх користувачів, зміну ролей користувачів, управління

статусом активності облікових записів.

CategoryService - забезпечує логіку роботи з категоріями продукції. Виконує валідацію назв категорій, перевіряє унікальність, управляє створенням, оновленням та видаленням категорій. Перевіряє можливість видалення категорії (чи не використовується вона в інгредієнтах).

IngredientService - управляє інгредієнтами системи, включаючи створення, оновлення, видалення та пошук. Реалізує логіку фільтрації інгредієнтів за категоріями, перевіряє існування категорії при створенні інгредієнта, валідує одиниці виміру.

SupplierService - обробляє бізнес-логіку постачальників. Відповідає за створення та оновлення профілів компаній, отримання профілю поточного користувача-постачальника.

OfferService - керує комерційними пропозиціями постачальників. Реалізує створення пропозицій з прив'язкою до поточного постачальника реалізує пошук та фільтрацію пропозицій.

OrderService - управляє життєвим циклом замовлень.

Всі сервіси анотовані @Service та @Transactional, що забезпечує управління транзакціями на рівні методів.

Репозиторії забезпечує абстракцію доступу до бази даних та інкапсулює всі операції з персистентними даними. Репозиторії використовують Spring Data JPA для автоматичної генерації SQL запитів на основі назв методів та надають можливість створення власних запитів через анотації @Query.

RoleRepository - надає методи для роботи з ролями користувачів.

UserRepository - управляє операціями з користувачами в базі даних.

CategoryRepository - забезпечує доступ до категорій продукції.

IngredientRepository - надає функціонал для роботи з інгредієнтами.

SupplierRepository - управляє даними постачальників.

OfferRepository - забезпечує роботу з комерційними пропозиціями.

```

@Repository 5 usages new *
public interface SupplierRepository extends JpaRepository<Supplier, Long> {
    Optional<Supplier> findById(Long userId); 3 usages new *

    Page<Supplier> findByRatingGreaterThanOrEqual(BigDecimal rating, Pageable pageable); 1 usage

    @Query("SELECT DISTINCT s FROM Supplier s JOIN s.offers o WHERE o.ingredient.category.id = :categoryId")
    Page<Supplier> findByOffersIngredientCategoryId(@Param("categoryId") Long categoryId, Pageable pageable);

    @Query("SELECT DISTINCT s FROM Supplier s JOIN s.offers o " +
        "WHERE s.rating >= :rating AND o.ingredient.category.id = :categoryId")
    Page<Supplier> findByRatingGreaterThanOrEqualAndOffersIngredientCategoryId(
        @Param("rating") BigDecimal rating,
        @Param("categoryId") Long categoryId,
        Pageable pageable);
}

```

Рис. 3.12-Приклад інтерфейсу-Репозиторію

OrderItemRepository - надає доступ до елементів замовлень.

ReviewRepository - забезпечує роботу з відгуками.

Шар моделей містить Java класи, що представляють структуру даних і безпосередньо відображаються на таблиці бази даних через JPA/Hibernate ORM.

Role - представляє роль користувача в системі. Відображається на таблицю roles та містить поля: id (первинний ключ), name (назва ролі: ADMIN, SUPPLIER, BUYER), description (опис ролі). Має зв'язок One-to-Many з сутністю User через колекцію користувачів. Анотована @Table(name = "roles").

User - центральна сутність системи, що представляє користувача. Відображається на таблицю users та містить поля: id, email, passwordHash, firstName, lastName, phone, isActive, createdAt, updatedAt. Має зв'язок Many-to-One з сутністю Role через поле role (анотоване @ManyToOne та @JoinColumn(name = "role_id")). Має зв'язок One-to-One з сутністю Supplier через поле supplier (анотоване @OneToOne(mappedBy = "user")). Використовує @JsonIgnore на полі supplier для запобігання циклічній серіалізації.

Category - представляє категорію продуктів харчування.

Відображається на таблицю categories та містить поля: id, name, description. Має зв'язок One-to-Many з сутністю Ingredient через колекцію інгредієнтів, що належать до цієї категорії.

Ingredient - описує інгредієнт/товар, доступний для торгівлі.

Відображається на таблицю ingredients та містить поля: id, name, description, unitOfMeasure (одиниця виміру). Має зв'язок Many-to-One з сутністю Category через поле category (анотоване @ManyToOne та @JoinColumn(name = "category_id")). Має зв'язок One-to-Many з сутністю Offer через колекцію пропозицій по цьому інгредієнту.

Supplier - представляє постачальника/компанію. Відображається на таблицю suppliers та містить поля: id, companyName, address, taxId, rating, description. Має зв'язок One-to-One з сутністю User через поле user (анотоване @OneToOne та @JoinColumn(name = "user_id", unique = true)). Використовує @JsonBackReference для уникнення циклічних посилань при серіалізації. Має зв'язок One-to-Many з сутністю Offer через колекцію пропозицій постачальника.

Offer - представляє комерційну пропозицію від постачальника.

Відображається на таблицю offers та містить поля: id, price, availableQuantity, minOrderQuantity, isActive, validFrom, validUntil, createdAt, updatedAt. Має зв'язок Many-to-One з сутністю Supplier через поле supplier (анотоване @ManyToOne та @JoinColumn(name = "supplier_id")). Має зв'язок Many-to-One з сутністю Ingredient через поле ingredient (анотоване @ManyToOne та @JoinColumn(name = "ingredient_id")). Має зв'язок One-to-Many з сутністю OrderItem.

Order - представляє замовлення покупця. Відображається на таблицю orders та містить поля: id, totalAmount, status (enum: PENDING, CONFIRMED, SHIPPED, DELIVERED, CANCELLED), deliveryAddress, createdAt, updatedAt. Має зв'язок Many-to-One з сутністю User через поле buyer (анотоване @ManyToOne та @JoinColumn(name = "buyer_id")). Має зв'язок One-to-Many

з сутністю OrderItem через колекцію orderItems (анотоване @OneToMany(mappedBy = "order", cascade = CascadeType.ALL)).

OrderItem - представляє окрему позицію в замовленні. Відображається на таблицю order_items та містить поля: id, quantity, pricePerUnit, subtotal. Має зв'язок Many-to-One з сутністю Order через поле order (анотоване @ManyToOne та @JoinColumn(name = "order_id")). Має зв'язок Many-to-One з сутністю Offer через поле offer (анотоване @ManyToOne та @JoinColumn(name = "offer_id")).

Review - представляє відгук покупця про постачальника. Відображається на таблицю reviews та містить поля: id, rating, comment, createdAt. Має зв'язок Many-to-One з трьома сутностями: Order через поле order, User (як buyer) через поле buyer, та Supplier через поле supplier. Всі зв'язки анотовані відповідними @ManyToOne та @JoinColumn.

3.2.1 Тестування back-end

Для верифікації коректності роботи back-end частини системи та перевірки функціональності REST API було використано інструмент Postman - популярну платформу для тестування та документування API. Postman дозволяє формувати HTTP запити різних типів (GET, POST, PUT, DELETE, PATCH), передавати параметри, заголовки та тіло запиту, а також аналізувати отримані відповіді від сервера.

