

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

_____ Аліна САВЧЕНКО

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

**Тема: «Вебзастосунок для обліку товарів та продажу з
використанням мікросервісної архітектури»**

Виконавець:

Кирило ГРЕЧКА

Керівник:

к.т.н., доцент Олена ТОЛСТІКОВА

Нормоконтролер:

к.т.н., доцент Олена ТОЛСТІКОВА

КИЇВ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютерних інформаційних технологій

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ:
завідувач кафедри КІТ
Аліна САВЧЕНКО

(підпис)

«_____» _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Гречки Кирила Сергійовича

(ПІБ випускника)

1. Тема роботи: «Вебзастосунок для обліку товарів та продажу з використанням мікросервісної архітектури» затверджена наказом ректора № 1976/ст від 29.09.2023р.
2. Термін виконання роботи: з 02 жовтня 2023 року по 31 грудня 2023 року.
3. Вихідні дані до роботи: застосунок на мовах програмування Java, JavaScript та використанням технологій Docker, Keycloak для демонстрації вебзастосунку для обліку і продажу товарів.
4. Зміст пояснювальної записки: 1. Аналіз та поняття обліку та продажу. 2. Проектування вебзастосунку. 3. Розробка та тестування вебзастосунку.
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: Перелік основних слайдів: 1. Мікросервісна архітектура. 2. Шлюз. 3. Авторизація користувачів. 4. База даних товарів. 5. База даних зображень. 6. Облік товарів. 7. Головна сторінка вебзастосунку. 8. Сторінка магазину.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Аналіз предметної області та огляд аналогів. Написання 1 розділу, аналіз та поняття технології	02.10.2023- 16.10.2023	
2.	Вибір та опис використаних технологій. Написання 2 розділу, проектування вебзастосунку	17.10.2023- 30.10.2023	
3.	Написання 3 розділу, розробка та тестування вебзастосунку	31.10.2023- 20.11.2023	
4.	Загальне редагування та друк пояснювальної записки	21.11.2023- 29.11.2023	
5.	Проходження нормоконтролю, перепліт пояснювальної записки.	30.11.2023- 18.12.2023	
6.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	19.12.2023- 22.12.2023	

7. Дата видачі завдання _____ 02.10.2023 р. _____

Керівник кваліфікаційної роботи _____ Олена ТОЛСТІКОВА
(підпис керівника)

Завдання прийняв до виконання _____ Кирило ГРЕЧКА
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Вебзастосунок для обліку товарів та продажу з використанням мікросервісної архітектури» містить: 94 сторінки, 50 рисунків, 21 використаних джерела.

Об'єкт дослідження – створення вебзастосунку з використанням мікросервісної архітектури при проектуванні його компонентів.

Предмет дослідження – аналіз і проектування вебзастосунку та його компонентів для процесу обліку і продажу товарів.

Мета кваліфікаційної роботи – отримати кінцевий продукт у вигляді робочого вебзастосунку для обліку товарів та продажу з використанням мікросервісної архітектури.

Методи дослідження – мова програмування Java, Javascript. Управління ідентифікацією Keycloak. Управління ізольованих linux-контейнерів Docker. Бази даних PostgreSQL, MongoDB.

Результати кваліфікаційної роботи рекомендується використовувати для інтегрування з різними системами для бізнесу.

JAVA, POSTGRESQL, MONGODB, VUE, SPRING BOOT, LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL, ACTIVE DIRECTORY, KEYCLOAK, JAVASCRIPT, CSS, HTML, MAVEN, GRADLE, NODE MODULES, GIT, GITHUB, MICROSERVICES, IDEA.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ТА ПОНЯТТЯ ОБЛІКУ ТА ПРОДАЖУ ТОВАРІВ	10
1.1. Еволюція систем обліку в сучасному бізнесі	10
1.2. Інноваційні підходи до обліку товарів та їх вплив на ефективність.....	13
1.3. Використання аналітики в обліку продажів та прийняття управлінських рішень	23
1.4. Роль інтернет-технологій у вдосконаленні обліку товарів та продаж	26
РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ	34
2.1. Опис вимог до застосунку	34
2.2. Значення обраних технологій у сучасному розвитку	35
2.3. Фундаментальні поняття	38
РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ.....	66
3.1. Розробка frontend системи.....	66
3.2. Розробка pictures системи.....	81
3.3. Розробка gateway системи	82
3.4. Розробка embedded-ldap системи	82
3.5. Розробка backend системи	83
3.6. Розгортання Keycloak та баз даних на Docker.....	85
ВИСНОВКИ	91
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	92

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

- LDAP – Lightweight directory access protocol
- AD – Access directory
- HTTP – Hypertext transfer protocol
- DNS – Domain name system
- API – Application programming interface
- IDE – Integrated development environment
- CTA – Call to click

ВСТУП

Важливість обліку товарів: Ефективний облік товарів є життєво важливим для успіху будь-якого підприємства. Це дозволяє компаніям точно відстежувати запаси, уникати надлишкових або недостатніх запасів і оптимізувати логістику та управління ланцюгом поставок. Точний облік надає дані, необхідні для прийняття обґрунтованих управлінських рішень, таких як планування закупівель, ціноутворення та розробка маркетингових стратегій.

Системи обліку та їх вплив на продажі: Різні методи обліку запасів (наприклад, FIFO - «першим прийшов, першим вийшов», LIFO - «останнім прийшов, першим вийшов», середньозважена вартість) безпосередньо впливають на оцінку запасів. Фінансові результати та стратегії ціноутворення Вибір методу бухгалтерського обліку може вплинути на прибутковість компанії, податкові зобов'язання та здатність конкурувати на ринку.

Технологічні аспекти обліку та продажу товарів: Сучасні технології, такі як системи управління запасами, системи POS (Point of Sale) та електронна комерція, відіграють важливу роль у сучасному обліку та продажах. Вони дозволяють автоматизувати процеси, забезпечуючи більшу точність, швидкість обробки даних і зручність для споживачів. Ці технології також спрощують збір і аналіз даних про клієнтів, що важливо для персоналізованого маркетингу та збільшення продажів.

Виклики та перспективи у сфері обліку та реалізації товарів: Сфера обліку та реалізації товарів постійно змінюється під впливом таких факторів, як зміни законодавства, нові вимоги до звітності та кібербезпеки. Компанії повинні адаптуватися до цих змін, інтегруючи новітні технології та методи бухгалтерського обліку, щоб залишатися конкурентоспроможними.

Ці аспекти створюють міцну основу для вступу, допомагаючи читачеві зрозуміти глибину та важливість бухгалтерського обліку та маркетингу в сучасному бізнес-середовищі.

Актуальність програм бухгалтерського обліку та продажів у сучасному бізнес-середовищі можна висвітлити через такі аспекти:

Автоматизація процесів: у сучасному світі, де швидкість і ефективність є ключовими факторами успіху, автоматизація за допомогою програм бухгалтерського обліку та продажів може значно підвищити продуктивність. Вони автоматизують рутинні завдання, такі як управління запасами, обробка замовлень, виставлення рахунків і звітність.

Аналіз і прийняття обґрунтованих рішень: ці програми збирають великі обсяги даних, що дозволяє проводити глибокий аналіз і отримувати корисну інформацію. Допомагає керівникам приймати більш обґрунтовані рішення щодо продажів, маркетингу, запасів і загальної бізнес-стратегії.

Мобільність і доступність: багато сучасних програм для бухгалтерського обліку та продажів пропонують мобільні версії, що дозволяє керівникам і співробітникам бути в курсі бізнес-процесів незалежно від їх місцезнаходження.

Інтеграція з іншими системами: сучасні програми часто можуть інтегруватися з іншими бізнес-системами та платформами (наприклад, системами CRM, платформами електронної комерції), створюючи синергію та забезпечуючи більшу ефективність управління бізнесом.

Відповідність нормативним вимогам. Забезпечення дотримання нормативних стандартів є важливим для будь-якого бізнесу. Програми для бухгалтерського обліку та продажів допомагають фінансовій звітності та іншим бізнес-операціям відповідати вимогам законодавства.

Підвищення рівня задоволеності клієнтів. Ефективне управління запасами та оптимізовані процеси продажів, увімкнуті програмами, можуть підвищити рівень задоволеності клієнтів, забезпечуючи швидке виконання замовлень і високу якість обслуговування.

Ці аспекти підкреслюють важливість програм бухгалтерського обліку та продажів у сучасному динамічному бізнес-середовищі, де швидкість, точність і ефективність є критично важливими факторами успіху.

Архітектура мікросервісів — це підхід до розробки програмного забезпечення, який передбачає створення програми як набору невеликих незалежних служб, які

спілкуються одна з одною через чітко визначені API. Основні функції та переваги архітектури мікросервісу включають:

Модульність: мікросервіси розроблені як незалежні модулі, які можна легко замінити або оновити, не впливаючи на решту системи. Це сприяє легкості внесення змін і швидкості розвитку.

Масштабованість: кожен мікросервіс можна масштабувати незалежно від інших, що дозволяє ефективно використовувати ресурси та оптимізувати реакцію на зміни навантаження.

Гнучкість у виборі технології: різні мікросервіси можуть використовувати різні технології (мови програмування, бази даних тощо), що дає командам свободу вибору найбільш підходящих інструментів для кожної конкретної служби.

Простота розгортання та відновлення: оскільки мікросервіси автономні, їх можна розгортати окремо, що спрощує оновлення системи та відновлення.

Об'єкт дослідження – створення вебзастосунку з використанням мікросервісної архітектури при проектуванні його компонентів.

Предмет дослідження – аналіз і проектування вебзастосунку та його компонентів для процесу обліку і продажу товарів.

Мета кваліфікаційної роботи – отримати кінцевий продукт у вигляді робочого вебзастосунку для обліку товарів та продажу з використанням мікросервісної архітектури.

Завдання кваліфікаційної роботи полягає у створенні вебзастосунку на міксервісній архітектурі з використанням різних технологій для збереження, обробки даних, авторизації користувачів, докеризації.

РОЗДІЛ 1

АНАЛІЗ ТА ПОНЯТТЯ ОБЛІКУ ТА ПРОДАЖУ ТОВАРІВ

1.1. Еволюція систем обліку в сучасному бізнесі

Ефективний контроль за фінансами є вирішальним для сучасного управління компанією. Розвиток систем обліку відображає значні зміни у бізнес-середовищі та технологічному просторі. Перехід від традиційних ручних методів до використання передових цифрових рішень підкреслює поточну парадигму у фінансовому управлінні та обробці даних.

Ручний облік в епоху початкового бізнесу

На початковому етапі розвитку бізнесу ручний облік відігравав ключову роль у збереженні та систематизації фінансової інформації. Малі підприємства того часу використовували прості записи для відстеження доходів, витрат і фінансових операцій.

Для реєстрації продажів, покупок і витрат використовували щоденники та паперові документи. Цей метод дозволяв підприємцям мати миттєвий доступ до важливої інформації та приймати зважені управлінські рішення.

Хоча ручний метод обліку був досить простим, він забезпечував компаніям можливість підтримувати фінансову строгість та контроль над бюджетом. Його ключова перевага полягала у легкій доступності для невеликих бізнесів з обмеженими коштами, надаючи можливість ефективно вести облік, незважаючи на відсутність складних фінансових систем.

З розвитком технологій і появою електронних систем обліку ручний метод почав витіснятися, але залишився невід'ємною частиною початкового етапу бізнесу. Цей період визначався прагненням до простоти та розуміння самої фінансової ситуації, що зробило ручний облік ефективним інструментом для підприємців того часу.

Кафедра КІТ

НАУ 23 03 04 000 ПЗ

Кафедра КІТ				НАУ 23 03 04 000 ПЗ			
	ПІБ			Літ.	Аркуш	Аркушів	
Розроб.	Гречка К.С.				10	23	РОЗДІЛ 1. АНАЛІЗ ТА ПОНЯТТЯ ОБЛІКУ ТА ПРОДАЖУ ТОВАРІВ ТП-215М -122
Керівник	Толстікова О.В.						
Н.Контр.	Толстікова О.В.						

Механічна ера та початок автоматизації

В двадцятому столітті відбувся розвиток механізації та автоматизації у сфері бізнесу. У цей час підприємства зіткнулися з потребою в удосконаленні та оптимізації своїх процесів. Механічні системи обліку, такі як ручні касові апарати та облікові машини, стали важливими інструментами для управління фінансами та реєстрації трансакцій.

Крім того, цей період характеризувався бажанням автоматизувати рутинні операції. Механічні пристрої дозволяли швидше і точніше обчислювати і записувати операції. Впровадження таких систем сприяло підвищенню ефективності робочих процесів та уникненню помилок у фінансовому обліку.

Поступово з'явилися перші форми комп'ютерів, які відкривали нові можливості для автоматизації бухгалтерського обліку. Це був крок у напрямку використання технологій для оптимізації процесів управління. Однак на той час ці технології були обмежені й використовувалися переважно у великих корпораціях.

Механічна ера та початок автоматизації були визначені прагненням до швидкості та точності в обробці інформації, що призвело до важливого етапу в розвитку систем бухгалтерського обліку та управління бізнесом.

Від комп'ютеризації до інтегрованих систем

У 20-му столітті розвивалася механізація та автоматизація у сфері бізнесу. У цей час компанії зіткнулися з необхідністю вдосконалювати та оптимізувати свої процеси. Механічні облікові системи, такі як ручні реєстратори та бухгалтерські машини, стали важливими інструментами для управління фінансами та реєстрації операцій.

Крім того, цей період характеризувався прагненням до автоматизації рутинних операцій. Механічні пристрої дозволяли швидше і точніше обчислювати і записувати операції. Впровадження таких систем дозволило підвищити ефективність робочих процесів та уникнути помилок у фінансовому обліку.

Поступово з'явилися перші форми комп'ютерів, які відкрили нові можливості для автоматизації бухгалтерського обліку. Це був крок у напрямку використання

технологій для оптимізації процесів управління. Однак на той момент ці технології були обмежені і використовувалися переважно у великих компаніях.

Епоха механіки та початок автоматизації були визначені прагненням до швидкості та точності в обробці інформації, що призвело до важливого етапу в розвитку бухгалтерського обліку та систем управління бізнесом.

Цей еволюційний процес використання технологій в бізнесі віддзеркалює стрімкий розвиток інформаційного суспільства. Використання даних та їхнє інтегроване використання стають ключовими ресурсами для прийняття стратегічних рішень та забезпечення конкурентоспроможності підприємств у глобальному економічному середовищі. На цьому тлі комп'ютеризація виграє нові горизонти, сприяючи автоматизації та точнішому веденню обліку. Програми обліку стають більш динамічними, що оптимізує рутинні завдання та підвищує ефективність управлінських процесів.

Виклики та можливості цифрової трансформації

Незважаючи на те, що цифрова трансформація відкриває багато можливостей для компаній, вона також стикається з важливими проблемами. Одним із ключових аспектів є необхідність значних інвестицій у впровадження нових технологій та навчання персоналу. Це може бути особливо важливим для малих і середніх підприємств, які можуть мати обмежені ресурси.

Ще одним викликом є найбільший ризик кібератак і порушення безпеки даних. Безпека цифрових систем стає невід'ємною частиною успішної трансформації, і вразливі місця можуть вплинути на конфіденційність і цілісність даних.

У зв'язку зі швидкістю змін і впровадженням нових технологій існує також проблема управління змінами та адаптації персоналу до нової фази розвитку. Важливо підтримувати культуру інновацій і навчання, щоб полегшити перехід без проблем.

Незважаючи на ці виклики, цифрова трансформація відкриває багато можливостей. Впровадження штучного інтелекту, аналізу даних та Інтернету речей може підвищити ефективність і забезпечити стратегічні переваги. Нові бізнес-моделі,

такі як платіжні системи та електронна комерція, можуть відкрити нові ринки та сприяти зростанню.

Додатково, цифрова трансформація сприяє поліпшенню відносин з клієнтами шляхом надання індивідуалізованих послуг, що веде до підвищення їхньої задоволеності. Завдяки легшому доступу до даних, компанії також здатні більш точно аналізувати та адаптуватися до змін на ринку.

Всі ці фактори відкривають новий етап у світі бізнесу, де компаніям необхідно адаптуватися до викликів та використовувати переваги цифрової еволюції для забезпечення своєї стабільності та конкурентоздатності.

1.2. Інноваційні підходи до обліку товарів та їх вплив на ефективність

Інновації у сфері обліку активів є стратегічною детермінантою розвитку бізнесу. Облік продукції, трансформований у стратегічну складову, спрямований на оптимізацію бізнес-процесів і досягнення конкурентних переваг. У даному дослідженні розглядається вплив інноваційних підходів на ефективність основних систем обліку продукції.

Сучасні технології передбачають інновації в обліку основної продукції. Використання автоматизованих систем обліку, штучного інтелекту та технологій блокчейн гарантує ефективну реалізацію товарів у режимі реального часу, зменшення помилок та оптимізацію ланцюжка поставок[1].

Сучасні технології в обліку основної продукції вимагають адаптації компаній до швидких змін у бізнес-середовищі. Автоматизація процесів, штучний інтелект для аналізу даних і технології блокчейн для гарантування надійності інформації є основою ефективного обліку товарів.

Технологічні інновації в сфері обліку призводять до прискорення та ускладнення процесів у виробництві та ланцюгах постачання. Штучний інтелект, адаптивні аналітичні платформи та блокчейн технології працюють разом, створюючи еластичне середовище для управління та нагляду за потоками товарів.

Незважаючи на всі переваги, інноваційні підходи до технологічних інструментів бухгалтерського обліку стикаються з проблемами, такими як високі

витрати на впровадження та навчання. Розробка систем, здатних адаптуватися до постійних технологічних змін, є завданням, яке вимагає значних ресурсів і витрат.

Інновації у сфері бухгалтерського обліку спричинили значні зміни в точності та швидкості цього важливого елементу управління бізнесом. Розглянемо, як інноваційні підходи впливають на обидва аспекти: точність і швидкість обліку.

Інноваційні методи внесли значні удосконалення в точність ведення обліку товарів та фінансових транзакцій. Використання передових технологій, таких як штучний інтелект і аналітичні алгоритми, забезпечує автоматизоване виявлення та корекцію помилок, що раніше могли негативно впливати на достовірність даних.

Завдяки цифровим технологіям можна відстежувати рух фінансових товарів і транзакцій в режимі реального часу, знижуючи ризик помилок в облікових даних. Важливо підкреслити, що висока точність бухгалтерського обліку необхідна для ефективного управління бізнесом і розробки стратегічних планів.

З точки зору швидкості, інновації також роблять значний внесок. Використання автоматизованих систем обліку та обробки даних дозволяє скоротити необхідний час на виконання рутинних операцій. Реальний час стає ключовим елементом, який інтегрується в сучасний бухгалтерський підхід.

Інновації також означають зміну традиційних методів обробки даних на використання штучного інтелекту для автоматичного аналізу та використання великих обсягів інформації. Це не тільки допомагає збільшити швидкість обліку, але й відкриває можливості для прогнозування та оптимізації комерційних процесів.

Впровадження аналітичних засобів в систему обліку продукції визначає новий етап в еволюції управління бізнесом. Ця роль аналітики є фундаментальною в контексті поточного бізнес-середовища та прагматичного використання технологій для оптимізації процесів і досягнення стратегічних цілей.

Аналітичні методи перетворюють управління активами в ключовий стратегічний елемент, полегшуючи збір, аналіз та тлумачення великих обсягів даних. Моніторинг запасів у реальному часі, поєднаний з технологічними нововведеннями, дозволяє компаніям швидко адаптуватися до змін на ринку і вдосконалювати свої стратегії управління запасами.

Однією з ключових функцій аналізу є моніторинг і прогнозування попиту на товари. Штучний інтелект і алгоритми аналізу даних, обробка історичних даних і фактори, що впливають, створюють прогнози, які служать основою для ефективного управління запасами та стратегій придбання.

Важливим аспектом ролі аналітики є оптимізація ланцюжка поставок. Аналіз даних про рух постачальників, товарів і логістики допомагає ефективно координувати та оптимізувати всі фази цього процесу.

Завдяки аналітиці можна складати глибоко деталізовані портрети клієнтів, враховуючи їх переваги та поведінку. Це надає змогу цілеспрямовано підходити до маркетингових стратегій та покращувати задоволеність клієнтів. Окрім внутрішнього застосування, аналітика також важлива для розуміння ринкових трендів та нововведень. Аналітичні інструменти сприяють адаптації підприємств до динамічних змін у галузі, дозволяючи оперативно реагувати на нові можливості та розробляти стратегії довгострокового розвитку.

Раціональним наслідком аналітичного підходу стає оптимізація товарного асортименту. Враховуючи результати аналізу продажів і попиту, компанії можуть коригувати свій асортимент, уникати надлишкових запасів і прискорювати обертання товарів. Завдяки Analytics управління ризиками отримує нові інструменти для визначення й оцінки можливих ризиків. Це дозволяє ефективно керувати невизначеністю в ланцюзі поставок і забезпечує гнучкість у вирішенні проблем.

В кінцевому підсумку, значення аналітики у сфері інноваційного обліку товарів є ключовим. Її вплив виходить за рамки лише підвищення ефективності внутрішніх процесів, вказуючи також на напрями розвитку компаній у швидкозмінному ринковому середовищі. Аналітика розкриває можливості для стратегічного управління та досягнення конкурентних переваг за допомогою передових даних та технологій. Впровадження аналітичних методів у облік товарів не тільки забезпечує точність та оптимізацію управлінських рішень, але й відкриває двері до гнучких та адаптивних стратегій, які є важливими для успішної діяльності в умовах постійних змін.

Однією з ключових переваг є здатність аналітики виділяти невидимі зв'язки та шаблони у великих обсягах даних. Це відкриває можливості для глибшого розуміння ринкових тенденцій, уподобань клієнтів і динаміки конкуренції.

Ефективне використання аналітичних інструментів вимагає, щоб компанія була не тільки технічно підготовлена, але й щоб пережила культурну трансформацію. Сприйняття аналізу як стратегічного інструменту, а не лише технічної функції, робить компанію більш адаптованою та здатною вирішувати складні завдання.

Важливо відзначити, що роль аналітики не обмежується використанням оптимізації існуючих процесів. Це також джерело інновацій і вдосконалень. Вивчення аналітичних результатів може запропонувати нові можливості для розвитку бізнесу, адаптації до змін і впровадження кращих практик.

Найважливішим аспектом є постійна еволюція аналітичних інструментів. Враховуючи розвиток технологій, важливо, щоб компанії були готові аналізувати дані про використання нових технологій. Це передбачає постійне навчання персоналу, оновлення програмного забезпечення та впровадження передових рішень у додаток.

Інтеграція обліку з електронною комерцією

Інтеграція бухгалтерського обліку з електронною комерцією є стратегічною ініціативою, яка визначає новий етап у розвитку сучасного бізнес-середовища. Такий підхід відкриває перед компаніями широкі можливості для оптимізації управління фінансами, ведення документації та розширення електронної торгівлі.

Інтеграція бухгалтерського обліку та електронної комерції передбачає об'єднання фінансової та операційної сторін бізнесу. Цей процес можливий завдяки використанню спеціалізованих програмно-технологічних рішень, які забезпечують автоматизацію бухгалтерського обліку, контроль запасів та взаємодію з платформами електронної комерції.

Основною перевагою інтеграції є здатність моніторингу фінансових транзакцій, пов'язаних з електронною торгівлею, в реальному часі, що дає можливість компанії точно контролювати свій фінансовий стан та швидко адаптуватися до змін у

цифровому просторі. Інший важливий елемент інтеграції – підвищення ефективності управління запасами, даючи можливість автоматичного моніторингу руху товарів та оптимізації запасів для запобігання дефіциту або надмірних запасів. Крім того, інтеграція створює єдиний вузол доступу до інформації для різних аспектів бізнесу, сприяючи покращенню комунікацій всередині компанії та забезпечуючи консистентність даних у всіх відділах.

Втім, окрім переваг, інтеграція систем обліку з електронною комерцією включає певні складності. Найважливішим аспектом тут є забезпечення безпеки даних і відповідність законодавчим нормам. Крім того, необхідно звертати увагу на технічні аспекти інтеграції для уникнення труднощів та забезпечення стабільності систем. Інтегрування обліку з електронною комерцією відкриває двері для детальної аналітики та стратегічного планування. Використання спеціалізованих аналітичних інструментів дозволяє компаніям збирати цінні дані про клієнтські уподобання, ефективність товарів і маркетингові тактики. У контексті інтенсивної конкуренції, важливість взаємодії між обліком та електронною комерцією зростає. Компанії, які ефективно інтегрують ці складові, отримують конкурентні переваги завдяки підвищенню продуктивності, зниженню витрат і поліпшенню обслуговування клієнтів. Основним викликом є потреба у навчанні співробітників та створенні умов для ефективного застосування новітніх технологій. Розвиток високого рівня професійних навичок серед персоналу є ключовим для успішної інтеграції та подальшого ефективного функціонування системи.

Важливим етапом інтеграції також є постійна оцінка результатів і коригування стратегії. Враховуючи динаміку електронної комерції та мінливі вимоги ринку, компанії повинні бути готові до постійного вдосконалення своїх процесів. Проте інтеграція бухгалтерського обліку з електронною комерцією є важливим етапом в еволюції комерційних процесів. Це дозволяє не тільки підвищити внутрішню ефективність компанії, але й стати більш конкурентоспроможною в сучасних умовах цифрового ринку.

Виклики інноваційного обліку товарів

Інноваційний облік основних продуктів визначає новий стандарт управління, але разом із його перевагами він стикається з рядом проблем, які вимагають ретельного аналізу та стратегічного підходу. Одним із викликів є необхідність адаптації до швидкої еволюції технологічного середовища. Використання новітніх технологій у сфері бухгалтерського обліку вимагає постійного вивчення та оновлення знань персоналу. Неадекватне формування може призвести до втрати ефективності та зниження конкурентоспроможності.

Ще одним важливим викликом є гарантування кібербезпеки в умовах інноваційного обліку. Зі збільшенням обсягу цифрових даних та їх трансформації в електронні платформи компанії стають більш уразливими до можливих кіберзагроз. Гарантування безпеки та конфіденційності стає важливим завданням, яке вимагає системного підходу та вкладення ресурсів.

Ще одним важливим аспектом є вартість впровадження інноваційного обліку. Придбання та впровадження новітніх технологій може бути дорогим, особливо для малих і середніх підприємств. Необхідно знайти ретельний баланс між витратами на впровадження та очікуваними економічними вигодами. Головним викликом є необхідність переосмислення бізнес-процесів. Інноваційний облік може вимагати не лише технічних змін, а й перегляду корпоративної культури та стратегічних підходів. Переконавання персоналу у важливості змін та їх активна підтримка стають ключовими факторами успішної інтеграції.

Інноваційний бухгалтерський облік також стикається з проблемами в сферах юридичної відповідальності та дотримання нормативних вимог. Зі зростанням кількості нормативних актів і стандартів компанії повинні гарантувати дотримання всіх нормативних актів і стандартів, необхідних для уникнення юридичних проблем. Синергія між технологічними інноваціями, навчанням персоналу, стратегічним управлінням і врахуванням етичних і правових аспектів інноваційного обліку активів є ключем до вирішення цих проблем.

Ще один серйозний виклик полягає у керуванні зростаючим обсягом даних, що виникає в ході інноваційних облікових процесів. Збільшення кількості інформації

може створити проблеми зі зберіганням, обробкою та аналізом даних. Компаніям необхідно розробити дієві стратегії для обробки та захисту даних, забезпечуючи їх надійність та конфіденційність. Окрім того, інноваційний облік стикається з викликами, пов'язаними з нестабільністю технологічного прогресу. Швидкі технологічні зміни можуть зробити застарілими вже впроваджені інновації та вимагають постійного оновлення систем. Також важливим є виклик у забезпеченні ефективної співпраці між різними відділами компанії, що може бути ускладнено внутрішньою бюрократією та обмеженою комунікацією. Нарешті, ключовим аспектом є управління змінами. Інноваційний облік вимагає значних змін у робочих процесах та культурі компанії. Успішна інтеграція включає не лише технічні адаптації, але й підтримку від керівництва, ефективне управління змінами та активне залучення персоналу до новацій. Загалом, інноваційний облік пропонує великі переваги, але подолання цих викликів вимагає не тільки технічної компетенції, але й стратегічного підходу до управління змінами та адаптації до динаміки сучасного бізнесу.

Використання аналітики в обліку продажів та прийняття управлінських рішень

У сучасному діловому світі застосування аналітики у сфері обліку продажів є вирішальним для ефективного керування та прийняття рішень. Аналітичні методи дозволяють компаніям системно аналізувати та розуміти великі масиви даних для розробки інформованих стратегій. Використання аналітики для оцінки ключових показників обсягів продажів допомагає бізнесу отримати детальні уявлення про тенденції та особливості продажів продуктів чи послуг. Аналітичні інструменти полегшують моніторинг та аналіз критичних показників, таких як прибуток, конверсійні ставки, частота повторних покупок та інше, що сприяє прогнозуванню та налагодженню стратегій залучення клієнтів. Аналіз даних продажів також відіграє ключову роль у визначенні ефективності рекламних акцій та маркетингових стратегій.

Використання методів аналізу ROI (повернення від інвестицій) допомагає ідентифікувати найбільш ефективні маркетингові заходи з точки зору прибутковості, що є важливим елементом стратегічного керування. Окрім цього, використання аналітики в обліку продажів впливає на процеси прийняття рішень, надаючи повне розуміння внутрішніх та зовнішніх аспектів бізнесу. Застосування аналітичних моделей допомагає виявляти тренди, слабкі точки та потенційні ризики. Розвиток прогностичних моделей та сценаріїв є суттєвою частиною аналітики в обліку продажів. Ці інструменти дозволяють компаніям створювати прогнози продажів, базуючись на різноманітних даних, що спрощує управління запасами і планування[2].

Ключовим елементом використання аналітики в обліку продажів є здатність забезпечити персоналізований підхід до кожного клієнта. Аналітичні інструменти уможливають налаштування маркетингових стратегій та послуг під конкретні потреби та поведінку покупців. Це не лише покращує задоволення клієнтів, але й сприяє зростанню їхньої вірності та залученості. Аналітика також відіграє важливу роль у ефективному управлінні запасами, дозволяючи точно аналізувати попит, тренди та інші важливі фактори для оптимізації стратегій управління запасами та логістики, уникаючи надлишків чи дефіциту товарів. Крім того, аналітика в обліку продажів є ключовою для виявлення можливостей покращення бізнес-операцій та зниження витрат. Аналіз внутрішніх процесів допомагає виявити місця для ефективності роботи та ресурсного оптимізму. Важливим також є аналіз для визначення цільової аудиторії та розробки стратегій приваблення нових клієнтів. Аналітичні інструменти дозволяють сегментувати ринок і сконцентрувати зусилля на найбільш обіцяючих сегментах, підвищуючи тим самим ефективність маркетингових ініціатив.

Кожен з цих аспектів застосування аналітики в обліку продажів та в процесі управлінського прийняття рішень складає частину цілісної аналітичної стратегії в бізнесі. Аналізуючи дані на різних рівнях та у різних контекстах, компанії здатні швидко адаптуватися до змін у ринковому середовищі та робити обґрунтовані стратегічні рішення для забезпечення успіху та стабільності на ринку. Особливо важливим є включення прогностичного аналізу в облік продажів, який дозволяє

передбачати майбутні ринкові тенденції та своєчасно адаптуватися до них. Прогнозування потенційних розвиткових сценаріїв надає підприємствам можливість гнучко реагувати на коливання попиту, зміни в ринкових умовах та інші зовнішні чинники.

Суттєва роль аналітики в обліку продажів також зосереджена на оптимізації спілкування з клієнтами. Інструменти аналізу відкривають можливості для розробки індивідуалізованих підходів у взаємодії, які відповідають на специфічні потреби кожного клієнта, тим самим забезпечуючи високу якість взаємовідносин та сприяння формуванню тривалих зв'язків. В умовах нинішнього бізнес-ландшафту аналітика в обліку продажів служить стратегічним засобом для керування динамікою та продуктивністю компанії. Неперервне відстеження трендів ринку, змін у споживацькому попиті та поведінці покупців надає можливість не тільки адаптуватися до ринкових умов, але й прогнозувати їх розвиток.

При аналізі аналітики в контексті обліку продажів, ключовим є розуміння її інтеграції в різноманітні аспекти керування підприємством. Зокрема, аналітика фокусується на вдосконаленні процесів у сферах управління збутом, маркетингової діяльності та стратегічного планування. У сфері управління збутом, аналітика використовується для моніторингу продуктивності та ефективності торговельних точок, оцінки результативності рекламних кампаній, та ідентифікації найбільш ефективних каналів розповсюдження. Ці дані лягають в основу прийняття рішень щодо поліпшення розподільчої мережі та маркетингових стратегій.

У сфері маркетингової діяльності, аналітика в обліку продажів забезпечує інструментарій для вимірювання результативності рекламних кампаній. Вона дозволяє ідентифікувати найбільш ефективні рекламні інструменти, оптимізувати бюджети на маркетинг та забезпечити максимальний вплив від інвестованих коштів. У контексті стратегічного планування аналітика в обліку продажів відіграє вирішальну роль у визначенні довгострокових цілей компанії, оцінці ризиків та можливостей на ринку, а також у прогнозуванні конкурентних переваг. Глибоке впровадження аналітики в цих областях підкреслює її значення як інтегрованого елемента управління, де вона виступає не тільки як засіб аналізу, а й як стратегічний

компонент в управлінні. Таким чином, аналітика в обліку продажів не лише вирішує актуальні завдання, але й сприяє розробці довгострокової стратегії та стабільному розвитку підприємства.

У порівнянні з традиційними методами, сучасний підхід до аналітики в обліку продажів включає застосування новітніх технологій. Такі інновації, як штучний інтелект та машинне навчання, стають ключовими компонентами аналітичних систем, дозволяючи автоматизувати обробку великих масивів даних і виділяти значущі висновки. Важливою частиною є також впровадження аналітичних систем прогнозування, що надає компаніям змогу передбачати ринкові тенденції, зміни в попиті та інші важливі фактори, що впливають на обсяг продажів. Таке прогнозування допомагає заздалегідь формувати реакції та планувати виробництво. Окрім того, велике значення аналітики полягає у керуванні ризиками, зокрема у своєчасному виявленні потенційних проблем та ризиків та оцінці їх впливу на продажі, що сприяє ефективному управлінню ризиками та мінімізації можливих втрат.

Підсумовуючи, інноваційне використання аналітики в обліку продажів є ключовим фактором досягнення конкурентної переваги та стабільності в поточному бізнес-середовищі. Забезпечуючи високу точність, прогнозні можливості та ефективне управління, аналітика стає не лише аналітичним інструментом, а й стратегічною складовою процесу управління, наповнюючи компанію потужними інтелектуальними ресурсами.

Ще одним важливим аспектом є використання аналізу для розробки програм лояльності. Завдяки аналізу поведінки та індивідуальним підходам компанії можуть створювати ефективні стратегії залучення та утримання клієнтів. Інтеграція даних з різних каналів дозволяє створити повне бачення клієнта та пропонувати індивідуальні пропозиції.

Уміння ідентифікувати та використовувати хвилі продажів у продажах також важливе. Аналітика дозволяє швидко реагувати на зміни споживчого попиту та реагувати на нові тенденції ринку. Це дозволяє компаніям йти на крок попереду конкурентів і ефективно адаптувати свої стратегії. Окрім соціального впливу, аналіз

продажів також враховує реакцію споживачів на різноманітні події та акції. Здатність визначати гарячі точки та тенденції в Інтернеті дозволяє компаніям керувати своєю репутацією та використовувати позитивні тенденції для залучення нових клієнтів.

Усі ці аспекти ілюструють глибокий вплив аналітики на облік продажів у різних сферах бізнесу. Це не лише інструмент для збору та обробки інформації, а й важливий інструмент управління, який формулює стратегії, прогнозує тенденції та гарантує ефективне управління бізнесом. Такий комплексний підхід до використання аналізу стає вирішальним для гарантування успіху в постійному ринковому середовищі.

1.3. Використання аналітики в обліку продажів та прийняття управлінських рішень

Використання аналітики в обліку продажів є ключовим елементом сучасного управління бізнесом. Такий комплексний підхід дозволяє не тільки відстежувати фінансові показники, а й отримувати детальну інформацію, яка стає основою для прийняття стратегічних управлінських рішень.

Однією з ключових переваг аналізу в обліку продажів є його здатність аналізувати великі обсяги даних у реальному часі. Це стає основою для швидкого реагування на зміни ринку та попит споживачів. Адреса, оперуючи оновленою інформацією, може ефективно адаптувати стратегії продажів і маркетингу, максимізуючи ефективність бізнесу [3].

Крім того, аналіз обліку продажів враховує не тільки кількісні, а й якісні аспекти. Він дозволяє аналізувати коментарі та реакції клієнтів, визначаючи їхні смаки та антипатії. Це важливо для формування продуктів і послуг, що відповідають реальним потребам споживачів.

Управління через бухгалтерський аналіз продажів включає також аналіз конкурентного середовища. Порівняння ефективності ваших стратегій із тими, що використовуються конкурентами, дозволяє визначити сильні та слабкі сторони та розробити більш конкурентоспроможні стратегії.

Важливою складовою використання аналітики в обліку продажів також є можливість налаштовувати стратегії для різних аудиторій. Аналіз даних дозволяє

визначити унікальні характеристики та потреби різних груп клієнтів, що стає основою для створення персоналізованих підходів для залучення та утримання клієнтів.

Не менш важливим є впровадження аналізу для підтримки цінових стратегій. Аналіз ринкової кон'юнктури та реакції споживачів на різноманітні цінові пропозиції дозволяє ефективно позиціонувати продукти чи послуги, максимізуючи прибуток.

Враховуючи аспекти прийняття управлінських рішень, аналіз обліку продажів впливає на стратегічне планування. Він дає можливість оцінити ефективність попередніх стратегій і адаптувати їх на основі аналітичних даних. Такий циклічний підхід дозволяє компаніям не тільки адаптуватися до змін ринку, а й активно впроваджувати інноваційні підходи.