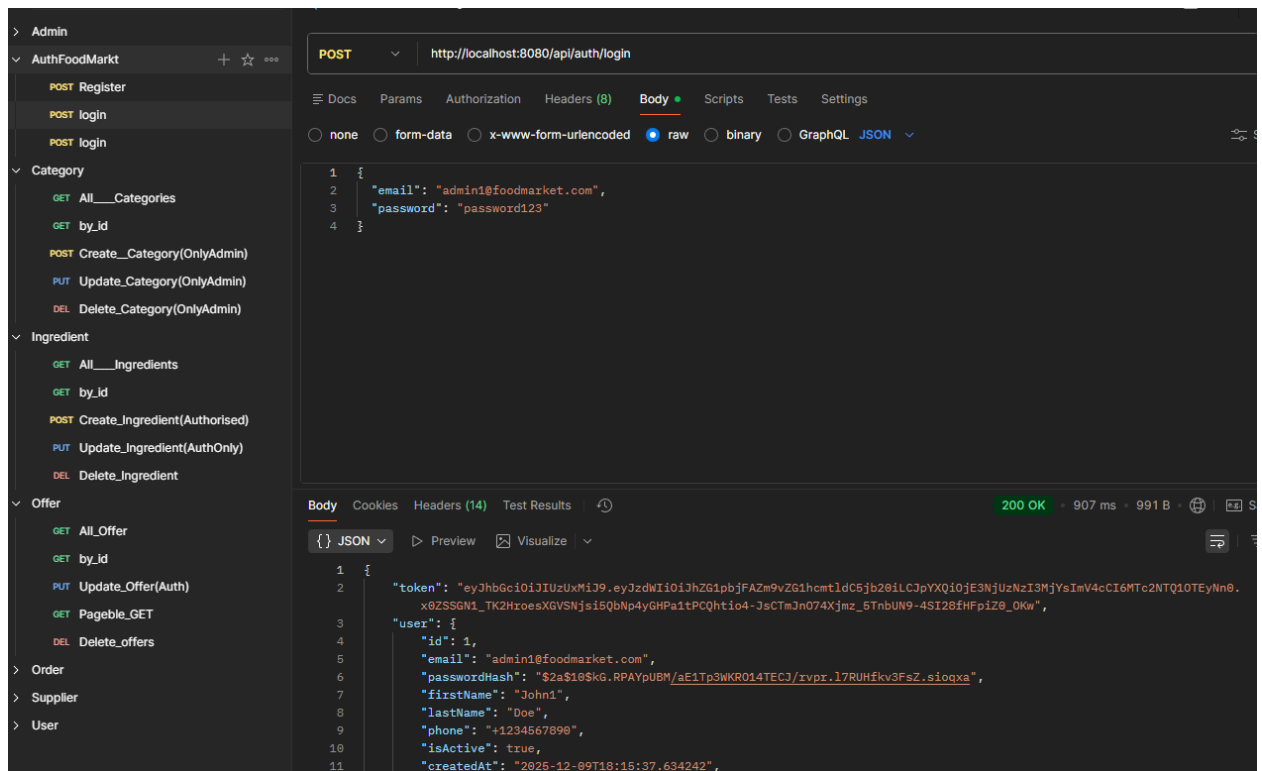


Рис. 3.13-Приклад тестування через запити в Postman

3.3. Frontend

Front-end частина системи FoodMarket реалізована на базі фреймворку Angular з використанням модульної архітектури, що забезпечує чітке розділення відповідальності між компонентами та полегшує масштабування додатку. Структура проєкту організована у кореневій директорії `src/app/` та поділяється на кілька основних блоків:

Кореневий рівень (`src/app/`):

Містить головні конфігураційні файли додатку:

`app.module.ts` - кореневий модуль, що об'єднує всі інші модулі та налаштовує провайдери

`app-routing.module.ts` - конфігурація маршрутизації

`app.component.ts/html/css` - кореневий компонент з базовою структурою (navbar, router-outlet, footer)

Директорія `core/`:

Містить основні сервіси, моделі даних, guards та interceptors, які використовуються по всьому додатку:

guards/ - містить AuthGuard для захисту маршрутів від неавторизованого доступу та RoleGuard для перевірки прав доступу за ролями

interceptors/ - AuthInterceptor для автоматичного додавання JWT токена до HTTP заголовків

models/ - TypeScript інтерфейси для типізації даних (User, Category, Ingredient, Supplier, Offer, Order), що відповідають структурі entities на backend

```
import { Supplier } from './supplier.model';
import { Ingredient } from './ingredient.model';

export interface Offer {
  id: number;
  supplier: Supplier;
  ingredient: Ingredient;
  price: number;
  availableQuantity: number;
  minOrderQuantity: number;
  isActive: boolean;
  validFrom: Date;
  validUntil: Date;
  createdAt: Date;
  updatedAt: Date;
}
```

Рис. 3.14-Приклад моделі

services/ - бізнес-логіка та взаємодія з API: CategoryService, IngredientService, SupplierService, OfferService, OrderService, AuthService, CartService

```

1  > import ...
6
7  @Injectable({ Show usages
8    providedIn: 'root'
9  })
10 export class CategoryService {
11   private apiUrl: string = `${environment.apiUrl}/categories`;
12
13   constructor(private http: HttpClient) {} no usages
14
15   getAllCategories(): Observable<Category[]> { Show usages
16     return this.http.get<Category[]>(this.apiUrl);
17   }
18
19   getCategoryById(id: number): Observable<Category> { Show usages
20     return this.http.get<Category>({ url: `${this.apiUrl}/${id}` });
21   }
22
23   createCategory(category: Category): Observable<Category> { Show usages
24     return this.http.post<Category>(this.apiUrl, category);
25   }
26
27   updateCategory(id: number, category: Category): Observable<Category> { Show
28     return this.http.put<Category>({ url: `${this.apiUrl}/${id}`, category):

```

Рис. 3.15-Приклад сервісу

Директорія features/:

Містить функціональні модулі додатку, кожен з яких відповідає за окремий розділ системи:

auth/ - модуль автентифікації з компонентами login (форма входу) та register.