Інтенсивний розвиток інформаційних технологій і наявність великої кількості даних створюють унікальні можливості для вдосконалення стратегій управління на рівні обліку продажів. Використання аналітики не обмежується статистичним аналізом, але включає інтелектуальні моделі, автоматичне навчання та прогностичні алгоритми. Це дозволяє розробляти більш глибокі стратегії, спрямовані на досягнення конкретних цілей.

Важливим аспектом є здатність аналітики в обліку продажів адаптуватися до змін у бізнес-середовищі. Системи можуть автоматично адаптувати свої моделі та методи аналізу для врахування нових тенденцій і мінливих умов ринку. Це забезпечує актуальність та ефективність стратегій, адаптованих до конкретного моменту та ситуації.

Однією з ключових переваг використання аналізу є можливість інтегрувати дані з різних джерел. Поєднання внутрішньої інформації з даними зовнішніх постачальників і соціальних мереж дозволяє отримати цілісний обсяг інформації про навколишнє середовище і клієнтів. Це стає основою для вичерпного аналізу та глибшого розуміння всіх факторів, що впливають на продажі.

Використання аналітики в управлінні продажами також підкреслює важливість прогностичного аналізу. Здатність передбачати зміни споживчого попиту, ринкових тенденцій і конкурентних стратегій стає невід'ємною частиною стратегічного

планування. Прогнозний аналіз допомагає компаніям приймати зважені рішення та ефективно адаптуватися до нових умов.

Аналітика у сфері управління продажами перетворюється на важливий інструмент для компаній, які прагнуть не лише підтримувати конкурентоспроможність, але й домагатися лідерства у своєму секторі. Однією з основних переваг аналітики є здатність інтегрувати інформацію з різноманітних джерел. Об'єднання внутрішніх даних з інформацією від зовнішніх постачальників та соціальних мереж дозволяє створити всеосяжний огляд ринкового середовища та клієнтської бази.

Така інтеграція дозволяє компаніям отримати не тільки повне уявлення про свій бізнес, але й глибше зрозуміти ринок і конкурентоспроможність. За допомогою аналітики управління продажами може визначити, які фактори впливають на споживачів, і ефективніше спрямовувати їхні зусилля.

Крім того, аналіз підкреслює важливість прогнозного аналізу. Передбачення змін споживчого попиту, ринкових тенденцій і конкурентних стратегій стає важливим для стратегічного планування. Здатність швидко адаптуватися до нових умов стає вирішальною для компанії, яка прагне не тільки вижити, але й процвітати в умовах невизначеності.

Прогнозний аналіз дозволяє компаніям ефективно використовувати дані для прогнозування майбутніх тенденцій і прийняття обґрунтованих рішень. Це особливо важливо в прискореному сучасному світі, де ринкові умови можуть змінитися миттєво.

Використання аналізу в управлінні продажами відкриває можливості для індивідуального підходу до клієнтів. Аналізуючи дані з різних каналів, компанія може зрозуміти індивідуальні потреби своїх клієнтів і запропонувати їм продукти чи послуги, які найкраще відповідають їхнім потребам.

Загалом аналітика стає не лише інструментом для аналізу минулого, а й ключовим фактором стратегічного планування та прийняття рішень в управлінні продажами. Здатність використовувати дані для прогнозування та адаптації дозволяє компаніям залишатися конкурентоспроможними в мінливому бізнес-середовищі.

Крім того, важливо відзначити, що аналіз в управлінні продажами створює можливості для вдосконалення цінових стратегій. Аналіз витрат і ринкова взаємодія дозволяють компаніям узгодити свою цінову стратегію, щоб максимізувати прибуток, залишаючись конкурентоспроможними.

Ще однією важливою перевагою є можливість виявлення та усунення проблем у процесі продажу. Завдяки аналізу даних можна виявити слабкі місця в ланцюжку продажів, що дозволить компанії вдосконалити свої стратегії та покращити якість обслуговування клієнтів [4].

Ураховуючи важливість цифрових технологій та Інтернету речей (IoT), аналітика стає ключовим інструментом для розвитку концепції "інтелектуальних продажів". Збір та обробка даних з IoT-пристроїв дозволяють компаніям надавати індивідуалізовані послуги та швидко адаптуватися до змін у попиті та ринкових умовах у режимі реального часу.

У підсумку, застосування аналітики в управлінні продажами перетворюється на ключовий стратегічний ресурс, що надає компаніям конкурентну перевагу. Аналітика забезпечує глибокий аналіз та прогнозування, роблячи процес управління продажами більш продуктивним та гнучким. Завдяки новітнім технологіям, підприємства здатні не тільки адаптуватися до ринкових змін, але й активно впливати на них, відкриваючи шлях до світлого майбутнього, наповненого конкурентними можливостями.

1.4. Роль інтернет-технологій у вдосконаленні обліку товарів та продаж

У сучасному діловому світі необхідні комплексні стратегії управління продажами. Використання аналітичних методів стає важливим у цьому процесі, оскільки це дозволяє компаніям збирати, обробляти та інтерпретувати величезні обсяги даних для прийняття ефективних управлінських рішень.

Аналітика в обліку продажів включає різноманітні підходи, такі як описова аналітика, яка допомагає зрозуміти минулі та поточні тенденції, і прогнозна аналітика, яка використовує дані для прогнозування майбутніх сценаріїв.

Інтеграція внутрішніх і зовнішніх даних в управлінні бізнесом стає стратегічною необхідністю. Загальний обсяг інформації про ринок і споживачів

дозволяє компаніям ефективно адаптуватися до змін і вдосконалювати стратегії з урахуванням реальних умов.

Використання аналітики також впливає на прийняття управлінських рішень. Детальний аналіз даних дає керівництву можливість не тільки реагувати на поточну ситуацію, але й розробляти стратегічні плани подальшого розвитку бізнесу.

Ключовими перевагами застосування аналітики у сфері управління продажами є здатність компаній швидко адаптуватися до змін у ринкових умовах. Цей метод дозволяє не тільки ефективно реагувати на поточні зміни, але й прогнозувати майбутні тренди, розробляючи стратегії, які відповідають найновішим ринковим вимогам та потребам клієнтів.

Прогресивні організації використовують аналітику для виявлення нових можливостей у розвитку продуктів чи послуг, а також для покращення каналів продажів. Вибірковий аналіз даних допомагає визначити прибуткові сегменти ринку та оптимізувати стратегії продажів для максимального задоволення клієнтів.

Крім того, аналітика в управлінні продажами дозволяє ефективно використовувати ресурси компанії. Оптимізація запасів, прогнозування попиту та управління цінами стають більш доступними завдяки детальному аналізу даних. Це впливає на фінансові результати та економію ресурсів.

Однак впровадження аналітики вимагає ретельного керування конфіденційністю та безпекою даних. Компанії повинні вдосконалювати свої технологічні засоби для захисту конфіденційної інформації, включаючи персональні дані клієнтів.

У сучасному діловому світі використання аналітики в управлінні продажами та прийнятті рішень відіграє ключову роль для стратегічного прогресу компанії. Ефективне застосування аналітики не лише допомагає фірмам розуміти нинішні тренди та рухи на ринку, але й адаптуватися до майбутніх змін, прогнозувати попит і вдосконалювати стратегії продажу. Глибоке осмислення та інтегрування цієї критичної складової в загальну стратегію бізнесу надає суттєву перевагу в конкурентній боротьбі, сприяючи зміцненню та тривалому успіху компанії на світовому ринку.

Ключовими перевагами застосування аналітики в процесі управління продажами є можливість компаній швидко адаптуватися до змін на ринку. Цей метод дозволяє не просто відповідати на поточні зміни, але й прогнозувати майбутні, розробляючи стратегії, які враховують найсвіжіші ринкові тренди та потреби споживачів.

Прогресивні організації використовують аналітику для виявлення нових можливостей у розвитку продуктів чи послуг, а також для покращення каналів продажів. Вибірковий аналіз даних допомагає визначити прибуткові сегменти ринку та оптимізувати стратегії продажів для максимального задоволення клієнтів.

Крім того, аналітика в управлінні продажами дозволяє ефективно використовувати ресурси компанії. Оптимізація запасів, прогнозування попиту та управління цінами стають більш доступними завдяки детальному аналізу даних. Це впливає на фінансові результати та економію ресурсів.

У конкурентному бізнес-середовищі, де інновації та швидкість реагування є ключовими, використання аналітики забезпечує конкурентну перевагу. Компанії можуть швидко реагувати на зміни споживчого попиту, адаптуючи свої стратегії продажів.

Загалом використання аналітики в управлінні продажами визначається як стратегічна необхідність, яка дозволяє компаніям успішно оптимізувати, аналізувати та адаптуватися до сучасних викликів бізнесу.

Виклики та перспективи використання аналітики в управлінні продажами

Не дивлячись на беззаперечні переваги, використання аналітики в управлінні продажами стикається із специфічними викликами. Зокрема, забезпечення безпеки та конфіденційності даних, адекватна підготовка персоналу та вартість впровадження нових технологій. Також існують труднощі у зборі та обробці великого обсягу даних, вимагаючи від компаній високої технічної компетентності.

У міру безперервного прогресу технологій, майбутнє використання аналітики в управлінні продажами виглядає обнадійливо, з обіцянками про глибшу інтеграцію та автоматизацію процесів. Розвиток та застосування технологій, таких як штучний

інтелект та машинне навчання, відкриє нові можливості для аналітичних систем, які зможуть самостійно ідентифікувати ринкові тренди та формувати висновки, що сприятимуть розробці більш ефективних стратегій. Такі інновації не тільки підвищують точність прогнозів, але й забезпечать компаніям більш глибоке розуміння потреб ринку та поведінки споживачів, дозволяючи їм швидко адаптуватися та залишатися на крок попереду конкуренції у швидкозмінному бізнес-середовищі [5].

Всі ці фактори підкреслюють, що використання аналітики в обліку продажів і прийнятті рішень перестає бути просто модним трендом, а стає фундаментальною необхідністю для компаній, які прагнуть стабільного розвитку і зміцнення своїх позицій. змагальний.

Адже успіх використання аналітики в управлінні продажами залежить від того, наскільки ефективно компанія зможе адаптуватися до ринкових умов, що швидко змінюються, і використовувати отримані дані для прийняття обґрунтованих і стратегічно важливих рішень.

Роль інтернет-технологій у вдосконаленні обліку товарів та продаж

Інтернет-технології в сучасному бізнес-середовищі відіграють ключову роль у вдосконаленні процесів обліку майна та продажу. Зростання обсягів даних і швидкості зв'язку створює унікальні можливості для компаній оптимізувати свою діяльність і підвищити ефективність.

Системи електронного обліку товарів дозволяють компаніям точно відстежувати рух товарів від постачальника до кінцевого споживача. Використання систем штучного інтелекту та Інтернету речей дає змогу автоматизувати процеси та уникнути людських помилок у бухгалтерському обліку.

Інтернет-технології розширюють горизонти у сфері продажів. Електронна комерція, онлайн-платформи та цифрові платформи дозволяють компаніям легко виходити на нові ринки та взаємодіяти з клієнтами без географічних обмежень.

Здатність персоналізувати та аналізувати великі обсяги даних дозволяє компаніям ефективно націлювати свої маркетингові зусилля та пропонувати клієнтам

персоналізовані пропозиції, які допомагають збільшити конверсію та задоволеність клієнтів.

Крім того, Інтернет-технології впливають на моделі співпраці та обмін інформацією між торговими партнерами. Хмарні технології забезпечують доступ до даних у режимі реального часу для всіх учасників ланцюга постачання, полегшуючи координацію та планування.

Як наслідок, Інтернет-технології стають необхідною складовою сучасного бізнесу, впливаючи на всі аспекти товарно-збутових та облікових процесів, забезпечуючи компаніям конкурентні переваги та підвищуючи їх адаптивність.

Цифрові інструменти та онлайн-платформи революціонізують сам процес продажів. Електронні касові апарати, мобільні платіжні системи та онлайн-комерція роблять транзакції простішими та швидшими, забезпечуючи зручність як підприємствам, так і клієнтам.

Реальночасова аналітика дозволяє компаніям швидко адаптуватися до змін у споживчому попиті, цінових стратегіях конкурентів та інших ринкових умовах. Застосування аналітичних даних для передбачення та регулювання запасів допомагає запобігти накопиченню непроданих товарів та сприяє більш ефективному управлінню оборотними коштами.

Системи управління відносинами з клієнтами (CRM) інтегрують дані з усіх точок контакту з клієнтом, що дозволяє персоналу забезпечувати більш індивідуалізований підхід та підтримувати довгострокові відносини.

Важливість забезпечення безпеки та конфіденційності інформації стає все більш актуальною з огляду на збільшення обсягу онлайн-транзакцій. Розробка та імплементація ефективних систем кібербезпеки стає ключовим аспектом стратегій компаній, які інтегрують інтернет-технології у свій бізнес-процес.

В майбутньому інтернет-технології продовжать трансформувати методи, якими компанії управляють своїми процесами. Активне впровадження штучного інтелекту, прогрес у розробці блокчейну та зростання застосування віртуальної реальності сприятиме новим інноваціям в області управління запасами та продажами, задаючи нові виміри продуктивності та взаємодії у бізнес-середовищі.

Вплив інтернет-технологій на ринок та продажі

За допомогою інтернет-технологій компанії мають можливість додатково вивчати споживчий попит і адаптувати свою стратегію продажів. Аналіз великих обсягів даних дозволяє визначати та прогнозувати ринкові тенденції, що важливо для розробки ефективних маркетингових кампаній і стратегій продукту.

Платформи онлайн-продажів створюють нові можливості для малого та середнього бізнесу. Компанії можуть легко створювати власні віртуальні магазини, де клієнти можуть переглядати та робити покупки, розширюючи своє глобальне охоплення.

Цифрові платіжні системи спрощують процеси оплати, знижують вартість обслуговування та підвищують зручність для клієнтів. Безпека та надійність цих технологій є важливими факторами, які впливають на рішення споживачів при виборі способу оплати.

Ефективне використання Інтернет-технологій дозволяє компаніям створювати персоналізовані пропозиції та рекламні кампанії, забезпечуючи краще розуміння потреб своїх клієнтів. Це сприяє підвищенню лояльності та покращенню взаємодії зі споживачами.

В умовах інтенсивної конкуренції, інтернет-технології стають не просто фактором конкурентоздатності, але й важливою умовою виживання бізнесу у цифровому світі. Можливості, які вони пропонують, залежать від здатності компаній швидко пристосовуватися до неперервних змін у технологічному середовищі та ефективно використовувати новітні інструменти для свого розвитку та прогресу.

Виклики та майбутнє інтернет-технологій у сфері продажів

З несумнівними перевагами інтернет-технологій при обліку товарів та продажах приходять і виклики. Одним із найважливіших є безпека та конфіденційність даних. Завдяки великому обсягу онлайн-транзакцій, зберігання та обробка особистих даних вимагає вдосконалених систем кіберзахисту, адже порушення безпеки може серйозно зашкодити репутації підприємства.

Майбутнє технологій онлайн-продажів обіцяє ще більше інтеграції та автоматизації. Розвиток штучного інтелекту та машинного навчання сприятиме створенню систем, які не тільки аналізують дані, але й здатні приймати рішення та покращувати стратегії продажів.

Важливим етапом розвитку є також посилення кіберфізичних систем управління ланцюгами поставок. Вони дозволять взаємодіяти та обмінюватися даними між різними ланками цього ланцюга, покращуючи координацію та скорочуючи терміни доставки товарів.

Ці аспекти підкреслюють, що Інтернет-технології не просто оновлюють роботу сучасних компаній, але й задають вектори їхнього майбутнього росту. Надаючи можливості для збільшення зручності, ефективності та інновацій, ці технології стають ключовими не тільки у сферах управління активами та продажів, але й формують новий вигляд економіки та відносин між бізнесами та їх клієнтами.

У висновках слід визначити основні аспекти ролі інтернет-технологій у вдосконаленні обліку товарів та продаж:

1. Ефективність та Зручність: Інтернет-технології дозволяють компаніям оптимізувати свої процеси, забезпечуючи ефективність та зручність у веденні обліку товарів та реалізації.

2. Глобальна Досяжність: онлайн-платформи розширюють ринковий потенціал підприємств, дозволяючи їм досягати нових аудиторій та встановлювати міжнародні партнерські зв'язки.

3. Персоналізація та Аналітика: застосування інтернет-технологій дозволяє підприємствам створювати персоналізовані підходи до клієнтів та використовувати аналітику для стратегічного прийняття рішень.

4. Безпека та Конфіденційність: у відкритому цифровому середовищі, забезпечення безпеки та конфіденційності даних стає критичним аспектом, який вимагає постійного вдосконалення.

5. Майбутні Перспективи: Розвиток інтернет-технологій визначає майбутні тренди, такі як розширення штучного інтелекту, кіберфізичних систем, та більше інтеграції в управління ланцюгом постачання.

Загалом Інтернет-технології трансформують сучасний бізнес, надаючи йому необхідні інструменти для ефективного управління та взаємодії на ринку. Його використання стає стратегічною необхідністю для компаній, які прагнуть досягти стійкого зростання та успішної конкурентоспроможності.

Ефективне управління процесами бухгалтерського обліку та продажів у корпоративному середовищі вимагатиме комплексного міждисциплінарного підходу. Це включає аналіз та інтеграцію різних аспектів, таких як маркетинг, логістика, фінансове планування та обслуговування клієнтів, які разом утворюють ефективну бізнес-стратегію. Головною метою є не лише підвищення прибутковості, але й забезпечення високої задоволеності клієнтів та реалізація стратегій стабільного та довгострокового розвитку компанії. Цей підхід вимагає гнучкості, адаптації до мінливих ринкових умов і постійних інновацій, щоб залишатися конкурентоспроможними та відповідати зростаючим вимогам ринку.

Перш за все, важливим аспектом є ефективне управління запасами. Це включає точний облік усіх товарів, відстеження їхнього руху, аналіз запасів і прогнозування майбутніх потреб у постачанні. Ключовим є баланс між запасами та поточними потребами ринку, щоб уникнути надлишку або дефіциту товарів.

Збут і маркетинг також відіграють важливу роль. Розробка стратегій продажів, які відповідають потребам і очікуванням клієнтів, разом з ефективними маркетинговими кампаніями можуть значно збільшити попит на товари та послуги компанії.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ

2.1. Опис вимог до застосунку

Застосунок з обліку товарів та продажу під час роботи та демонстрації повинен відповідати таким вимогам:

- перегляд усіх товарів від різних постачальників, купування;
- перегляд власних покупок;
- в залежності від ролі користувача додавання товару до магазину, його оновлення, видалення;
- реєстрація та вхід до облікового запису;
- отримання обліку по товару в залежності від ролі користувача;
- в залежності від ролі користувача мати змогу погоджувати замовлення клієнту і відправляти товар.

При аналізі сучасних вимог та аналогів, застосунок повинен мати:

- зручний та прости користувацький інтерфейс;
- застосунок повинен бути розроблений на мові Java, Javascript;
- застосунок повинен відображатися коректно у будь якому браузері.