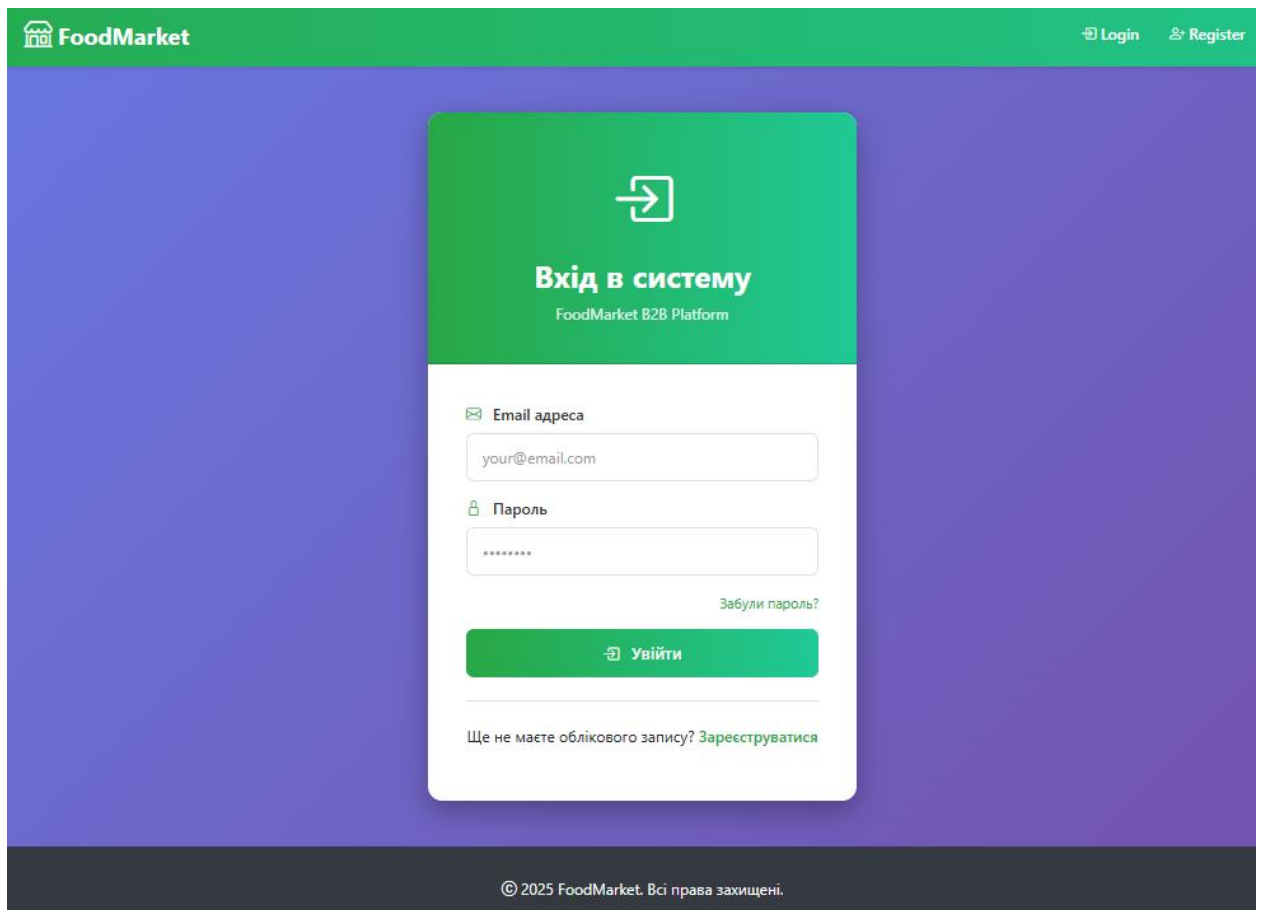
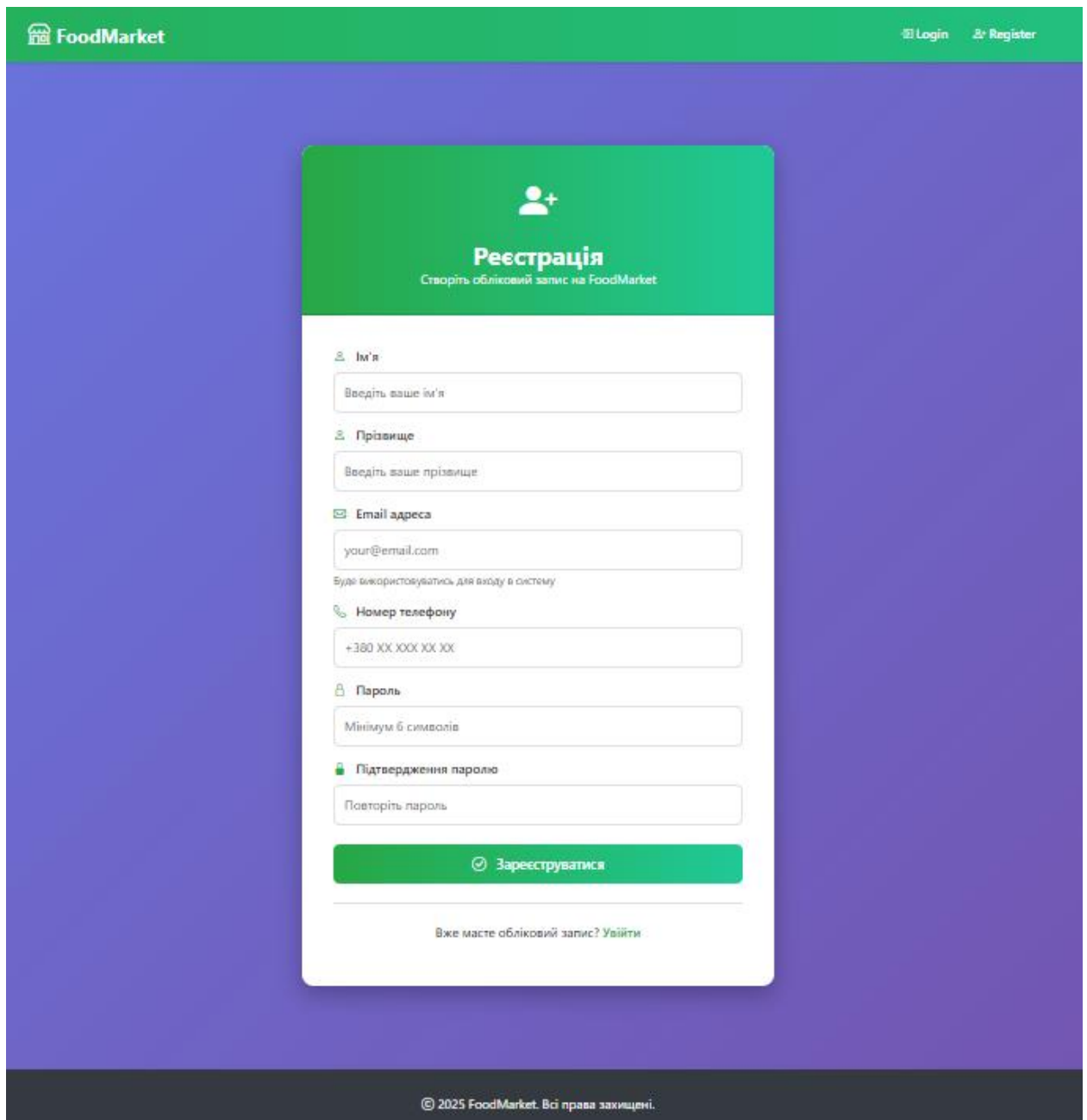


Рис. 3.16-Сторінка для входу в систему



The image shows a registration form for FoodMarket. The form is centered on a purple background. It has a green header with the FoodMarket logo and 'Login' and 'Register' links. The form itself has a green header with a user icon and the title 'Реєстрація' (Registration), with the subtitle 'Створіть обліковий запис на FoodMarket' (Create an account on FoodMarket). The form contains several input fields: 'Ім'я' (Name) with placeholder 'Введіть ваше ім'я' (Enter your name), 'Прізвище' (Surname) with placeholder 'Введіть ваше прізвище' (Enter your surname), 'Email адреса' (Email address) with placeholder 'your@email.com' and a note 'Буде використовуватись для входу в систему' (Will be used for system login), 'Номер телефону' (Phone number) with placeholder '+380 XX XXX XX XX', 'Пароль' (Password) with placeholder 'Мінімум 6 символів' (Minimum 6 symbols), and 'Підтвердження паролю' (Password confirmation) with placeholder 'Повторіть пароль' (Repeat password). A green button with a checkmark and the text 'Зареєструватися' (Register) is at the bottom of the form. Below the button is a link: 'Вже маєте обліковий запис? Увійти' (Already have an account? Log in). The footer of the page is dark grey with the text '© 2025 FoodMarket. Всі права захищені.' (© 2025 FoodMarket. All rights reserved.)

FoodMarket Login Register

Реєстрація
Створіть обліковий запис на FoodMarket

Ім'я
Введіть ваше ім'я

Прізвище
Введіть ваше прізвище

Email адреса
your@email.com
Буде використовуватись для входу в систему

Номер телефону
+380 XX XXX XX XX

Пароль
Мінімум 6 символів

Підтвердження паролю
Повторіть пароль

Зареєструватися

Вже маєте обліковий запис? Увійти

© 2025 FoodMarket. Всі права захищені.

Рис. 3.17-Реєстрація в систему

home/ - головна сторінка з описом платформи, відображенням категорій та кнопками для неавторизованих користувачів

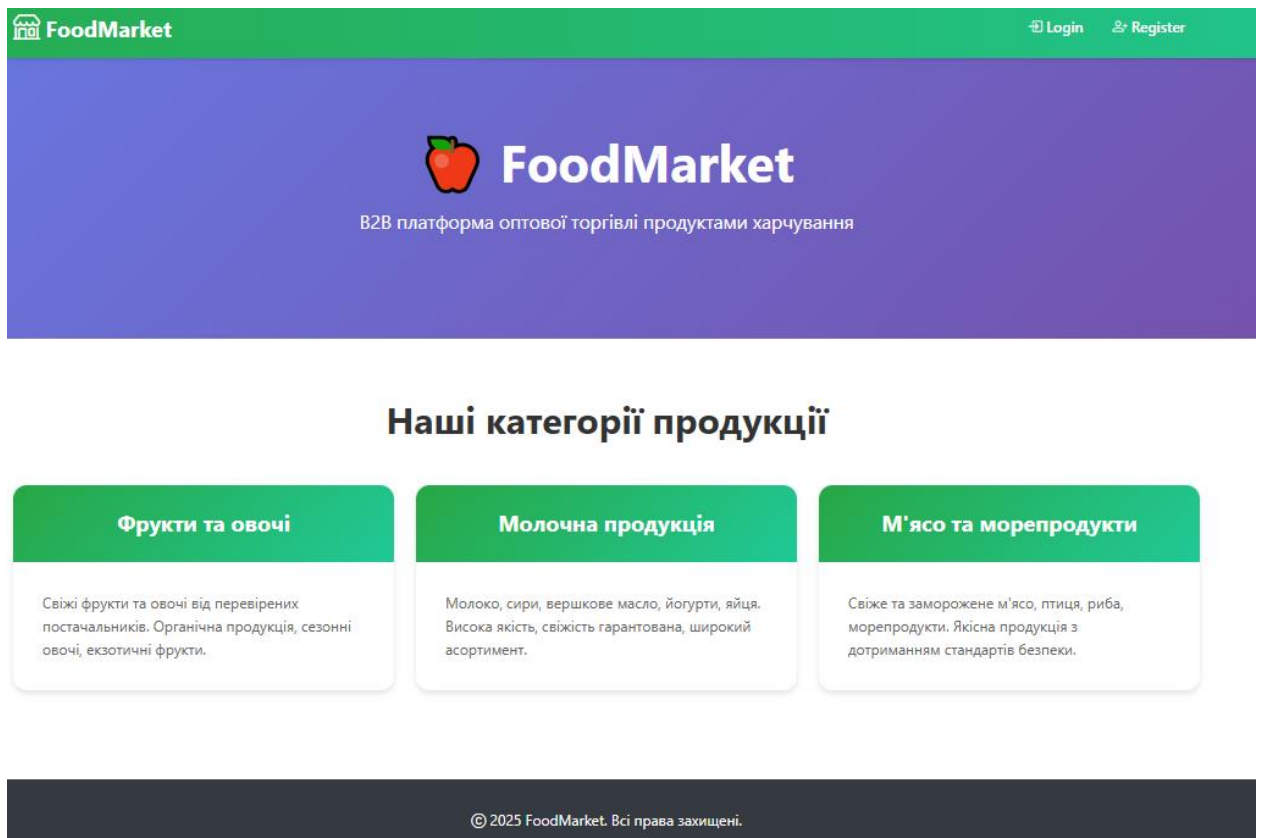


Рис. 3.18-Домашня сторінка

categories/ - модуль управління категоріями для адміністраторів, включає category-list (таблиця категорій з пошуком) та category-form (форма створення/редагування)

ingredients/ - модуль управління інгредієнтами для адміністраторів, містить ingredient-list (таблиця з фільтрацією за категоріями) та ingredient-form (форма створення/редагування інгредієнтів)

suppliers/ - модуль роботи з постачальниками, включає supplier-profile (перегляд профілю компанії з рейтингом) та supplier-form (створення та редагування профілю постачальника)

Мої пропозиції

Поточні пропозиції

Томати свіжі
Фрукти та овочі

Активна

Ціна за кг	Доступно	Мін. замовлення
25.50 грн	1000 кг	10 кг

Термін дії
до 15.01.2025

Редагувати

Деактивувати

Морква органічна
Фрукти та овочі

Активна

Ціна за кг	Доступно	Мін. замовлення
18.00 грн	500 кг	5 кг

Термін дії
до 20.01.2025

Редагувати

Деактивувати

Салат айсберг
Фрукти та овочі

Неактивна

Ціна за шт	Доступно	Мін. замовлення
12.00 грн	300 шт	5 шт

Термін дії
Прострочено

Редагувати

Активувати

Створити нову пропозицію

Інгредієнт

Оберіть інгредієнт...

Доступна кількість

1000

Дійсна з

дд.мм.тттг

Ціна за одиницю (грн)

25.50

Мінімальна кількість замовлення

10

Дійсна до

дд.мм.тттг

Створити пропозицію

Рис. 3.19-Сторінка постачальника для створення пропозицій

offers/ - модуль управління пропозиціями, містить offer-browse (каталог пропозицій для покупців з фільтрами), offer-list (список власних пропозицій для постачальників) та offer-create (форма створення нової пропозиції)

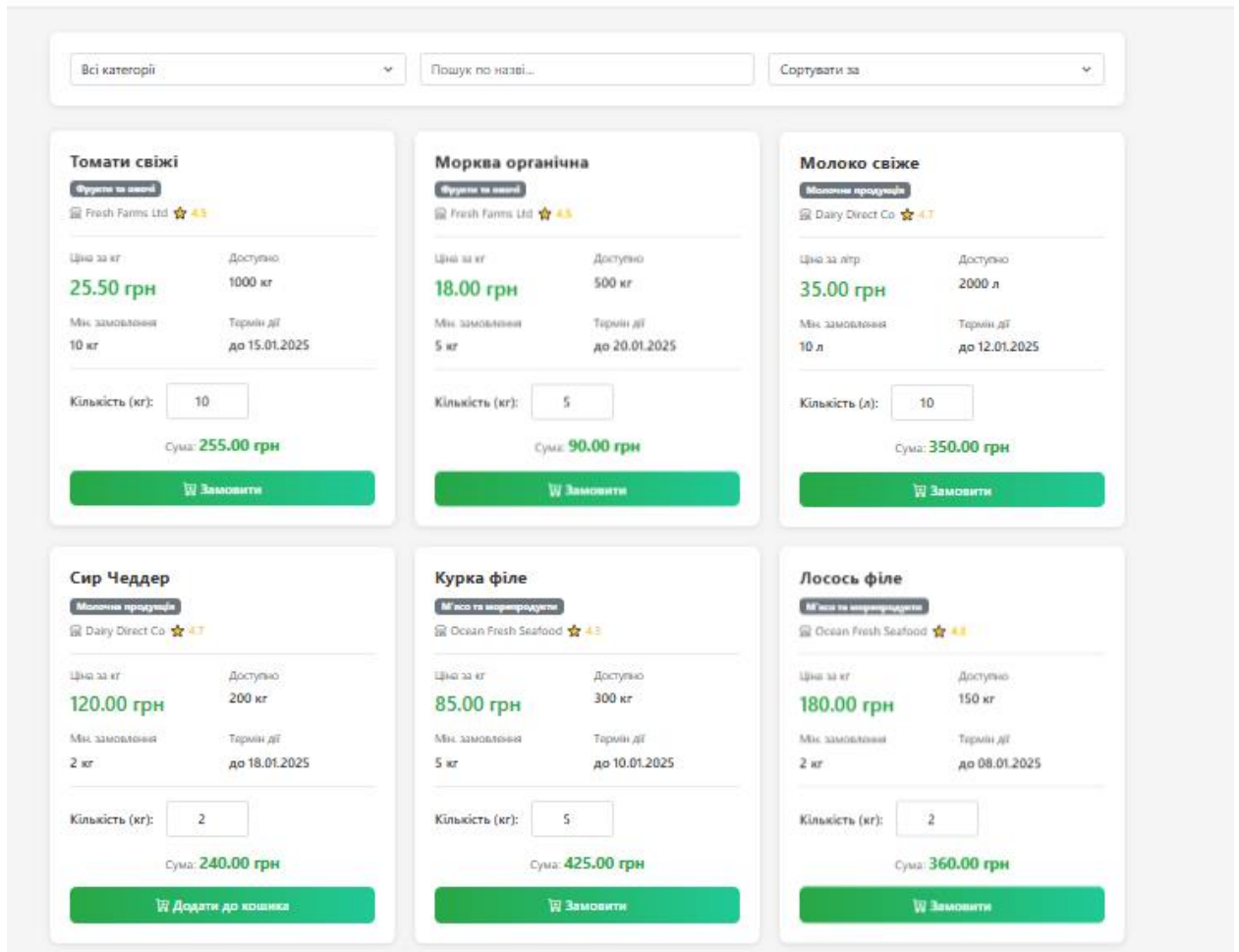


Рис. 3.20-Перегляд пропозицій

orders/ - модуль роботи із замовленнями, включає order-list (історія замовлень з фільтрацією за статусом) та order-create (оформлення нового замовлення з кошиком)

Кожен feature-модуль має власний routing модуль для налаштування внутрішньої маршрутизації .