Кафедра КІТ

НАУ 23 03 04 000 ПЗ

Кафедра КІТ				НАУ 23 03 04 000 ПЗ			
	ПІБ			Літ.	Аркуш	Аркушів	
Розроб.	Гречка К.С.				34	31	РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ ТП-215М - 122
Керівник	Толстікова О.В.						
Н.Контр.	Толстікова О.В.						

2.2. Значення обраних технологій у сучасному розвитку

Сфера розробки програмного забезпечення зазнала значної еволюції, і сучасна розробка додатків стала ключовою зміною парадигми. Цей підхід характеризується гнучкістю, масштабованістю та інтеграцією передових технологій, фундаментально переосмислюючи те, як організації розробляють, розгортають програмні додатки та керують ними.

Історичний контекст і еволюція:

Традиційні та сучасні підходи: протиставлення лінійних, монолітних методологій минулого гнучким, модульним підходам сьогодення.

Драйвери змін: вивчення технологічних, ділових і споживчих тенденцій, які викликали необхідність цього переходу до сучасних методологій розробки додатків.

Теоретичні основи:

Принципи програмної інженерії: обговорення основних теорій і принципів, які інформують про сучасну розробку програм, включаючи модульність програмного забезпечення, теорію систем і гнучкі методології.

Архітектурні шаблони: аналіз архітектурних шаблонів, таких як мікросервіси, вивчення їх походження, еволюції та ролі в сучасних практиках розробки (рис. 2.1).

Feature	Modern Applications	Legacy Applications
Technology	Cloud computing, microservices, containers	Older technologies, monolithic architectures
Development practices	Agile, DevOps, CI/CD	Waterfall, manual testing
Agility	Highly agile, able to change quickly	Less agile, difficult to change
Scalability	Highly scalable, able to handle large volumes of traffic	Less scalable, can struggle to handle large volumes of traffic
Security	More secure, built with security in mind	Less secure, may have security vulnerabilities
Purpose	Used for new business initiatives or to replace legacy applications	Used for critical applications that businesses can't easily replace

Рис. 2.1. Різниця у сучасних та застарілих додатках

Основні особливості сучасної розробки додатків:

Хмарні рідні архітектури: дослідження ролі хмарних обчислень у сучасному розвитку, включаючи концепції масштабованості, стійкості та використання ресурсів на вимогу.

Мікросервіси та сервіс-орієнтована архітектура: детальне дослідження мікросервісів як шаблону проектування, включаючи тематичні дослідження та порівняльний аналіз із монолітними архітектурами.

Безперервна інтеграція та безперервна розробка (CI/CD): обговорення методологій та інструментів для CI/CD та їх впливу на життєвий цикл розробки програмного забезпечення.

Інтеграція DevOps: розробка конвергенції розробки та операцій, а також її трансформаційного впливу на організаційну культуру та процеси розробки програмного забезпечення.

Технологічний ландшафт:

Мови програмування та фреймворки: Вивчення ролі таких мов, як Java та JavaScript, у розробці сучасних додатків і того, як різні фреймворки підтримують ці методології.

Технології баз даних: поглиблений аналіз ролі баз даних, як SQL (наприклад, Postgres), так і NoSQL (наприклад, Mongo), у сучасних екосистемах додатків.

Механізми безпеки: Вивчення інтеграції практик безпеки в розробку додатків, зосереджуючись на таких технологіях, як OAuth і Keycloak для автентифікації та авторизації.

Методологічні міркування:

Шаблони проектування та найкращі практики: обговорення шаблонів проектування, які зазвичай використовуються в сучасній розробці додатків, і найкращі практики (рис. 2.2) їх реалізації.

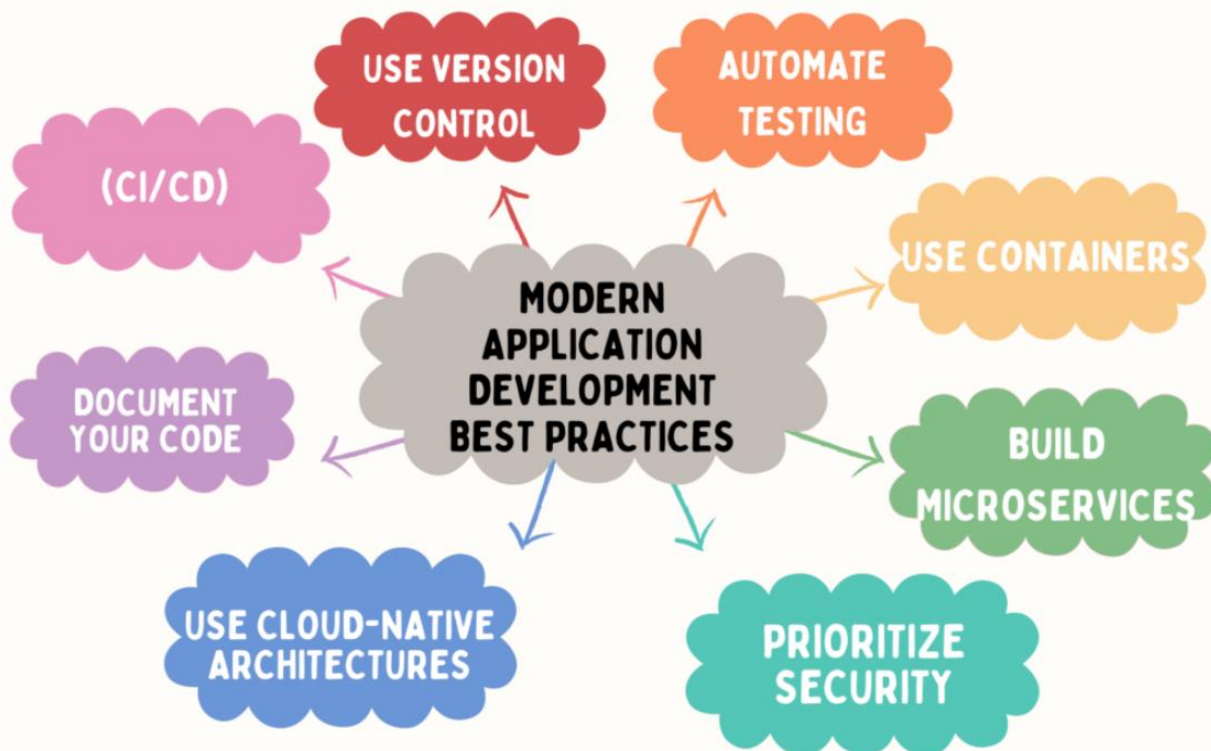


Рис. 2.2. Найкращі практики

Тестування та забезпечення якості: Аналіз ролі тестування в забезпеченні якості сучасних програм, включаючи стратегії автоматизованого тестування та показники якості.

Виклики та майбутні напрямки:

Управління складністю та ресурсами: вирішення проблем, пов'язаних із розробкою сучасних програм, включаючи управління ресурсами та вимоги до навичок.

Нові тенденції: виявлення та обговорення нових тенденцій і технологій, які, ймовірно, вплинуть на майбутнє розробки додатків, таких як ШІ, машинне навчання та безсерверні обчислення.

Парадигма сучасної розробки додатків – це комплексна відповідь на динамічні потреби цифрової ери, наголошуючи на гнучкості, масштабованості та інтеграції передових технологій. Це стратегічний імператив для організацій, які прагнуть залишатися конкурентоспроможними та швидко реагувати на цифровому ринку.

2.3. Фундаментальні поняття

2.3.1. Мікросервіси

Архітектура мікросервісів — це архітектурний стиль, який структурує програму як набір невеликих автономних служб, кожна з яких реалізує окрему бізнес-можливість у обмеженому контексті. Цей підхід різко контрастує з традиційною монолітною архітектурою, де програми тісно інтегровані, а зміни в одній функції часто вимагають змін в інших частинах програми (рис. 2.3).

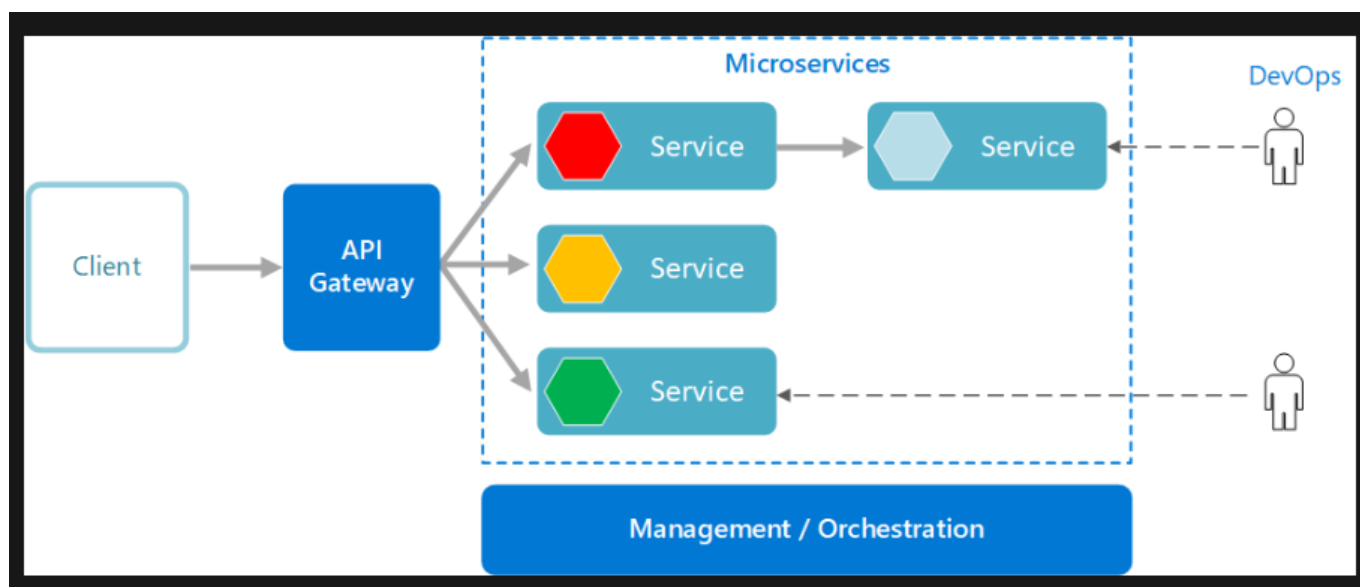


Рис. 2.3. Вигляд мікросервісної архітектури

Мікросервіси — невеликі, незалежні, слабо пов’язані сервіси — розроблені як малі та незалежні, кожен з яких має окрему кодову базу, якою керує невелика команда розробників. Ця незалежність дозволяє розгортати служби незалежно, підвищуючи гнучкість і знижуючи ризик, пов’язаний із розгортанням цілих програм [6-7].

Завдяки поліглотному програмуванню команди можуть вибирати різні технологічні стеки, бібліотеки або фреймворки для кожної служби, яка підтримує різноманітне та адаптоване технологічне середовище.

Зв’язок через API, сервіси спілкуються один з одним за допомогою чітко визначених API, зберігаючи внутрішні реалізації прихованими та розділеними..

Керування/оркестровка та шлюз API (рис. 2.4): такі технології, як Kubernetes, використовуються для оркестрування мікросервісів, керування розміщенням сервісів,

виявлення збоїв і балансування навантаження. Шлюз API служить точкою входу для клієнтів, направляючи запити до відповідних служб і вирішуючи наскрізні проблеми, такі як автентифікація та журналювання.

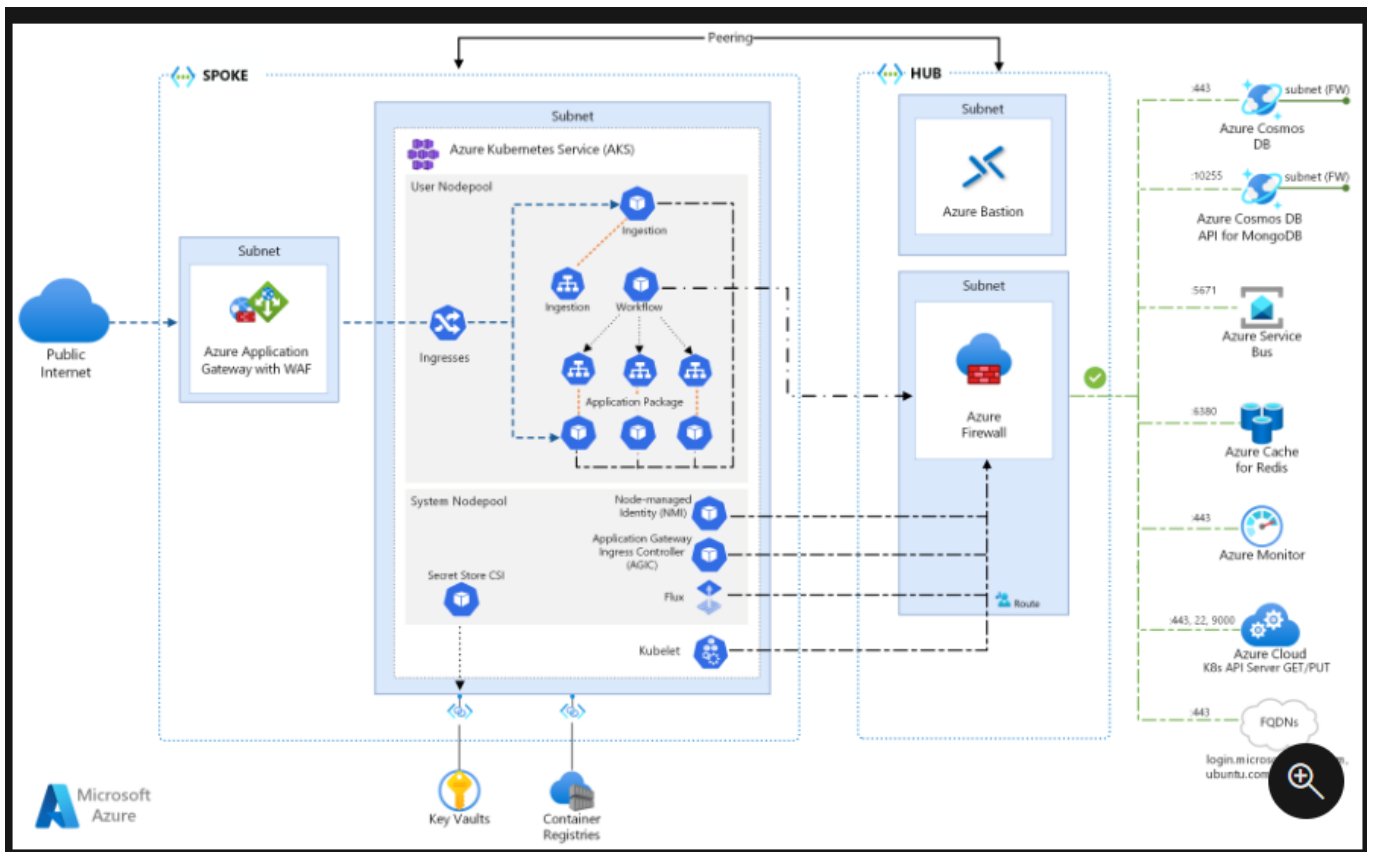


Рис. 2.4. Кубернетіс

Впровадження сервісу Kubernetes вимагає значних знань у сфері DevOps, тому він не реалізований у цій архітектурі. Але для великих застосувань на цю технологію обов'язково варто звернути увагу [8].

Гнучкість і масштабованість: мікросервіси пропонують швидші цикли розгортання, кращу масштабованість і ізоляцію помилок. Якщо одна служба стає недоступною, це не перериває роботу всієї програми. Крім того, сервіси можна масштабувати незалежно, що дозволяє ефективніше використовувати ресурси.

Ізоляція даних: кожен мікросервіс керує власними даними, що спрощує оновлення схем і керування даними.

Складність і розвиток. Додаток мікросервісів є складнішим загалом із проблемами розробки, тестування та залежностями від послуг. Перевантаженість мережі та затримка також можуть бути проблемами через зв'язок між службами.

Управління та контроль версій, децентралізована розробка можуть призвести до появи кількох мов і фреймворків, що ускладнить обслуговування. Контроль версій і зворотна і пряма сумісність також є критичними проблемами [9].

Ці відомості забезпечують повне розуміння архітектури мікросервісів, висвітлюючи її переваги, проблеми та ключові принципи проектування. Ця архітектура стає все більш популярною для створення масштабованих, стійких додатків, які можна розгортати незалежно.

2.3.2. Keycloak

Основні поняття та функції:

Керування областями: у Keycloak область діє як простір імен для керування метаданими та конфігураціями. На основі конкретних вимог можна створити декілька зон. У кожній області Keycloak керує різними об'єктами, такими як клієнти, федерації користувачів, інтеграції LDAP або Active Directory, а також керування користувачами, включаючи користувачів і групи.

Realm Keycloak схожий на простір імен, який дозволяє вам керувати всіма вашими метаданими та налаштуваннями. Ви можете мати кілька доменів залежно від ваших вимог. Загалом, рекомендується уникати використання основної зони, призначеної лише для адміністрування.

На малюнку (рис. 2.5) ви можете побачити інформацію, якою Keycloak дозволяє вам керувати, а саме:

1. Клієнти (за заявку).
2. Управління конфігурацією.
3. Спеціальні теми (інтерфейс користувача).
4. Події.
5. Федерація.

6. Інтеграція LDAP або Active Directory.

7. Керування користувачами (користувачі та групи).

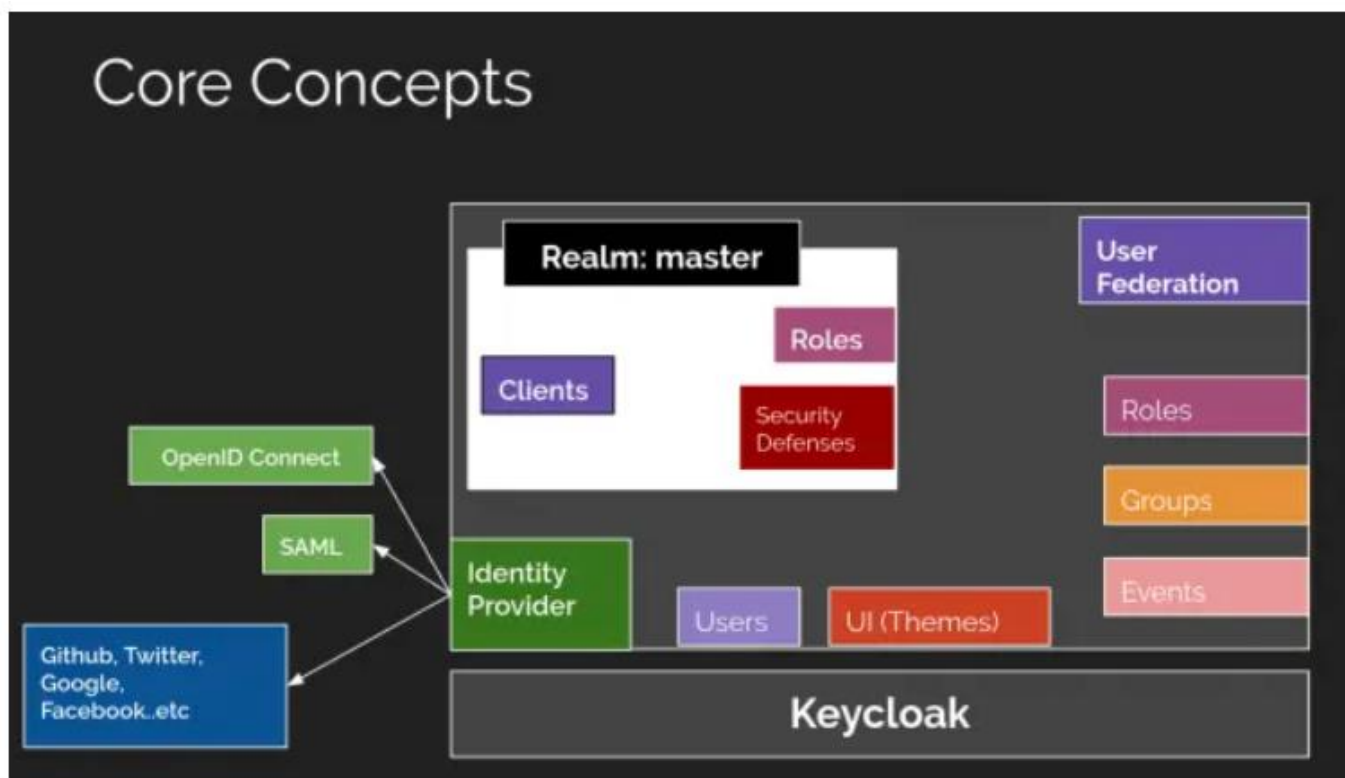


Рис. 2.5. Ключові концепти

Надійність і підтримка протоколів: Keycloak розроблено відповідно до стандартних протоколів безпеки, забезпечуючи рішення динамічного єдиного входу (SSO). Він підтримує такі ключові стандарти, як OAuth 2.0, OpenID Connect і SAML 2.0. Ця підтримка забезпечує інтеграцію з різними інструментами та корпоративними додатками, що робить Keycloak універсальним рішенням для сучасних веб-додатків, API та мікросервісів [10].

Безпека з Keycloak. Кластеризація та автоматичне масштабування: Keycloak підтримує кластеризацію, що дозволяє запускати кілька екземплярів для високої доступності та покращеної продуктивності. Kubernetes Horizontal Pod Autoscaler (HPA) можна використовувати для автоматичного масштабування розгортання Keycloak на основі використання ресурсів, покращуючи його масштабованість і продуктивність [11].

Практики безпеки. Безпека розгортання Keycloak у Kubernetes життєво важлива для захисту програм і даних користувачів. Найкращі практики включають

впровадження мережесих політик і керування доступом на основі ролей (RBAC) для обмеження потоку мережевого трафіку між модулями та безпеці реалізації Keycloak.

Keycloak пропонує потужні можливості контролю доступу з підтримкою керування доступом на основі ролей (RBAC) і детальної політики авторизації. За допомогою RBAC ви можете визначати ролі користувачів і керувати дозволами доступу на основі цих ролей, дозволяючи застосувати точне керування доступом до ваших програм і служб. Детальні можливості авторизації Keycloak дозволяють визначати політики та дозволи на основі ресурсів, надаючи вам ще більше контролю над доступом користувачів до певних ресурсів у вашій програмі.

Keycloak можна інтегрувати з різними технологіями (рис. 2.6), одними з яких може бути Kubernetes.

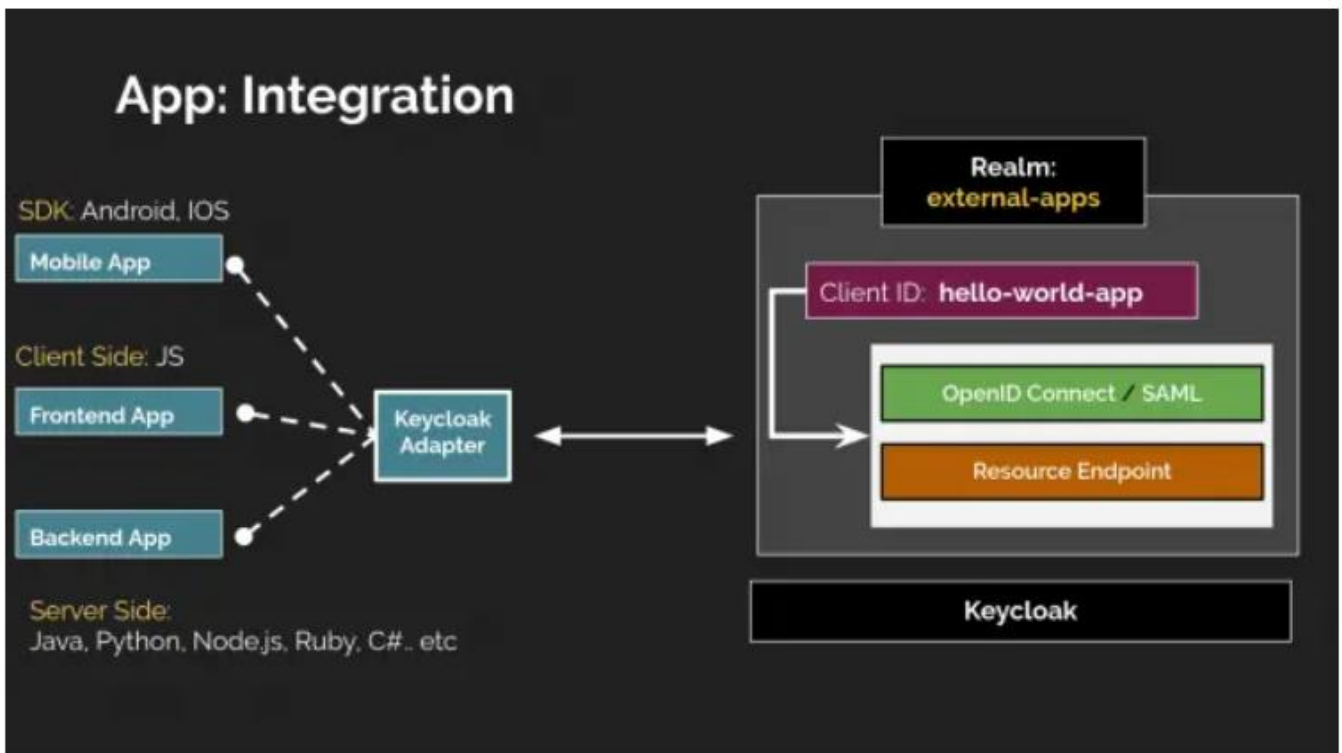


Рис. 2.6. Інтеграція з Keycloak

Keycloak постає як надійне, масштабоване та гнучке рішення IAM, яке особливо підходить для сучасних веб-середовищ. Його сумісність зі стандартними протоколами та легкість інтеграції з Kubernetes робить його оптимальним вибором для захисту веб-додатків, API та мікросервісів. Адаптованість рішення до різних

середовищ розгортання в поєднанні з широким набором функцій робить Keycloak надійним інструментом у сфері керування ідентифікацією та доступом.

2.3.3. OAuth

Протокол OAuth. OAuth, відкритий стандартний протокол авторизації, відіграє важливу роль у сучасній веб-безпеці. Це дозволяє безпечно призначати доступ від однієї служби до іншої без прямого надання облікових даних користувача. Цей протокол покращує моделі доступу на основі дозволів, у яких програми сторонніх розробників запитують доступ до ресурсів, що належать власнику ресурсу, але розміщені на сервері ресурсів. Ключовий сервер авторизації, який успішно пройшов цей процес, перевіряє особу запитувача та видає маркери доступу, які потім застосовуються для доступу до необхідних ресурсів. Ця модель гарантує, що інформація про обліковий запис користувача не буде розголошена, таким чином підвищуючи безпеку онлайн-взаємодій.

Розвиток OAuth відображає змінюючу складність і потреби безпеки цифрових взаємодій. Спочатку задуманий як спрощений спосіб надання користувачам веб-додатків доступу до їх інформації на інших веб-сайтах, OAuth перетворився на надійну структуру, здатну підтримувати широкий спектр програм. OAuth 2.0, найновіший стандарт, пропонує підвищену гнучкість і безпеку, що робить його кращим вибором для реалізації авторизації в сучасних веб-додатках. Цей стандарт визначає різні потоки авторизації (рис. 2.7), кожен з яких адаптований до конкретних сценаріїв, починаючи застосовувати від веб-додатків і мобільних додатків на стороні сервера та клієнта.

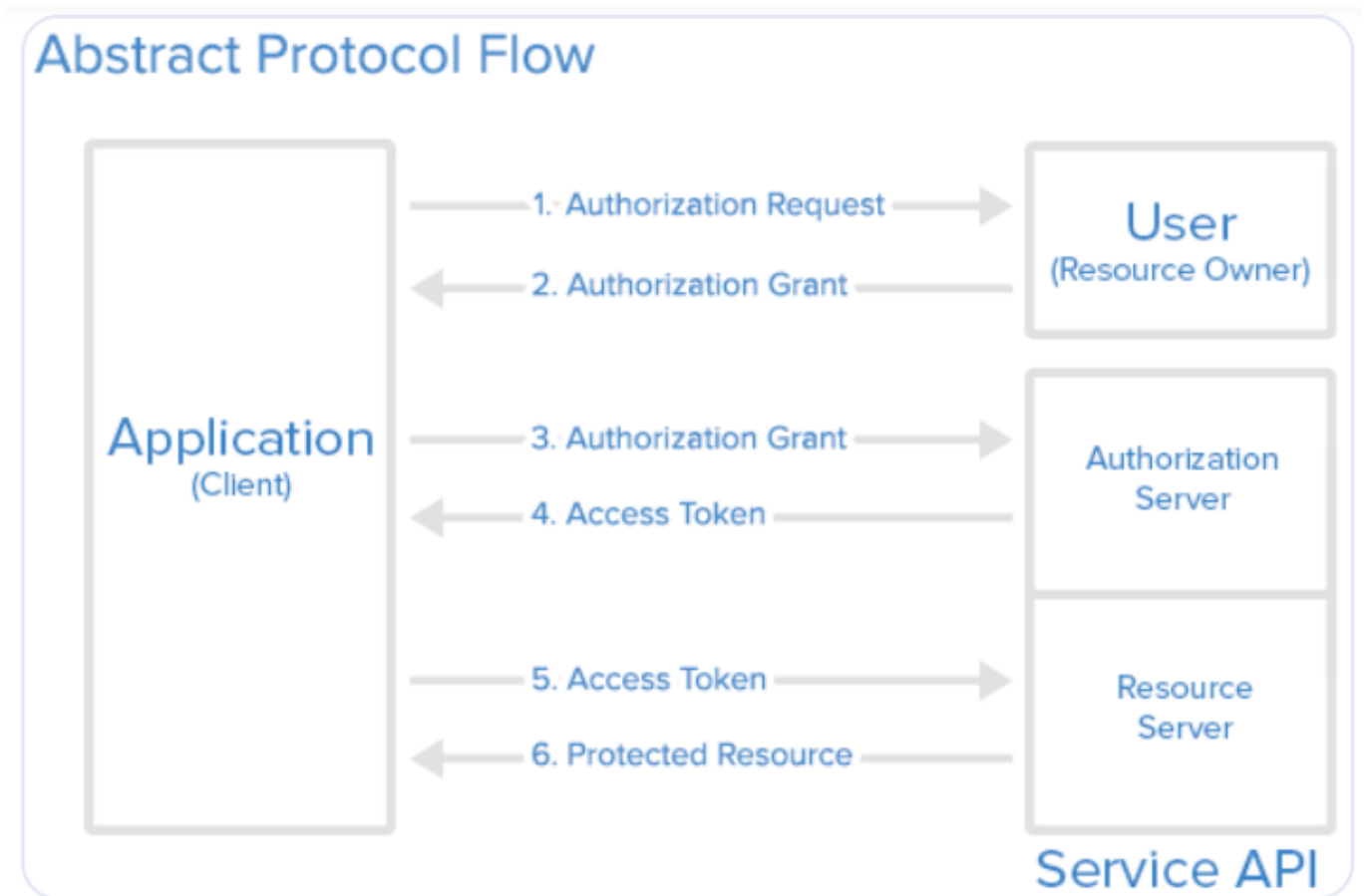


Рис. 2.7. Абстрактний потік авторизації

Методи авторизації та токени доступу. У рамках безпеки, протокол OAuth пропонує декілька методів авторизації, призначених для різних сценаріїв застосування. До них відносяться потік авторизаційного коду для веб-серверних додатків (рис. 2.8), неявний потік для програм, що працюють у браузері або на мобільних пристроях, та потік облікових даних власника ресурсу для додатків, яким довіряють. Центральним елементом у цих методах є тимчасовий токен доступу, який дозволяє доступ до захищених ресурсів без необхідності використання паролів користувачів. Безпечне управління цими токенами є критично важливим, адже вони представляють згоду користувача та дозволи на доступ [12].

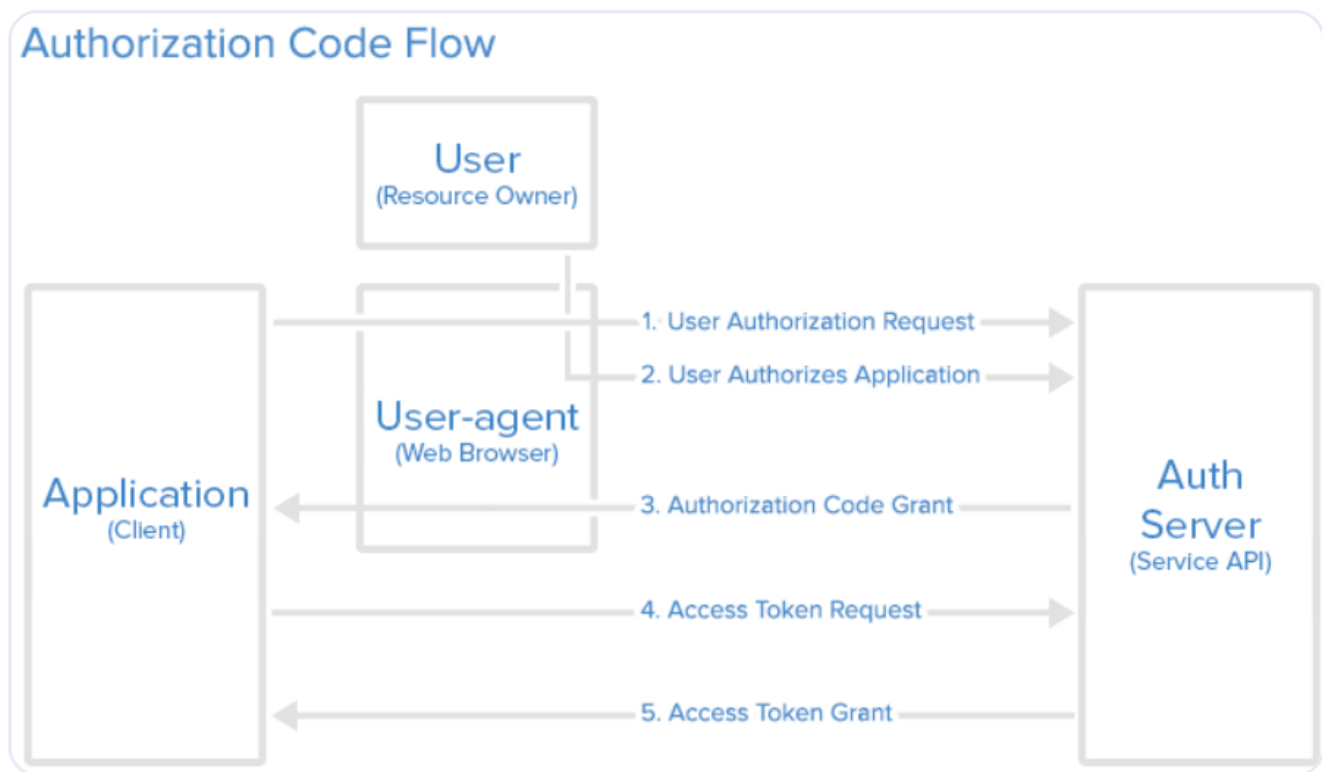


Рис. 2.8. Потік коду авторизації

Безпечний призначений доступ та інтеграція сторонніх розробників

Структура OAuth має вирішальне значення для безпечного керування сторонніми інтеграціями та доступом. Використовуючи OAuth, додатки можуть надавати третім сторонам доступ до певних даних, не відкриваючи повні облікові дані користувача. Цей підхід є основоположним для багатьох сучасних дизайнів програм, де служби взаємопов'язані, а доступність даних користувача має бути збалансована з проблемами конфіденційності та безпеки. Здатність OAuth надавати обмежений доступ, який можна відкликати, робить його ідеальним вибором для таких сценаріїв.

Практичні кроки та передовий досвід. Реалізація OAuth у програмі вимагає глибокого розуміння її різних компонентів і потоків. Процес передбачає налаштування клієнта OAuth, налаштування сервера авторизації та впровадження правильного потоку для потреб програми. Передові практики включають використання HTTPS для всіх взаємодій, належну перевірку маркерів доступу та забезпечення обмеження обсягу доступу, наскільки це необхідно. Регулярні

перевірки безпеки та оновлення відповідно до останніх специфікацій OAuth 2.0 також мають вирішальне значення для підтримки цілісності впровадження.

Загальні випадки використання та приклади. OAuth знаходить свої програми в кількох поширених випадках використання, наприклад, дозволяє користувачам входити на веб-сайт за допомогою своїх облікових записів у соціальних мережах (вхід через соціальні мережі) або дозволяє стороннім програмам отримувати доступ до даних користувача з іншої служби (наприклад, доступ до контактів Google із третьої служби, партійна електронна пошта). Ці варіанти використання підкреслюють універсальність OAuth у забезпеченні безпечного та зручного доступу до веб-ресурсів, що робить його наріжним каменем безпеки сучасних веб-додатків.

2.3.4. PostgreSQL

PostgreSQL (рис. 2.9), база даних з відкритим вихідним кодом, відома своєю стабільністю, підтримує різноманітні функції SQL і доповнює мову SQL функціями для ефективного керування навантаженням даних. Він широко використовується в мобільних, веб-додатках, геопросторових і аналітичних програмах завдяки своїй надійності та відповідності стандартам, включаючи семантику ACID, підтримку зовнішніх ключів, об'єднань, переглядів, тригерів і збережених процедур. PostgreSQL також може похвалитися великою кількістю типів даних, масштабованістю та можливостями динамічного завантаження.