Директорія shared/:

Містить спільні компоненти, що використовуються на всіх сторінках:

navbar/ - навігаційна панель з меню.

footer/ - футер з інформацією про платформу та контактними даними.

Висновки до розділу 3

У третьому розділі здійснено проектування та реалізацію B2B платформи оптової торгівлі продуктами харчування FoodMarket, визначивши архітектурні рішення та технологічний стек для побудови системи. Для організації бази даних обрано реляційну СУБД PostgreSQL, взаємодія з якою відбувається через Spring Data JPA з використанням об'єктно-реляційного відображення (ORM).

Розроблено багатoshарову архітектуру backend-частини, що включає шари контролерів, сервісів, репозиторіїв та моделей даних. Реалізовано систему автентифікації на основі JWT токенів, що забезпечує розмежування доступу між трьома ролями користувачів: адміністраторами, постачальниками та покупцями.

Подальшим кроком стало проведення тестування функціоналу системи за допомогою інструменту Postman, що підтвердило коректність роботи REST API endpoints. Front-end частину системи розроблено з використанням сучасних веб-технологій. Реалізований користувацький інтерфейс відзначається зручною навігацією, чітким структуруванням інформації та комплексним функціоналом для управління бізнес-процесами оптової торгівлі.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП-ПРОЄКТУ НА БАЗІ ПЛАТФОРМИ

4.1. Опис ідеї стартапу та її бізнес-модель

Проект платформи FoodMarket є інноваційним рішенням у сфері оптової торгівлі продовольчими товарами, що поєднує ефективність сучасних цифрових інструментів із прозорістю комерційних операцій. Основною ідеєю стартапу є створення зручної, безпечної та функціональної онлайн-платформи для взаємодії між постачальниками продуктів харчування та покупцями (ресторанами, кафе, магазинами). Платформа орієнтована на малий та середній бізнес у сфері громадського харчування, роздрібної торгівлі, а також окремих підприємців, які потребують якісних продуктів за конкурентними цінами.

Головною метою стартапу є забезпечення прозорості та простоти організації торгових операцій між постачальниками та покупцями. Це досягається шляхом впровадження інтегрованих інструментів для управління пропозиціями, оформлення замовлень, відстеження поставок та системи рейтингів. Унікальність ідеї полягає в тому, що платформа надає:

Можливість створювати та управляти власними торговими пропозиціями. Інструменти для встановлення цін, мінімальних партій замовлення та термінів дії пропозицій. Систему управління залишками товарів в режимі реального часу. Доступ до широкої аудиторії потенційних покупців. Зручний пошук та фільтрацію продуктів за категоріями, ціною та постачальниками. Можливість порівнювати пропозиції різних постачальників

комісійна модель: платформа стягує невелику комісію за кожну транзакцію. Це забезпечує стабільний дохід для подальшого розвитку;

преміум-функції: організатори зборів можуть купувати доступ до додаткових функцій, таких як розширена аналітика, персоналізовані звіти та інструменти автоматизації.

Таблиця 4.1

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення онлайн платформи для забезпечення торгівлі між постачальниками та закупівельниками у ресторанному бізнесі	Використовується ресторанами, кафе, постачальниками	Простота створення пропозиції.
Надання інструментів для автоматизації процесів закупівель та звітності перед бізнесом	Закупівельники які потребують ефективного інструменту для управління пропозиціями та звітами	Можливість автоматичного формування звітів
Створення інтерактивного інтерфейсу до зручного процесу онлайн закупівель	Підходить для бізнесу який шукає способи закупівель через UX/UI	Покращений досвід користувач, підвищення рівня довіри до платформи за допомогою зручного інтерфейсу

4.2. Технологічний аудит ідеї

Таблиця 4.2

Технологічний аудит ідеї

№	Ідея проекту	Технології та реалізація	Наявність технологій	Доступність технологій
1	Створення веб-платформи для закупівель	Мова програмування Java	Є в наявності	Доступна безкоштовно
2	Реалізації фронтенд платформи	Angular	Є в наявності	Доступна безкоштовно
3	Використання бекенд фреймворку	Spring Boot	Є в наявності	Доступна безкоштовно
4	СУБД	PostgreSQL	Є в наявності	Доступна безкоштовно
5	Середовище розробки	IntelliJ Idea	Є в наявності	Доступна Безкоштовно для базового використання
6	Дизайн інтерфейсу користувача	Figma та Canva	Є в наявності	Доступні безкоштовно для базового використання

Проект базується на сучасних програмних засобах, що забезпечують високу ефективність розробки, масштабованість та зручність у використанні.

На бекенд-рівні використовується фреймворк Spring Boot, який

забезпечує модульну архітектуру для зручного управління складністю додатка та надає широкий спектр готових рішень для швидкої розробки enterprise-додатків. Для роботи з базами даних застосовується PostgreSQL, що надає потужні інструменти для обробки великих обсягів даних, підтримує ACID-транзакції для забезпечення їхньої цілісності та пропонує розширені можливості для роботи зі складними запитам.

Для взаємодії з базою даних використовується Spring Data JPA з Hibernate як ORM інструментом, що дозволяє працювати з даними на рівні об'єктів Java без необхідності написання складних SQL-запитів. Архітектура проєкту побудована за принципом розділення відповідальності (Separation of Concerns) з чітким поділом на шари: Controller (обробка HTTP-запитів), Service (бізнес-логіка) та Repository (доступ до даних).

Середовище розробки IntelliJ IDEA, завдяки своїй потужній інтеграції з Spring Framework, Maven та іншими інструментами Java-екосистеми, дозволяє розробникам ефективно створювати, налагоджувати та розгортати програмний код. Для управління залежностями та збірки проєкту використовується Apache Maven, який автоматизує процес компіляції, тестування та пакування додатку.

Для тестування API використовується Postman, який забезпечує зручний інтерфейс для перевірки роботи endpoints. Система контролю версій Git з платформою GitHub забезпечує ефективну командну роботу та відстеження змін у коді.

Таким чином, використання цих технологій дозволяє реалізувати проєкт відповідно до сучасних стандартів якості enterprise-розробки, забезпечує його надійність, масштабованість та доступність для розробників і кінцевих користувачів. Вибір Java та Spring Boot як основного технологічного стеку гарантує довгострокову підтримку, широку спільноту розробників та можливість інтеграції з корпоративними системами.

4.3. Оцінка ринкових перспектив

Оцінка ринкових можливостей є критично важливим етапом у розробці та впровадженні стартап-проєкту. Вона дозволяє зрозуміти поточний стан ринку, визначити ключових гравців, дослідити динаміку попиту та аналізувати потенційні перешкоди для виходу на ринок. Для цього були враховані основні показники, які характеризують ринкові можливості для платформи FoodMarket, такі як кількість головних гравців, динаміка ринку оптової торгівлі продуктами харчування, наявність бар'єрів для входу та вимоги до сертифікації продовольчих товарів.