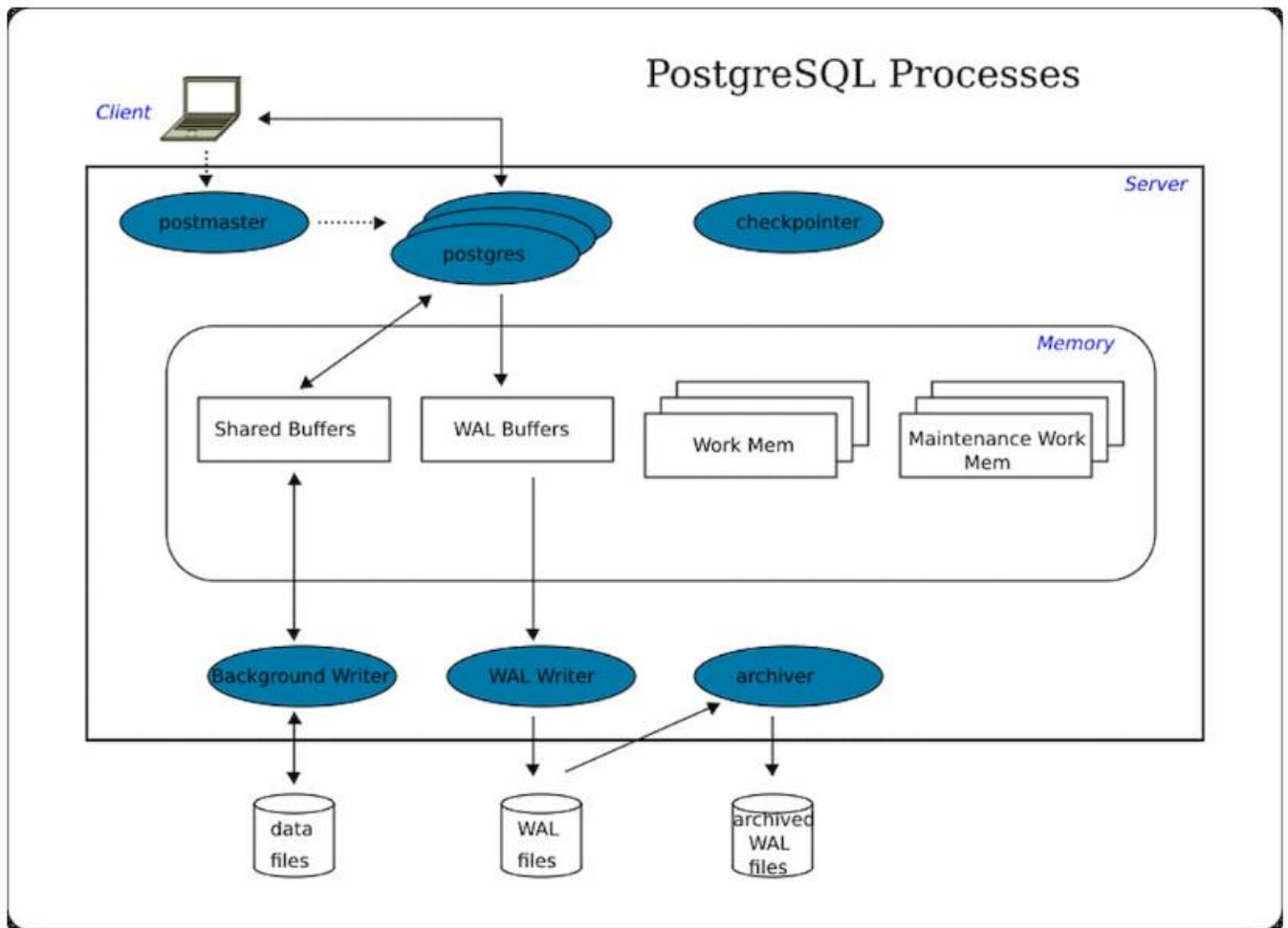


Рис. 2.9. Архітектура PostgreSQL

Продуктивність. Операції запису в PostgreSQL можна виконувати одночасно без необхідності блокування читання/запису. Індеси використовуються для прискорення запитів під час роботи з великими обсягами даних, дозволяючи базам даних знаходити певний рядок без необхідності переглядати всі дані.

За допомогою PostgreSQL ви навіть можете створити індекс виразу, який оперує результатом виразу чи функції, а не просто значенням стовпця. Також підтримується часткова індексація, коли індексується лише частина таблиці. Він також підтримує розпаралелювання запитів на читання, своєчасну компіляцію виразів (JIT) і вкладені транзакції (через точки збереження) для досягнення високої продуктивності та ефективності [13].

PostgreSQL дуже розширюваний, оскільки він заснований на каталозі, що означає, що інформація зберігається в базах даних, стовпцях, таблицях тощо.

PostgreSQL не лише зберігає більший обсяг інформації у своїх каталогах, але також докладну інформацію про типи даних, методи доступу, функції тощо. Ви навіть можете писати свої коди з різних мов програмування без необхідності перекомпілювати базу даних і визначати типи даних.

PostgreSQL проти інших RDBMS. Порівняно з іншими СУБД, такими як MySQL, PostgreSQL виділяється широким набором функцій, надійною підтримкою транзакцій, високою продуктивністю запису та придатністю як для транзакційних баз даних, так і для сховищ даних. Його здатність обробляти дані JSON і геопросторові запити разом із можливостями повнотекстового пошуку відрізняє його від типової RDBMS.

Ключові функції PostgreSQL: розширена підтримка та функції SQL.

1. Успадкування: PostgreSQL дозволяє користувачам створювати таблиці, які чітко моделюють їхні структури даних, підтримуючи успадкування для ефективного керування зв'язками складних даних.

2. Віконні функції: важливі для аналітичних програм, віконні функції в PostgreSQL дозволяють виконувати операції над кількома рядками, на відміну від агрегатних функцій, які повертають лише один рядок.

3. Підтримка даних JSON: підтримка PostgreSQL типів даних і функцій JSON дозволяє обробляти робочі навантаження NoSQL, що робить його гнучким для різних сценаріїв зберігання даних.

Унікальний набір функцій PostgreSQL дозволяє використовувати його як транзакційну базу даних або сховище даних. Чудова підтримка транзакцій і висока продуктивність запису роблять його ідеальним кандидатом для обробки онлайн-транзакцій. У той же час його багаті вбудовані функції, включаючи Windows і аналітичні функції, дозволяють використовувати його як сховище даних, яке аналітики можуть використовувати для отримання інформації.

Типи даних JSON і функції JSON у PostgreSQL надають можливість виконувати роботи NoSQL, використовуючи PostgreSQL. Ще одна унікальна характеристика PostgreSQL - це здатність зберігати та обробляти геопросторові дані, що реалізовано

через плагін ГІС. Функціонал повнотекстового пошуку в PostgreSQL спрощує пошук в документах без потреби у складних регулярних виразах. Крім того, PostgreSQL підтримує ряд розширених можливостей, таких як наслідування схем, стовпці зі складними типами даних, віртуальні таблиці тощо.

Варіанти використання:

1. Урядові та геопросторові дані: за допомогою PostGIS PostgreSQL служить надійним сховищем даних для ГІС і служб на основі розташування.
2. Фінансова галузь: ідеально підходить для OLTP і аналітичних навантажень завдяки сумісності з ACID і швидкій обробці.
3. Наукові дані: ефективно керує великими наборами даних, пропонуючи потужні аналітичні можливості.
4. Веб-технологія: підходить для динамічних веб-сайтів і програм, які є частиною стеку LAPP.
5. Виробництво: підтримує додатки в масштабі Інтернету та оптимізує продуктивність ланцюга поставок.
6. Поєднання цих функцій робить PostgreSQL універсальною та потужною системою баз даних, здатною обробляти широкий спектр додатків, від складних веб-сервісів до великомасштабних аналітичних робочих навантажень.

2.3.4. MongoDB

Характеристики та засади MongoDB, провідної NoSQL бази даних, включають її здатність до масштабування, високої ефективності та постійного доступу. Ця система помітно відрізняється від стандартних реляційних систем управління базами даних (RDBMS) завдяки ряду унікальних характеристик. Одна з ключових особливостей MongoDB - це її здатність обробляти великі маси даних у розподіленій мережі. MongoDB не використовує фіксованих схем, що дозволяє вставляти дані без попередньої визначеності структури. Ця гнучкість полегшує зберігання комплексних і різноманітних даних у порівнянні з більш жорсткими та структурованими вимогами RDBMS. Концепція розробки MongoDB зорієнтована на створення надійного та легко

масштабованого рішення, яке може обробляти різні формати даних та складні операції. Це робить її ідеальною для застосувань, пов'язаних із великими даними та аналітикою в реальному часі.

MongoDB була створена з акцентом на продуктивність та гнучкість для розробників. Як документо-орієнтована база даних, вона зберігає дані у форматі документів, які у свою чергу згруповані в колекції. Робота з документними моделями є більш інтуїтивно зрозумілою для розробників, адже документи є самодостатніми і можуть бути представлені як об'єкти. Це дозволяє розробникам зосереджуватися на збереженні та обробці потрібних даних без необхідності перейматися розподілом даних по різних таблицях.

Документи об'єднані в колекції. Однак документи в одній колекції не обов'язково повинні мати однаковий набір полів. Це те, що ми називаємо «гнучкою схемою». Ця гнучкість дозволяє розробникам швидше виконувати ітерації та переміщувати дані між різними схемами без будь-яких простоїв. Однак, якщо ви хочете заблокувати свою схему в певний момент, ви можете зробити це, застосувавши правила перевірки до своїх колекцій [14].

Структура та гнучкість документа JSON (рис. 2.10). MongoDB зберігає дані у форматі BSON (Binary JSON), який є серіалізацією документів типу JSON у двійковому кодуванні. Ця модель документа дозволяє MongoDB зберігати складні ієрархічні зв'язки, масиви та інші вкладені структури в одному записі. Це дозволяє базі даних більш природно узгоджуватися з об'єктно-орієнтованими мовами програмування, спрощуючи процес розробки. Гнучкість, яку пропонує ця структура документа, схожа на JSON, дозволяє легко змінювати схему даних без простою, що є значною перевагою в гнучких середовищах розробки, де вимоги можуть швидко змінюватися.

```

{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}

```

The diagram shows a JSON document with four fields: `name`, `age`, `status`, and `groups`. Each field is pointed to by an arrow labeled "field: value".

Рис. 2.10. Вигляд документа

Зрештою, оптимальне балансування навантаження залишається одним із святих граалів керування великомасштабними базами даних для зростаючих корпоративних програм. Правильний розподіл мільйонів клієнтських запитів між сотнями або тисячами серверів може зробити помітну (і дуже цінну) різницю в продуктивності.

На щастя, завдяки функціям горизонтального масштабування, таким як реплікація та шардинг, MongoDB підтримує масштабне балансування навантаження. Платформа може обробляти кілька одночасних запитів на читання та запис тих самих даних, використовуючи найкращі в своєму класі протоколи блокування та керування паралельним доступом, які забезпечують узгодженість даних. Немає необхідності додавати зовнішній балансувальник навантаження: MongoDB гарантує, що кожен користувач має узгоджене представлення та якість даних, до яких йому потрібно отримати доступ.

Горизонтальне масштабування та розподіл даних. MongoDB розроблено для горизонтальної масштабованості, яка передбачає розподіл даних між декількома машинами. Цей підхід відрізняється від вертикального масштабування, яке вимагає додавання більшої потужності до однієї машини. Горизонтальне масштабування або сегментування в MongoDB дозволяє обробляти дуже великі набори даних і обсяги транзакцій, розподіляючи дані між кількома серверами. Такий розподіл допомагає збалансувати навантаження та підтримувати високу продуктивність навіть при великому зростанні даних.

Комплекти фрагментації та репліки. Розбиття на фрагменти в MongoDB передбачає розбиття даних на менші, більш керовані частини, які називаються «фрагментами». Кожен шард є окремою базою даних, і разом вони утворюють всю

базу даних. Ця конфігурація дозволяє розподіляти дані між кількома машинами, забезпечуючи таким чином масштабоване середовище. На додаток до шардингу, MongoDB використовує набори реплік для забезпечення надмірності та високої доступності. Набір реплік — це група серверів MongoDB, які підтримують однаковий набір даних, забезпечуючи надлишковість даних і високу доступність. Ця функція забезпечує продуктивність системи навіть у разі відмови апаратного забезпечення.

Налаштування MongoDB передбачає інсталяцію сервера MongoDB на головній машині та його конфігурацію відповідно до конкретних вимог програми. Управління MongoDB включає такі завдання, як моніторинг її продуктивності, масштабування бази даних шляхом додавання або видалення вузлів, а також забезпечення цілісності та безпеки даних. MongoDB пропонує кілька інструментів і послуг для полегшення цих завдань керування, включаючи хмарну службу MongoDB Atlas, яка забезпечує повністю керований досвід MongoDB.

Випадки та приклади реального використання. MongoDB використовується в широкому спектрі додатків, від простих веб-додатків до складних великомасштабних систем. Серед поширених випадків використання:

1. Управління вмістом і доставка: модель гнучкої схеми MongoDB ідеально підходить для систем керування вмістом (CMS), де типи вмісту можуть змінюватися з часом.

2. Мобільні програми: структура документа JSON добре відповідає вимогам мобільних програм до даних, забезпечуючи безперебійну синхронізацію даних між програмою та базою даних.

3. Аналітика в реальному часі: здатність MongoDB обробляти великі обсяги даних і ефективні функції індексування роблять її придатною для аналітичних програм у реальному часі.

4. Програми IoT: MongoDB може обробляти різноманітні дані та об'єми, створені пристроями IoT, що робить його популярним вибором для рішень для керування даними IoT.

2.3.5. LDAP

Вивчення LDAP, огляд протоколу та використання. LDAP (Lightweight Directory Access Protocol) є основним протоколом для керування користувачами, атрибутами та автентифікацією в організаціях протягом майже трьох десятиліть. Це дозволяє безпечно керувати користувачами та ІТ-ресурсами в каталозі, контролюючи доступ через комп'ютерну мережу. Розроблений у 1993 році протокол LDAP розвинувся на основі протоколу служб каталогів X.500 для надання ефективних служб каталогів зі зменшеними накладними витратами та використанням пропускну здатності.

Структура каталогу LDAP та ієрархія даних (рис. 2.11). LDAP використовує ієрархічну структуру дерева, відому як інформаційне дерево каталогу (DIT). DIT складається з записів (користувачі, групи, принтери, сервери, програми тощо), кожна з яких має атрибути, що їх описують. Структура включає один корінь, гілки та листи, де корінь охоплює весь каталог, а листи представляють окремі сутності без підлеглих.

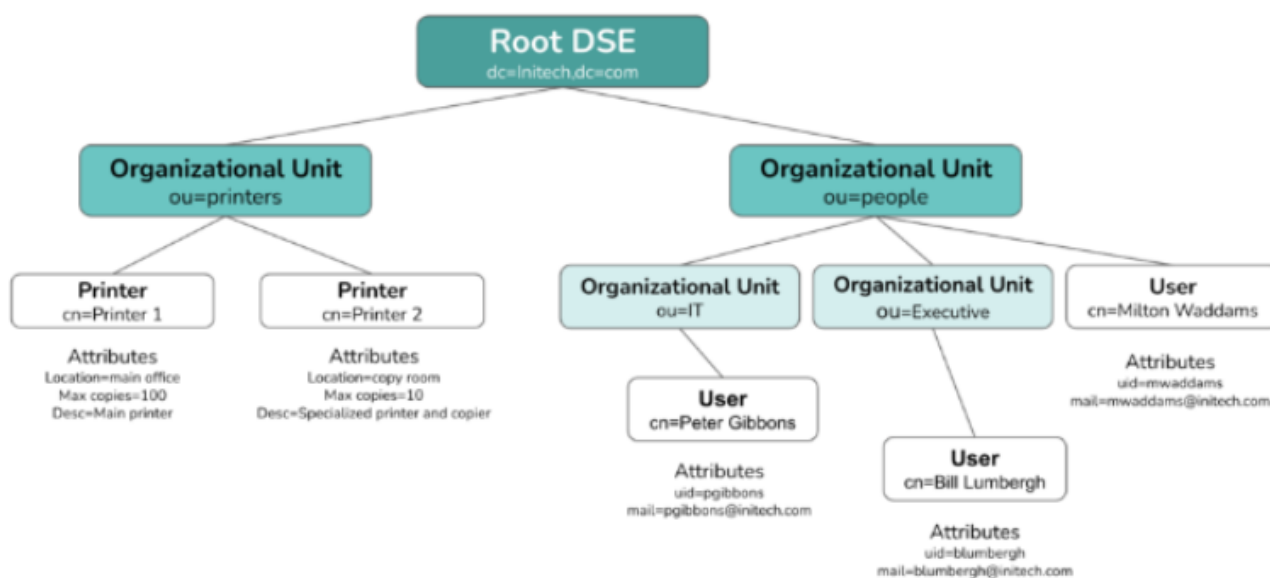


Рис. 2.11. Спрощена структура directory information tree

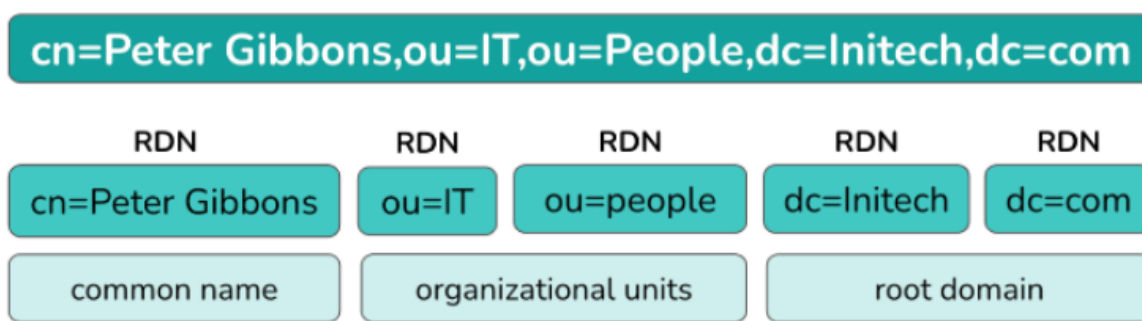
DN: Distinguished Name

Це унікальний ідентифікатор для запису LDAP, який визначає всі атрибути, призначені об'єкту. Він може містити кілька точок даних і містити відносні розпізнавальні імена (RDN) у рядку, розділеному комами (рис. 2.12).

Формат DN працює як географічні координати, тільки навпаки: він визначає розташування каталогу шляхом переліку кожного підрозділу в порядку зростання.

У той час як географічні координати починаються із загальних і стають більш конкретними, DN починається з назви об'єкта й працює до основного компонента каталогу — наприклад, домену вашого сервера LDAP — утворюючи повну адресу, що вказує на розташування запису в дереві LDAP.

```
cn=Peter Gibbons,ou=IT,ou=People,dc=Initech,dc=com
```



For Peter Gibbons, a programmer in the IT department at Initech, the DN may be formatted as `cn=Peter Gibbons,ou=IT,ou=People,dc=Initech,dc=com`

Рис. 2.12. Приклад з distinguished name

Основні функції LDAP включають оновлення, запити та автентифікацію інформації каталогу. Автентифікація виконується за допомогою операції прив'язки, що дозволяє користувачеві спілкуватися з каталогом LDAP. Після автентифікації користувачі отримують доступ до призначених їм ресурсів на основі їхніх груп і ролей у каталозі [15].

Налаштування LDAP для різних середовищ. LDAP є універсальним і може бути налаштований для різних середовищ, включаючи технічні програми, серверну інфраструктуру, файлові сервери та мережеве обладнання. Його гнучкість робить його придатним для різноманітних ІТ-ресурсів, від локальних до хмарних систем, таких як AWS.

LDAP використовується в різних секторах, від уряду та охорони здоров'я до освіти та фінансів. Передові практики включають розуміння та ефективне використання його ієрархічної структури, оптимізацію дизайну схеми та забезпечення надійних заходів безпеки в процесах автентифікації та авторизації.

2.3.6. Java

Java, розроблена компанією Sun Microsystems у середині 1990-х років, зазнала значної еволюції з моменту свого створення. Спочатку розроблена для інтерактивного телебачення, Java знайшла своє місце в широкому світі розробки програмного забезпечення завдяки своїй платформи-незалежній природі, якій сприяє віртуальна машина Java (JVM). Протягом багатьох років Java розвивалася завдяки численним оновленням, додаючи такі функції, як лямбда-вирази, Stream API, а нещодавно — модульність за допомогою Project Jigsaw у Java 9[16].

Ключові характеристики та екосистема. Основні функції Java включають:

1. Незалежність від платформи: філософія Java «напишіть один раз, запустіть будь-де» ввімкнена JVM.

2. Об'єктно-орієнтоване програмування: Java в основному об'єктно-орієнтована, що сприяє повторному використанню коду та модульності.

3. Багаті API: Java надає великі бібліотеки та API для різних завдань, від роботи в мережі до аналізу XML.

4. Надійне керування пам'яттю: автоматичне збирання сміття допомагає ефективно керувати пам'яттю.

5. Можливості багатопоточності: модель потоків Java дозволяє паралельне програмування.

6. Велика спільнота та екосистема: Java має величезну екосистему, включаючи фреймворки, інструменти та велику спільноту розробників.

Java в мікросервісах і хмарних програмах. Java Frameworks для мікросервісів (наприклад, Spring Boot)

Такі фреймворки, як Spring Boot, стали синонімом Java у контексті мікросервісів. Вони пропонують такі функції, як:

1. Автоматична конфігурація: спрощення налаштування та налаштування.
2. Автономні програми: легко створювати мікросервіси як незалежні програми.
3. Підтримка вбудованих серверів: підтримка вбудованих Tomcat, Jetty або Undertow без необхідності окремого розгортання.
4. Інтеграція екосистеми Spring: плавна інтеграція з іншими проектами Spring.

Порівняння фреймворків (рис. 2.13):

Spring Boot	Oracle Helidon	AxonIQ	DropWizard	Quarkus
Simple data management	Cloud-native tools	Tiered storage	Fast bootstrap	Container first
Fault tolerance options	Supports gRPC	Disaster recovery	Detailed metrics	Imperative & reactive programming
Distributed messaging	Tracing capabilities	Append-only	High productivity	Kubernetes
Various capabilities	Robust security	Robust security	Multiple integrations	Simple development

Рис. 2.13. Порівняння різних Java фреймворків

Найкращі методи масштабованості та продуктивності. Щоб забезпечити масштабованість і продуктивність мікросервісів на основі Java:

1. Оптимізація параметрів JVM: налаштування параметрів JVM для оптимальної продуктивності.

2. Ефективна взаємодія з базою даних: використання пулу з'єднань і оптимізованих запитів.

3. Асинхронна обробка: використання можливостей одночасної обробки Java.

4. Шаблони проектування мікросервісів: впровадження шаблонів, таких як автоматичні вимикачі, для стійкості.

Реальні приклади додатків Java. Java широко використовується в різних галузях промисловості. Приклади включають фінансові послуги для обробки транзакцій, платформи електронної комерції для серверних служб і розробку програм для Android[17].

Так як при розробці використовується Spring Boot, то опишемо коротко про нього.

Spring Boot — це фреймворк із відкритим вихідним кодом на основі Java, який використовується для створення автономних програм Spring продуктивного класу з мінімальними зусиллями. Це спрощує завантаження та розробку нових додатків Spring, надаючи різні функції та зручності.

Компоненти Spring Boot (рис. 2.14)



Spring Framework Runtime

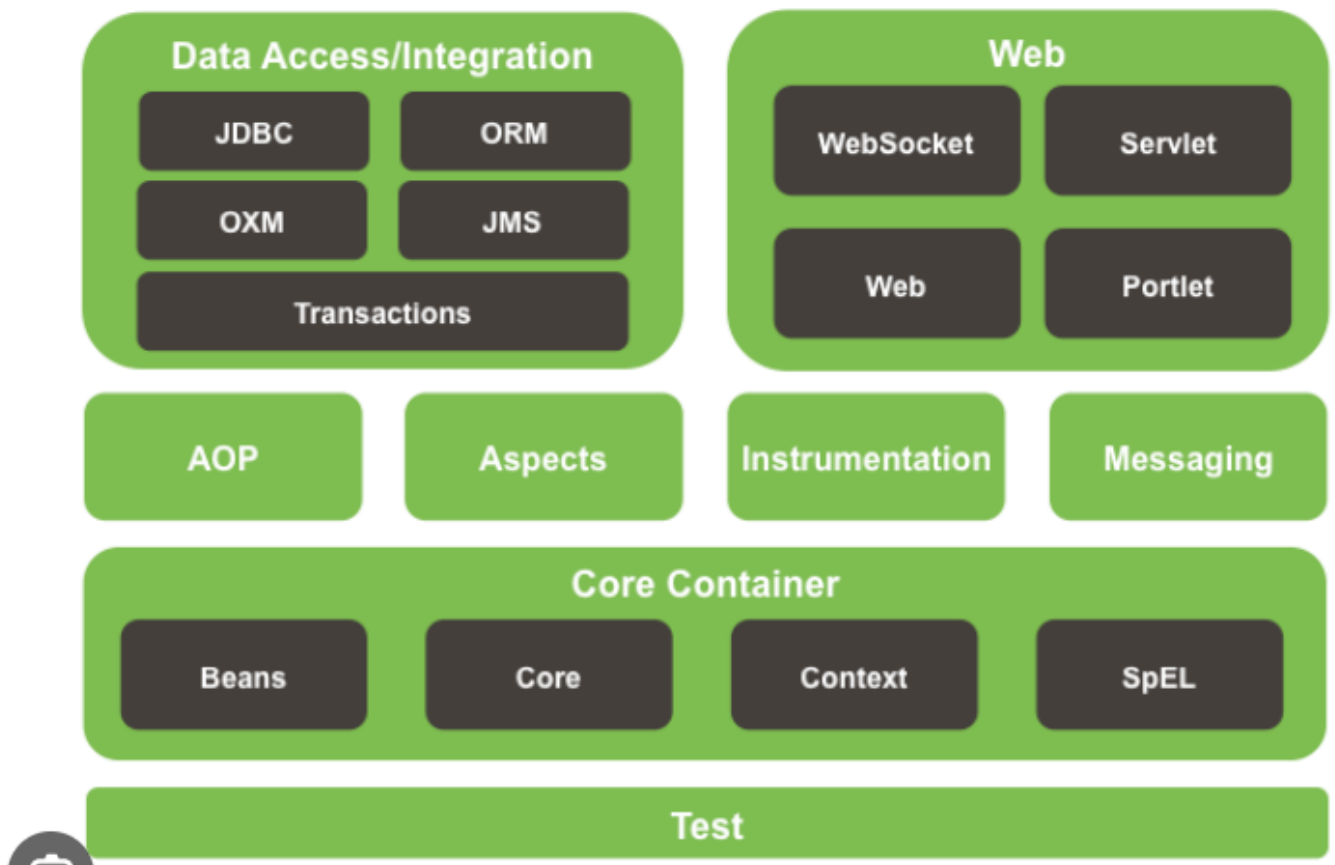


Рис. 2.14. Компоненти Spring Boot

Ось деякі ключові аспекти Spring Boot:

1. Автоматична конфігурація: Spring Boot автоматично налаштовує вашу програму на основі доданих вами залежностей jar. Наприклад, якщо у вашому проекті є залежність `spring-boot-starter-web`, Spring Boot автоматично налаштовує вашу програму як веб-програму.

2. Автономність: програми Spring Boot є автономними, тобто їм не потрібен зовнішній сервер. Вбудовані сервери, такі як Tomcat, Jetty або Undertow, можна автоматично налаштувати та запустити програму.

3. Opinionated Defaults: Spring Boot надає «початкові» залежності для спрощення конфігурацій збірки. Ці запуски є набором зручних дескрипторів залежностей, які ви можете включити у свою програму.

4. Інтеграція екосистеми Spring: він легко інтегрується з іншими проектами Spring, такими як Spring Data, Spring Security і Spring Cloud, пропонуючи послідовний досвід розробки.

5. Готовий до виробництва: Spring Boot пропонує такі функції, як показники, перевірки працездатності та зовнішню конфігурацію, щоб зробити його придатним для робочих середовищ.

6. Відсутність генерації коду та відсутність вимог щодо конфігурації XML: фреймворк мінімізує використання конфігурації XML і не потребує генерації коду, сприяючи підходу конфігурації на основі коду.

2.3.7. JavaScript

Еволюція та сучасні тенденції. JavaScript, спочатку створений Бренданом Айхом у 1995 році, значно розвинувся завдяки процесу стандартизації через ECMAScript (ES). Ключові функції, представлені в різних версіях ECMAScript, включають шаблонні літерали, класи, обіцянки, `async/await` тощо. Найновіша версія, ES2022, принесла функції, що покращують можливості JavaScript і зручність для розробників. TypeScript, наднабір JavaScript, розроблений Microsoft, набув популярності завдяки статичному введенню та виявленню помилок, що призвело до більш надійного коду. Зараз TypeScript широко використовується в таких фреймворках, як Angular, React і Vue.js.[18].

Екосистема JavaScript: фреймворки та інструменти. Сучасні фреймворки та бібліотеки JavaScript стали важливими у веб-розробці, пропонуючи попередньо написаний код, компоненти та багаторазові утиліти. Відомі фреймворки включають React, Vue.js, Svelte, Next.js, Nuxt.js, Gatsby та Solidjs, кожен з яких надає унікальні функції та переваги для різних вимог проекту. Менеджери пакетів, такі як NPM і Yarn, спрощують адаптацію та керування цими фреймворками та бібліотеками.

Хоча JavaScript також можна використовувати для бекенд-розробки. У цьому випадку ми будемо використовувати його для розробки інтерфейсної системи.

Розробка повного стеку за допомогою JavaScript (Node.js, React, Angular). Спочатку обмежуючись інтерфейсною розробкою, JavaScript розширився до внутрішньої частини за допомогою Node.js. Це середовище виконання, створене на движку JavaScript V8 Chrome, забезпечує високопродуктивні масштабовані програми. Він поєднав інтерфейсну та бек-енд розробку в одну мову, спрощуючи процес розробки. Можливість розробити повний стек як у зовнішньому, так і у внутрішньому JavaScript спрощує робочий процес і допомагає досягти стабільних результатів проекту.

Інтеграція з мікросервісами та API. Node.js надзвичайно спрощує інтеграцію JavaScript із мікросервісами та API, дозволяючи безперешкодно працювати JavaScript у серверних середовищах. Ця інтеграція має вирішальне значення для сучасної веб-розробки, особливо для створення масштабованих і ефективних веб-додатків.

Різні тематичні дослідження підкреслюють універсальність і можливості JavaScript у різних сценаріях:

1. JRPass: оптимізація системи бронювання для японської залізничної мережі для збільшення продажів.

2. CustomEase: Створення хмарної програми для компанії, що займається дизайном інтер'єрів, із спеціальними функціями CRM, CMS, ERP і командного спілкування.

3. AMLD: розробка веб-програми для керування подією зі штучним інтелектом із багатьма доповідачами та відвідувачами.

4. Mediatask: Створення ринку для креслярів та архітекторів з інтелектуальними алгоритмами та рішеннями для автоматизації бізнесу.

5. TRAVELDUCK: Створення цифрового ринку для бутикових пригодницьких подорожей і заходів.

6. SnowShow: розробка системи бронювання для великого європейського оператора зимових турів за допомогою рішень Machine Learning.

Зауваження щодо продуктивності та масштабованості. Підвищення продуктивності JavaScript включає різні передові практики, такі як використання

HTTP/2, скорочення HTML, оптимізація змін DOM, стиснення файлів, мінімізація коду, кешування та ефективна обробка подій. Такі інструменти, як Google Closure Compiler, YUI Compressor і UglifyJS, допомагають оптимізувати роботу JavaScript шляхом стиснення та мінімізації коду.

При розробці використовувався Vue фреймворк, тому опишемо деяку частину про нього.

Vue.js — популярний фреймворк JavaScript для створення інтерфейсів користувача. Він відмінно підходить для створення інтерактивних і динамічних веб-додатків. Vue.js пропонує прогресивний підхід до розробки, що дозволяє легко інтегрувати його в існуючі проекти. Його основна бібліотека зосереджена на рівні перегляду, що дозволяє розробникам легко інтегрувати його з іншими бібліотеками або існуючими проектами. Vue.js відомий своєю простотою та гнучкістю, що робить його придатним як для початківців, так і для досвідчених розробників. Завдяки активній спільноті та обширній документації Vue.js є потужним інструментом для створення сучасних веб-додатків.

Vue.js працює шляхом створення реактивних компонентів (рис. 2.15), які представляють різні частини інтерфейсу користувача веб-додатку. Ці компоненти можна багаторазово використовувати та можуть містити шаблони HTML, логіку JavaScript і стилі CSS. Vue.js використовує віртуальну DOM для ефективного оновлення фактичної DOM, забезпечуючи швидке й ефективне відтворення змін. Він також забезпечує надійну систему зв'язування даних, що забезпечує автоматичну синхронізацію між моделлю та поданням.

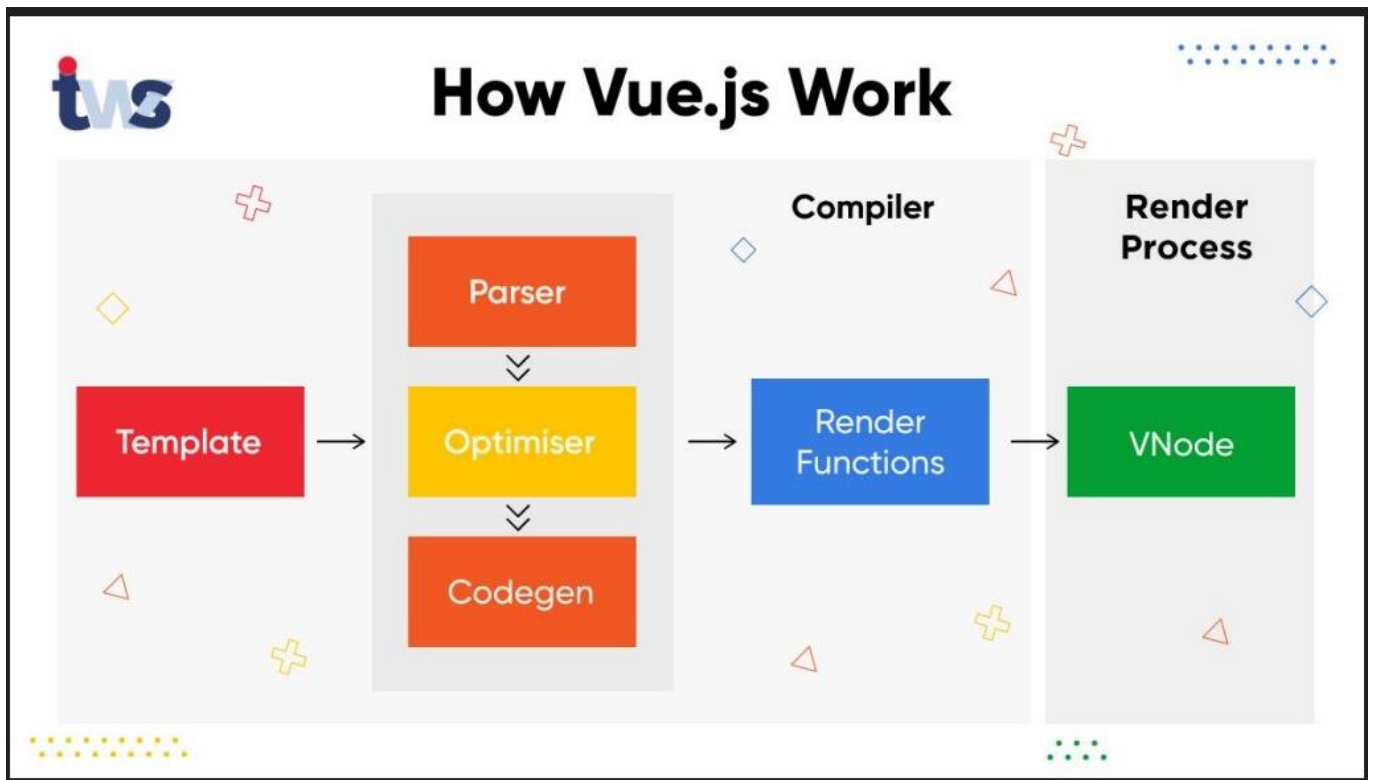


Рис. 2.15. Як працює Vue фреймворк

Vue.js підтримує директиви, користувацькі директиви та перехоплювачі життєвого циклу, надаючи розробникам детальний контроль над своїми програмами. Загалом Vue.js спрощує розробку інтерфейсу, дозволяючи легко створювати адаптивні та інтерактивні веб-додатки.

2.3.8. Gateway

Роль і функції: шлюз діє як проміжний рівень між клієнтами та набором мікросервісів. Він забезпечує єдину точку входу для всіх запитів клієнтів у систему, спрощуючи керування багатьма службами. Це включає керування та налаштування мережеских правил і аспектів безпеки, які є особливо складними в розподіленій системі з мікросервісами, розповсюдженими на кількох кластерах і хмарах (рис. 2.16).

Шлюз в архітектурі мікросервісів.

Функції: шлюз API, який виконує роль сервера або проксі-сервера L7, функціонує як централізована точка для керування викликами API, спрямовуючи їх до правильного мікросервісу. Він інкапсулює складність основної системи, виконує

перевірки безпеки, такі як автентифікація та авторизація, перш ніж трафік досягне служби, і дозволяє службам зосередитися на своїх основних функціях [19].

Проблеми, які вирішуються: шлюзи вирішують різні проблеми, такі як виявлення служб і маршрутизація трафіку, проблеми безпеки через загальнодоступні кінцеві точки служб, обробка різноманітних протоколів для взаємодії та проблеми затримки, спричинені прямим зв'язком між клієнтом і мікросервісом.