Оцінка була виконана з урахуванням ринку B2B-платформ для торгівлі продуктами харчування, які надають послуги з організації оптових закупівель для підприємств громадського харчування та роздрібною торгівлі. Основними конкурентами виступають як спеціалізовані онлайн-маркетплейси продовольчих товарів, так і традиційні оптові бази, що вже мають усталені бізнес-процеси та клієнтську базу. Незважаючи на це, завдяки орієнтації на малий та середній бізнес, цифровізацію процесу замовлень та систему прозорих рейтингів постачальників, розроблена платформа має унікальну конкурентну перевагу.

Динаміка ринку характеризується стабільним зростанням попиту на цифрові рішення в секторі B2B-торгівлі продуктами харчування, особливо після пандемії COVID-19. За даними аналітичних агентств, ринок онлайн-торгівлі в сегменті HoReCa (готелі, ресторани, кафе) та роздрібною торгівлі продовжує активно розвиватися, демонструючи щорічний приріст 15-20%. Це свідчить про високу зацікавленість бізнесу у доступних, швидких та прозорих способах організації закупівель продовольчих товарів.

Таблиця 4.3

Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн/ум.од	Залежно від угод з бізнесом
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Обмежень немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	10%

Таким чином, аналіз ринку свідчить про перспективність запуску платформи для онлайн закупівель у ресторанному бізнесі.

Таблиця 4.4

Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару

Таблиця 4.4(продовження)

1	Потреба в пошуку пропозицій від постачальників	Малий та середній бізнес що потребує регулярних поставок продукції	Активність у пошуку Платформ із прозорою можливістю вибору конкретного постачальника	Зручний Інтерфейс та пошук потрібних товарів
2	Потреба в пошуку клієнтів	Малий та середній бізнес що потребує потенційних клієнтів	Активність у пошуку Платформ із можливістю розміщення товарів	Надання регулярних звітів, клієнтська база

Таблиця 4.5

Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Неоптимальна система монетизації	Проект не отримає достатньо коштів на життя	Швидка заміна монетизаційної політики

Таблиця 4.5(продовження)

2	Новий сильний конкурент	Користувачі припиняють послуговуватися функціями проєкту	Впровадження нового функціоналу, відсутнього у конкурента.
3	Зловмисні дії	Ризик кібератак, спрямованих на крадіжку даних користувачів або компрометацію системи	Використання сучасних можливостей хмарних сервісів

Таблиця 4.6

Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Пошук додаткових інвесторів	Можливість залучення інвестицій через співпрацю з міжнародними партнерами	Розширення партнерської мережі та залучення експертів для оптимізації розподілу коштів

Таблиця 4.6(продовження)

2	Впровадження нових функцій	Додавання інтерактивних елементів, таких як онлайн чати чи інструменти для візуалізації даних	Підвищення залучення користувачів та покращення зручності використання платформи
3	Розширення географії діяльності	Вихід на міжнародні ринки	Адаптація до регіональних особливостей

Таблиця 4.7

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства(можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції: олігополія	На ринку присутні кілька великих компаній, що надають схожі рішення	Розробка унікального функціоналу, маркетингові кампанії з акцентом на диференціацію продукту
За рівнем конкурентної боротьби: глобальний	Конкуренти можуть запропонувати рішення у багатьох країнах світу	Розширення ринків збуту, адаптація до вимог міжнародних стандартів

Таблиця 4.7(продовження)

За галузевою ознакою: внутрішньо-галузева	Конкуренція у вузькому сегменті	Створення технологій, які зможуть відповідати потребам більш широкої аудиторії
Товарно-родова конкуренція	Існують альтернативні продукти, які можуть виконуватисьхожі функції	Покращення функціоналу основного продукту, додавання нових можливостей
Ненова конкуренція	Змагання в удосконаленні дизайну, якості, користувацького досвіду	Постійне оновлення дизайну і технічних характеристик продукту
Не марочна інтенсивність	Споживачі більше орієнтуються на функціонал, ніж на бренд	Акцент на демонстрації практичних переваг продукту перед конкурентами

Таблиця 4.8

Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
------------------	---------------------------	-----------------------	---------------	---------	------------------

Таблиця 4.8(продовження)

Приклад и :	ChefHero, MarketMan	Локальні бізнеси з продажу	Хмарні сервіси AWS, GCP	Постачальник и та ресторанний бізнес	Сервіси закупівель загального призначення
Висновк и	Ринок характериз ує ться високою конкуренці є ю з боку великих гравців	Високі бар'єри входу, але зростанн я інтересу до нішевих рішень	Висока залежніст ь від якісних ІТ- сервісів	Різноманітніс ть клієнтів з унікальними запитами	Потреба в інноваційно му підході для залучення користувачів

Таблиця 4.9

Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Відсутність необхідності великих затрат	Для розробки проєкту, потрібна мінімальна кількість платних рішень для хостинга .

Таблиця 4.9(продовження)

2	Можливість успішної монетизації	Потенційна кількість зацікавлених магазинів у співпраці, надасть можливість отримувати достатньо інформації, а також платню, за надання послуг.
3	Інтуїтивний інтерфейс	Простий і зрозумілий дизайн, який дозволяє будь-якому користувачеві легко створювати нові збори,переглядати статистику та підтримувати проекти.

Таблиця 4.10

Порівняльний аналіз сильних та слабких сторін даним проектом

№	Фактор конкуретноспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з продуктом						
			-3	-2	-1	0	+1	+2	+3
1	Відсутність необхідності великих затрат			+					
2	Можливість успішної монетизації						+		
3	Інтуїтивний інтерфейс						+		

Таблиця 4.11

SWOT- аналіз стартап-проекту

Сильні сторони	Слабкі сторони
1. Унікальність функціоналу	1. Наявність сильних конкурентів на ринку

Таблиця 4.11 (продовження)

2. Сучасні технологічні рішення	2. Залежність від зовнішніх інвестицій
3. Високий рівень безпеки даних	3. Залежність від зовнішніх інвестицій
4. Орієнтація на клієнтоорієнтованість	4. Обмежені ресурси для маркетингових кампаній
Можливості:	Загрози:
1.Залучення стратегічних партнерів	1.Швидкий вихід нових конкурентів
2. Розширення Функціональності відповідно до потреб клієнтів	2.Економічна нестабільність, що впливає на фінансування
3. Привернення інвестицій для масштабування	4.Зміни в законодавстві, що впливають на діяльність

Таблиця 4.12

Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Терміни реалізації
1	Проведення масштабної інформаційної кампанії у соціальних мережах	Висока, основний ресурс— фінансування для рекламного бюджету	1-2 місяці

Таблиця 4.12(продовження)

2	Організація партнерських заходів із кафе та ресторанами	Висока, головний ресурс — співпраця із залученими партнерами	2-4 місяці
3	Публікація аналітичних статей про платформу на спеціалізованих ресурсах	Середня, головний ресурс — підготовка матеріалів та комунікація з медіа	3-6 місяців

4.4. Розробка ринкової стратегії стартап-проекту

Таблиця 4.13

Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу всегмент
1	Ресторани та кафе	Висока	Високий	Середня	Простий
2	Постачальники	Висока	Висока	Висока	Середній
3	Представники кулінарних шкіл	Середня	Середній	Низька	Середня

Таблиця 4.14

Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
---	--------------------------------------	---------------------------	--	---------------------------

Таблиця 4.14(продовження)

1	Стратегія розширення співпраці	Сегментоване охоплення	Налагодження партнерств з ресторанами впровадження нових функцій для залучення закупівельників	Стратегія партнерського маркетингу
2	Стратегія диференціації	Виокремлення ключових переваг	Унікальний дизайн платформи, простота використання	Стратегія підвищення лояльності
3	Стратегія технологічного зростання	Нарощування функціоналу та автоматизації	Розширення функціональних можливостей платформи, інтеграція API для зручності користувачів	Стратегія інноваційного розвитку

Таблиця 4.15

Визначення базової стратегії конкурентної поведінки

№	Чи є проект "першопрохідцем" на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Так, мета компанії — залучення нових споживачів через зручність платформи	Компанія частково копіює прозорість, але додає унікальні функції для моніторингу	Стратегія зайняття конкурентної ніші

Таблиця 4.15(продовження)

2	Так	Компанія орієнтується на створення нового сегменту	Ключові особливості включають простоту інтерфейсу та функції автоматизації закупівель	Стратегія інноваційного розвитку
3	Ні	Основний акцент на переваги сервісу для існуючих користувачів	Повністю унікальний функціонал для інтеграції іншими маркетплейсами	Стратегія підвищення клієнтської лояльності

Таблиця 4.16

Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Прозорість закупівель. Інтуїтивний дизайн, зручний доступ до статистики	Стратегія диференційованого маркетингу	Унікальність функціоналу, адаптивний дизайн	Унікальність, Надійність, Прозорість
2	Інтергація з кафе та ресторанами	Стратегія посилення партнерства	Кросплатформеність, ефективність	Інноваційність, Доступність

4.5. Розробка маркетингової програми стартап-проекту

Таблиця 4.17

Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами(існуючі або такі, що потрібно створити)
1	Забезпечення прозорості закупівель	Повна прозорість процесу закупівель	Автоматизовані звіти для постачальників та бізнесу
2	Інтуїтивно зрозумілий інтерфейс для користувачів	Простота навігації та легкість у використанні	Унікальний дизайн, адаптація для різних пристроїв та навчальні матеріали для нових користувачів

Таблиця 4.18

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за здумом	Веб-система для закупівель для кафе та ресторанів

Таблиця 4.18(продовження)

II. Товар у реальному виконанні	Властивості/характерис тики	М/Нм	Вр/Тх /Тл/Е/Ор
	Швидкодія та надійність	Нм	Тх
	Ефективність	Нм	Тл
	Користувацький інтерфейс	Нм	Е
	Якість: функціональна веб-система з модулями управління персоналом		
	Пакування: власний сайт		
III. Товар із підкріплення М	Марка: FoodMarket		
	Програмний продукт		
Програмний продукт			
За рахунок чого потенційний товар буде захищено від копіювання: захищений програмний код із захистом авторських прав, а також використання ліцензії			

Таблиця 4.19

Визначення меж встановлення ціни

№	Рівень цін на товари- замінники	Рівень цін на товари аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межа встановлення ціни на товар/послугу
1	Бескоштовно	Бескоштовно	>10\$	0\$ - 5\$

Таблиця 4.20

Формування системи збуту

№	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
---	---	---	-------------------------	-----------------------------

Таблиця 4.20(продовження)

1	Оплата рекламної кампанії.	-	До приватних підприємств яким необхідне просування власних	Веб-сайт, а також рекомендації між споживачами
---	----------------------------	---	--	--

Таблиця 4.21

Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Доступ до Платформи через веб-браузер	Електронна пошта, соціальні мережі, реклами в Інтернеті	Інноваційність, зручність, надійність	Представлення унікально о функціоналу та користі	Пояснення переваг, функцій Платформи та її місії
Використання мобільного застосунку	Push-сповіщення реклама у мобільних додатках	Доступність, мобільність, інтерактивність	Мотивування завантаження мобільного додатку	Демонстрація зручності використання в мобільному форматі

Таблиця 4.21 (продовження)

Створення та підтримка ресторанного бізнесу	Онлайн форуми та спільноти	Ефективність, підтримка важливих ініціатив	Довіра до платформи	Презентація реальних прикладів успішних співпраць
---	----------------------------	--	---------------------	---

Висновки до розділу 4

Четвертий розділ присвячено розробці стартап-проекту на основі розробленої платформи FoodMarket, орієнтованої на оптимізацію взаємодії між постачальниками продуктів харчування та підприємствами громадського харчування і роздрібною торгівлі.

У межах цього розділу було описано ідею стартап-проекту, включаючи функціонал платформи, її унікальні можливості щодо організації B2B-торгівлі продовольчими товарами та ключові переваги над конкурентами. Проведено аналіз ринкових конкурентів у сегменті онлайн-маркетплейсів продуктів харчування, що дозволило чітко виділити сильні сторони платформи (система рейтингів, прозорість ціноутворення, автоматизація замовлень) та спланувати її подальший розвиток;

Проведено аналіз сучасних ринкових можливостей для запуску проекту в секторі цифрової торгівлі продовольчими товарами. Виявлено ключові фактори, які можуть сприяти або перешкоджати успішному виходу платформи на ринок, зокрема зростаючий попит на цифровізацію закупівель, необхідність побудови довіри між учасниками та конкуренція з традиційними оптовими базами. Сформовано план дій для мінімізації ризиків, пов'язаних із загрозами, такими як складність залучення постачальників та обмежені маркетингові ресурси на початковому етапі;

Визначено перспективи запуску продукту на ринок у порівнянні з конкурентами. Розроблено базову стратегію розвитку та конкурентної поведінки, орієнтовану на створення екосистеми довіри через систему верифікації постачальників та відгуків покупців, а також визначено цільові групи користувачів (локальні фермерські господарства, оптові постачальники, ресторани, кафе, роздрібні магазини) і методи їх залучення. Це включало аналіз ключових ринкових сегментів та потенційних клієнтів, а також розробку механізмів позиціонування платформи як надійного

цифрового посередника в ланцюгу поставок продуктів харчування;

Розроблено комплексну маркетингову програму для просування платформи серед постачальників та покупців. Описано способи досягнення цільової аудиторії через спеціалізовані галузеві ресурси, професійні спільноти в соціальних мережах, участь у виставках та B2B-заходах харчової індустрії. Визначено ключові маркетингові повідомлення, спрямовані на демонстрацію економічної вигоди від використання платформи (зниження витрат на пошук постачальників, прозорість цін, скорочення часу на оформлення замовлень), та запропоновано механізми стимулювання користувачів до активного залучення через програми лояльності та знижки на комісію для перших користувачів.

У підсумку, розроблено стартап-проект, який включає всі необхідні аспекти для успішного запуску та просування платформи FoodMarket на ринок B2B-торгівлі продуктами харчування. Проект враховує як ринкові виклики (конкуренція, необхідність побудови довіри, логістичні особливості), так і потреби користувачів (зручність, прозорість, економія часу та коштів), що робить його перспективним у сучасних умовах цифровізації торгівлі продовольчими товарами.

ВИСНОВКИ

Магістерська робота "Система автоматизації закупівель у ресторанному бізнесі" висвітлює актуальну проблему, пов'язану з автоматизацією процесів закупівель продуктів харчування для закладів HoReCa (готелі, ресторани, кафе). Робота спрямована на розробку повнофункціонального веб-сервісу, який забезпечує спрощену взаємодію між постачальниками харчових продуктів та представниками ресторанного бізнесу, оптимізуючи процес формування та обробки замовлень.

Актуальність дослідження визначається необхідністю вирішення проблем, з якими стикаються заклади ресторанного бізнесу: відсутність централізованої платформи для пошуку постачальників, складність порівняння цін та асортименту, неефективність комунікації з постачальниками через різні канали зв'язку, а також необхідність ручної обробки великої кількості замовлень. В контексті цифровізації бізнес-процесів це стає критичним аспектом підвищення ефективності роботи закладів харчування.

Мета дослідження полягає у створенні ефективної B2B платформи, яка враховує специфічні потреби ресторанного бізнесу щодо закупівель продуктів харчування та надає зручний інструмент для управління пропозиціями постачальників, каталогізації продукції за категоріями та інгредієнтами, а також оформлення замовлень покупцями.

Об'єктом дослідження є процеси закупівель продуктів харчування у ресторанному бізнесі, а предметом — методи та засоби автоматизації взаємодії між постачальниками та представниками закладів HoReCa через веб-платформу з використанням сучасних технологій розробки програмного забезпечення.

Науково-практична цінність полягає у розробці комплексного рішення на основі багатoshарової REST API архітектури з використанням Spring Boot

для серверної частини, Angular для клієнтського інтерфейсу та PostgreSQL для зберігання даних, що забезпечує масштабованість, безпеку та зручність використання системи різними типами користувачів.

У першому розділі проведено огляд існуючих рішень для веб-сервісів закупівель у ресторанному бізнесі. Детально розглянуто платформи ChefHero, MarketMan, Chefsheet, Orderly, PurchasePlus та Transgourmet/Prodega з метою виявлення їхніх переваг та недоліків. Аналіз показав, що наявні рішення не завжди враховують специфіку українського ринку, зокрема необхідність детального опису складу продукції через інгредієнти, гнучкого управління категоріями товарів та адаптації до локальних бізнес-процесів.

У другому розділі проведено огляд технологій для розробки системи. Для backend-частини розглянуто Spring Framework та FastAPI, для frontend — React, Angular та Vue, для баз даних — PostgreSQL та MongoDB. Проаналізовано варіанти розгортання: AWS, Azure та On-Premises рішення. На основі порівняльного аналізу обрано технологічний стек: Spring Boot для серверної частини, Angular для клієнтського інтерфейсу та PostgreSQL для реляційного зберігання даних.

У третьому розділі здійснено проектування та реалізацію веб-платформи для автоматизації закупівель. Розроблено структуру бази даних з підтримкою сутностей користувачів, категорій, інгредієнтів, комерційних пропозицій та замовлень. Реалізовано багатоварштову архітектуру backend-частини з використанням Spring Data JPA для персистентності даних та Spring Security для автентифікації на основі JWT токенів. Створено frontend-частину на Angular, яка включає модуль адміністратора для управління довідниками, інтерфейс постачальника для створення пропозицій та інтерфейс покупця для перегляду каталогу і оформлення замовлень.

У четвертому розділі розроблено стартап-проект на базі платформи. Описано бізнес-модель проекту, проведено технологічний аудит ідеї, здійснено оцінку ринкових перспектив з аналізом попиту та конкурентного

середовища. Розроблено ринкову стратегію та маркетингову програму стартап-проекту, визначено методи монетизації через комісію з угод, преміум-підписки для постачальників та додаткові аналітичні сервіси.

Практичним результатом роботи є повнофункціональний веб-сервіс, який забезпечує повний цикл бізнес-процесів від публікації пропозиції постачальника до формування замовлення покупцем з підтримкою рольової моделі доступу для трьох типів користувачів: адміністраторів, постачальників та покупців (представників ресторанного бізнесу).

Перспективами подальшого розвитку є інтеграція платіжних систем для автоматизації розрахунків, впровадження модуля аналітики та звітності з візуалізацією статистики закупівель, інтеграція з системами управління запасами ресторанів (inventory management), а також розширення функціоналу для автоматичного формування замовлень на основі аналізу залишків продукції.

Отже, дана наукова робота має на меті підкреслити важливість розробки та впровадження спеціалізованих B2B платформ для ефективної автоматизації процесів закупівель у ресторанному бізнесі, що дозволить підвищити ефективність роботи закладів HoReCa, знизити витрати на комунікацію з постачальниками та забезпечити прозорість торгових операцій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. [NetSuite. Restaurant Inventory & Supply Chain Management. <https://www.netsuite.com/>].
2. [Fourth. Restaurant Procurement Challenges and Solutions. <https://uk.fourth.com/article/restaurant-procurement>].
3. [NetSuite. Restaurant Supply Chain Management. <https://www.netsuite.com/>].
4. [Supy. Restaurant Procurement System Overview. <https://www.supy.io/>].
5. [BetaKit. “ChefHero raises \$12.6 million to help restaurants order supplies.” <https://betakit.com/chefhero-raises-12-6-million/>].
6. [Joe Produce. “ChefHero: a better way for restaurants to order from suppliers.” <https://joeproduce.com/>].
7. [The Spoon. “ChefHero Rebrands as Notch to Expand Its Restaurant Supply Platform.” <https://thespoon.tech/chefhero-rebrands-as-notch/>].
8. [MarketMan Official Website. <https://www.marketman.com/>].
9. [G2. “MarketMan Reviews & Ratings.” <https://www.g2.com/products/marketman/reviews>].
[Chefsheet Official Site. <https://www.chefsheets.com/>].
10. [Chefsheet Official Site. <https://www.chefsheets.com/>].
11. [Orderly. Smart Restaurant Procurement Platform. <https://www.getorderly.com/>].
12. [Hospitality Tech. “How Orderly Helps Restaurants Save Time on Supply Management.” <https://hospitalitytech.com/>].
13. [PurchasePlus. Procurement Software for the Hospitality Industry. <https://www.purchaseplus.com/>].
14. [Transgourmet Schweiz. About the Company. <https://www.transgourmet.ch/>].

15. [Prodega / Transgourmet. Official Website. <https://www.prodega.ch>].
16. [COOP Group Annual Report. Transgourmet Overview. <https://www.coop.ch/en/company/transgourmet.html>].
17. [Amazon Web Services. Frontend vs Backend - How Backend Works. <https://aws.amazon.com/what-is/backend/>].
18. [Spring Framework. Official Documentation. <https://spring.io>].
19. [Baeldung. Introduction to Spring Framework and AOP. <https://www.baeldung.com/spring-framework>].
20. [FastAPI. Official Documentation. <https://fastapi.tiangolo.com>].
21. [Real Python. Data Validation with Pydantic and FastAPI. <https://realpython.com/pydantic-fastapi/>].
22. [TestDriven.io. Building Fast APIs with FastAPI. <https://testdriven.io/blog/fastapi/>].
23. [React. Official Documentation. <https://react.dev>].
24. [W3Schools. React Tutorial. <https://www.w3schools.com/react/>].
25. [MDN Web Docs. Introducing JSX and React Components. https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_components].
26. [Angular. Official Documentation. <https://angular.io>].
27. [W3Schools. Angular Tutorial. <https://www.w3schools.com/angular/>].
28. [GeeksForGeeks. Introduction to Angular CLI. <https://www.geeksforgeeks.org/angular-cli/>].
29. [Vue.js. Official Guide. <https://vuejs.org>].
30. [W3Schools. Vue Tutorial. <https://www.w3schools.com/vue/>].
31. [MDN Web Docs. Vue Components and State Management. https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Vue_components].
32. [PostgreSQL Global Development Group. PostgreSQL Documentation. <https://www.postgresql.org/>].
33. [W3Schools. PostgreSQL Tutorial.

- <https://www.w3schools.com/postgresql/>].
34. [MongoDB Inc. MongoDB Documentation. <https://www.mongodb.com/>].
 35. [W3Schools. MongoDB Tutorial. <https://www.w3schools.com/mongodb/>].
 36. [Docker Inc. What is Docker? <https://www.docker.com/resources/what-container/>].
 37. [Amazon Web Services. AWS Cloud Overview. <https://aws.amazon.com/what-is-aws/>].
 38. [Microsoft Azure. Overview. <https://azure.microsoft.com/>].
 39. [Red Hat. What is On-Premise Software? <https://www.redhat.com/en/topics/cloud-computing/what-is-on-premise-software>].
 40. [Red Hat. What is OpenShift? <https://www.redhat.com/en/technologies/cloud-computing/openshift>].