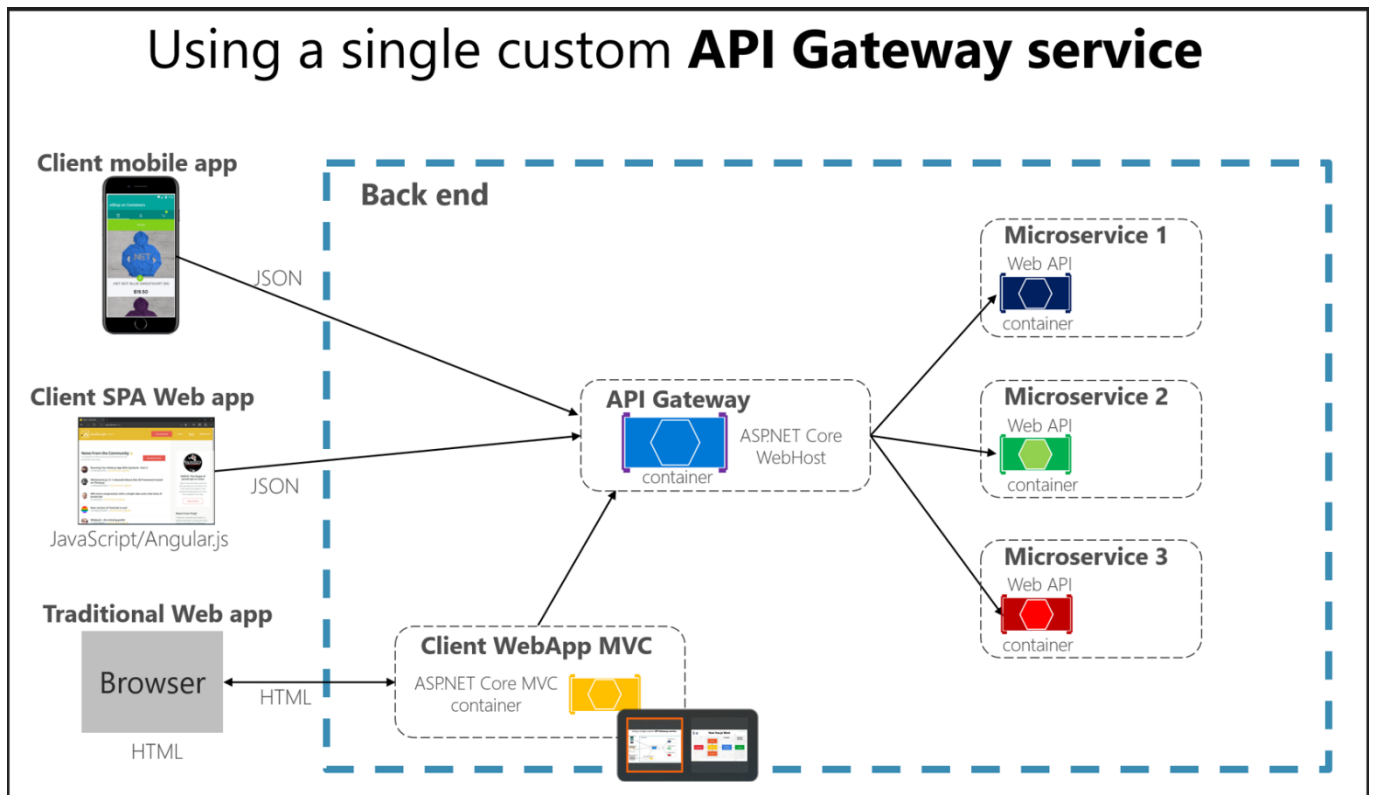


Рис. 2.16. Робота gateway на прикладі

Найкращі практики впровадження шлюзу. Ключові функції: важливі функції шлюзів API включають перевірку запитів, автентифікацію та авторизацію, виявлення служб, трансляцію протоколів і динамічну маршрутизацію.

Розширені функції: пропонують розширену маршрутизацію трафіку, обмеження швидкості, обробку помилок, розрив ланцюга, відстеження, кешування та завершення SSL.

Переваги: покращена безпека додатків, більша гнучкість в управлінні та масштабуванні мікросервісів, кращі можливості монетизації для постачальників API та покращений досвід роботи з користувачем.

Приклади ефективного використання шлюзу. Найпопулярніші інструменти з відкритим кодом: Tyk API Gateway, Kong API Gateway і KrakenD API Gateway є добре відомими інструментами з відкритим кодом, кожен з яких пропонує унікальні функції, такі як підтримка кількох протоколів, хмарні можливості та безсерверна архітектура.

Проблеми: незважаючи на свої переваги, шлюзи API також мають обмеження, такі як обмежена обробка трафіку на межі, нездатність обробляти певні типи трафіку, відсутність видимості внутрішніх комунікацій і проблеми безпеки на межі. рух схід-захід.

Інтеграція шлюзів у складні системи. Приклади: На жаль, у наданих джерелах відсутні конкретні приклади з детальним описом інтеграції шлюзів у складні системи. Однак описані вище загальні принципи та найкращі практики будуть застосовні до широкого діапазону складних системних сценаріїв.

Загалом шлюзи, зокрема шлюзи API в архітектурі мікросервісів, значно спрощують складність керування кількома службами, підвищують безпеку та підвищують ефективність і масштабованість системи. Однак, як і будь-який інструмент, вони мають свої обмеження і повинні бути ретельно інтегровані в загальну архітектуру системи.

2.3.9. Docker

Основи технології докерів і контейнерів. Docker — це платформа контейнеризації, яка дозволяє пакувати програми та їхні залежності в портативні контейнери. Контейнери легкі, ізольовані та можуть стабільно працювати в різних середовищах.

Роль Docker у розробці та розгортанні програм. Docker спрощує розробку та розгортання додатків, забезпечуючи узгоджене середовище від розробки до виробництва. Це спрощує керування залежностями та прискорює процес розгортання.

Робота докера представлена нижче (рис. 2.17):

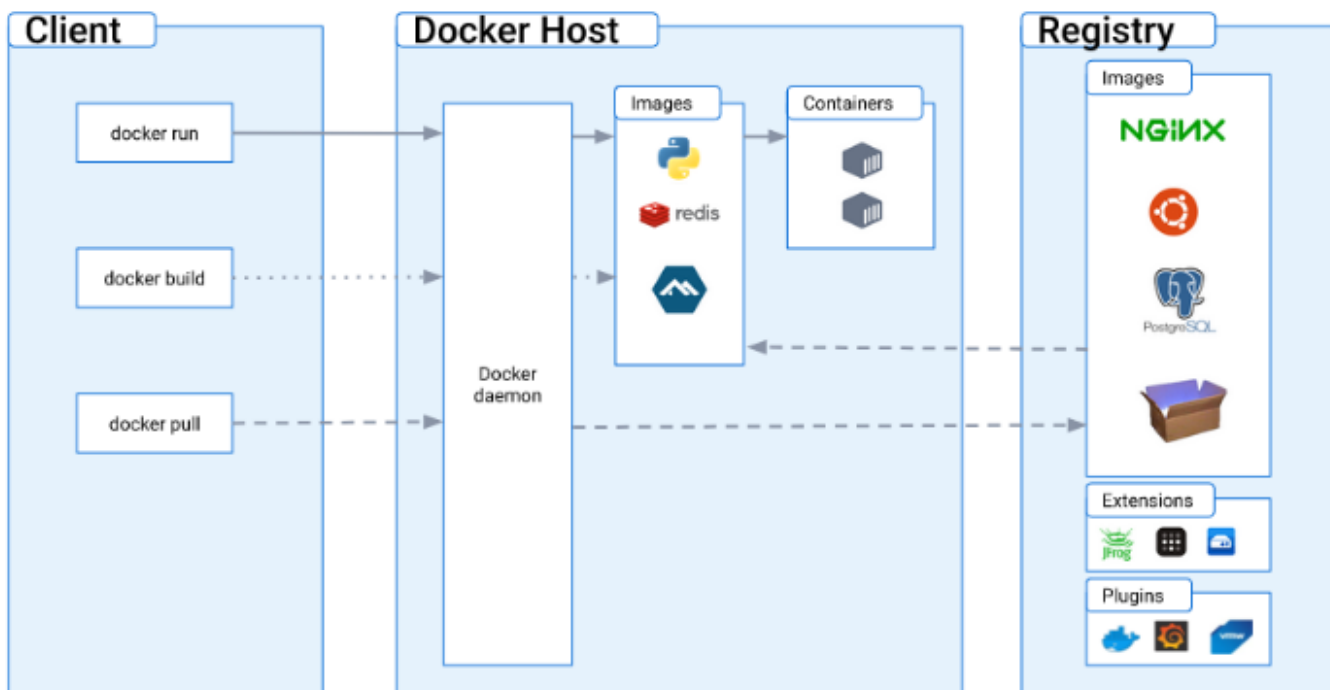


Рис. 2.17. Робота докеру

Оркестрація контейнера з Kubernetes. Kubernetes — це популярна платформа оркестровки контейнерів, яка керує розгортанням, масштабуванням і керуванням контейнерами Docker у хмарному середовищі.

Docker для розробки, тестування та виробництва. Docker є цінним у розробці та тестуванні через копіювання робочих середовищ. Це забезпечує сумісність і масштабованість під час переходу від розробки до виробництва [20,21].

Тематичні дослідження, що демонструють реалізації Docker. Реальні приклади демонструють ефективність Docker у різних галузях, оптимізуючи продуктивність і масштабованість, одночасно оптимізуючи керування програмами.

РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ

3.1. Розробка frontend системи

Будемо використовувати IntelliJ IDEA для розробки фронтенду, підключивши до неї плагін Vue.js (рис. 3.1).

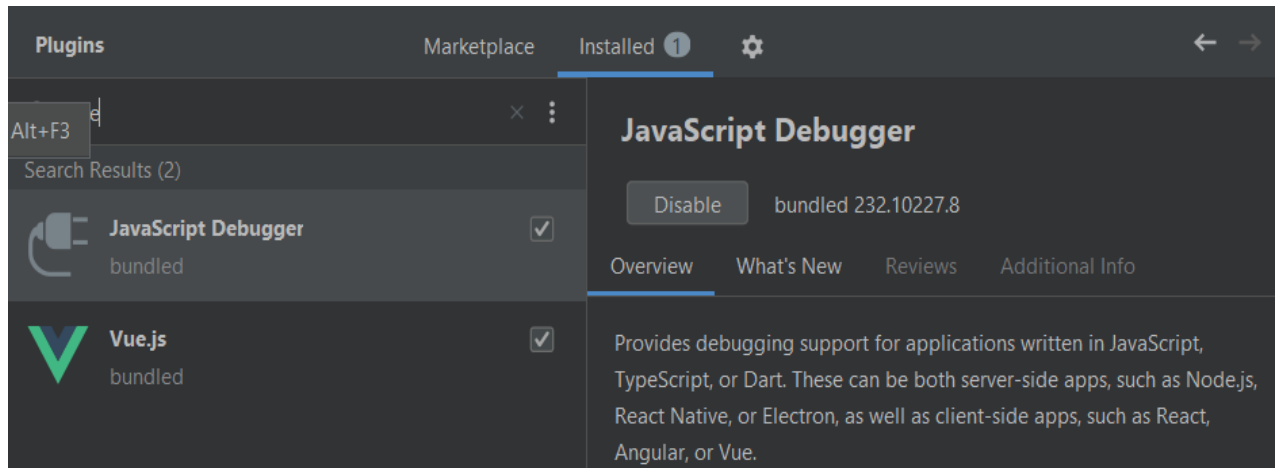


Рис. 3.1. Додавання Vue.js плагіну до IDEA

Під час розробки будемо часто оновлювати такі файли як webpack.config.js та package.json.

Розробимо структури проекту (рис. 3.2).

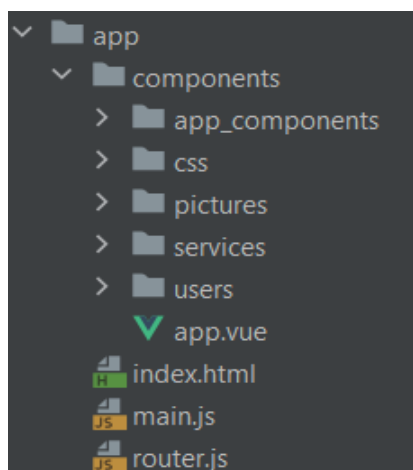


Рис. 3.2. Структура frontend системи

Кафедра КІТ				НАУ 23 03 04 000 ПЗ			
	<i>ПІБ</i>			РОЗДІЛ 3. РОЗРОБКА ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Розроб.</i>	Гречка К.С.					66	24
<i>Керівник</i>	Толстікова О.В.				ТП-215М - 122		
<i>Н.Контр.</i>	Толстікова О.В.						

Пройдемося по кожному компоненту додатку і розглянемо їх відображення на користувацькому інтерфейсі.

Застосунок має для користувачів 5 ролі:

- Admin
- Customer
- Supplier
- Storekeeper
- Anonymous

Роль користувача anonymous дається усім користувачам, які ще не здійснили вхід до застосунку, їх можливості обмежені. Для отримання певних повноважень потрібно пройти авторизацію на сайті. Після отримання певних прав верхня панель має змінити свій вигляд в залежності від ролі. Різні панелі в залежності від ролі користувача (рис. 3.3)

В залежності від ролі користувача головне меню може відрізнятися один від одного, наприклад різниця між Customer та Anonymous.



Рис. 3.3. Користувач anonymous та customer

Завдяки цим функціям користувач отримує свої права, а також може залишатися на сайті певний час. Адже кожні 5 хв його access_token змінюється на новий і він може залишатися на сайті.

```
computed: {  
  loader() {  
    let role = authService.getRole();  
    return () => import(`./users/${role}.vue`);  
  },  
},  
  
mounted() {  
  authService.refreshToken()  
    .then(() => this.component = () => this.loader());  
}
```

Рис. 3.4. Функції отримання ролі та оновлення токена

Сайт складається з 5 основних компонентів:

- Home
- Cabinet
- Cart
- Shop
- Warehouse

Home. Доступний усім користувачам. Це головна сторінка, яку бачить користувач при вході на сайт (рис. 3.5).

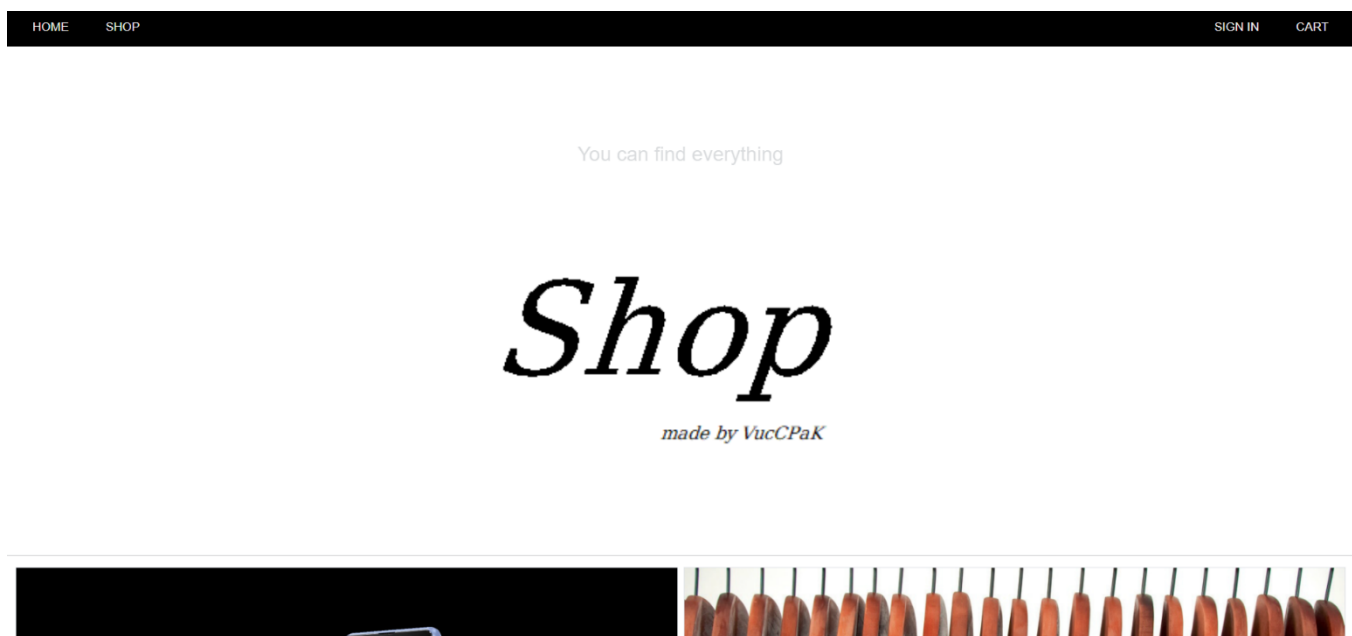


Рис. 3.5. Вигляд головної сторінки

Також можна переглянути категорії, які перекинуть вас на певний розділ магазину, де ви зможете передивитися певний товар (рис. 3.6).

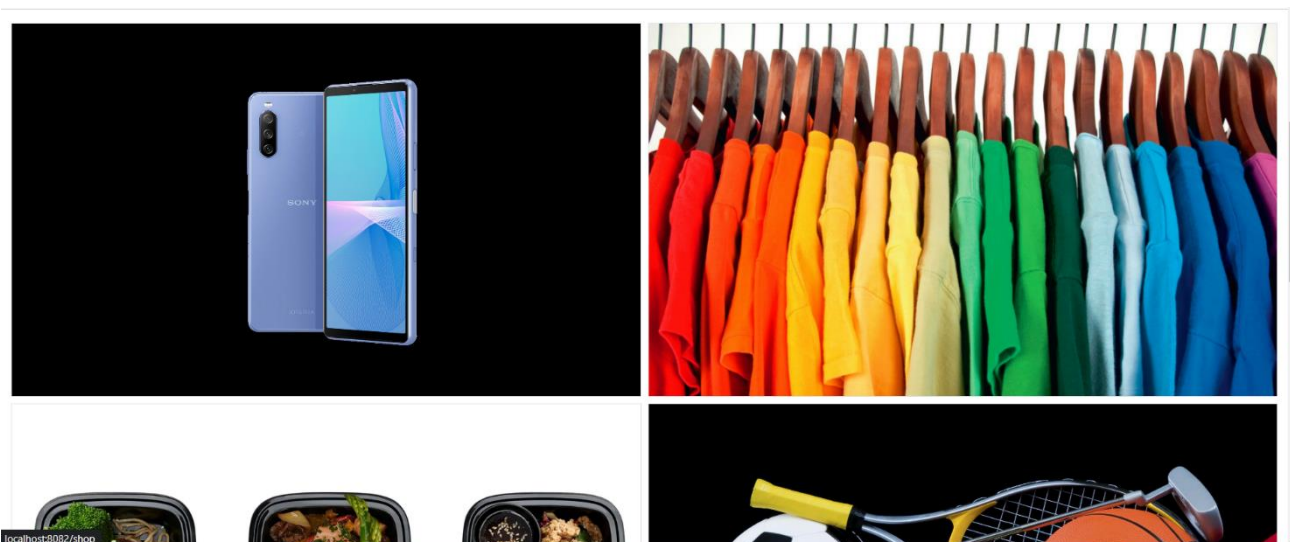


Рис. 3.6. Деякі категорії товарів на головній сторінці

Shop. Магазин де знаходяться товари (рис. 3.7).

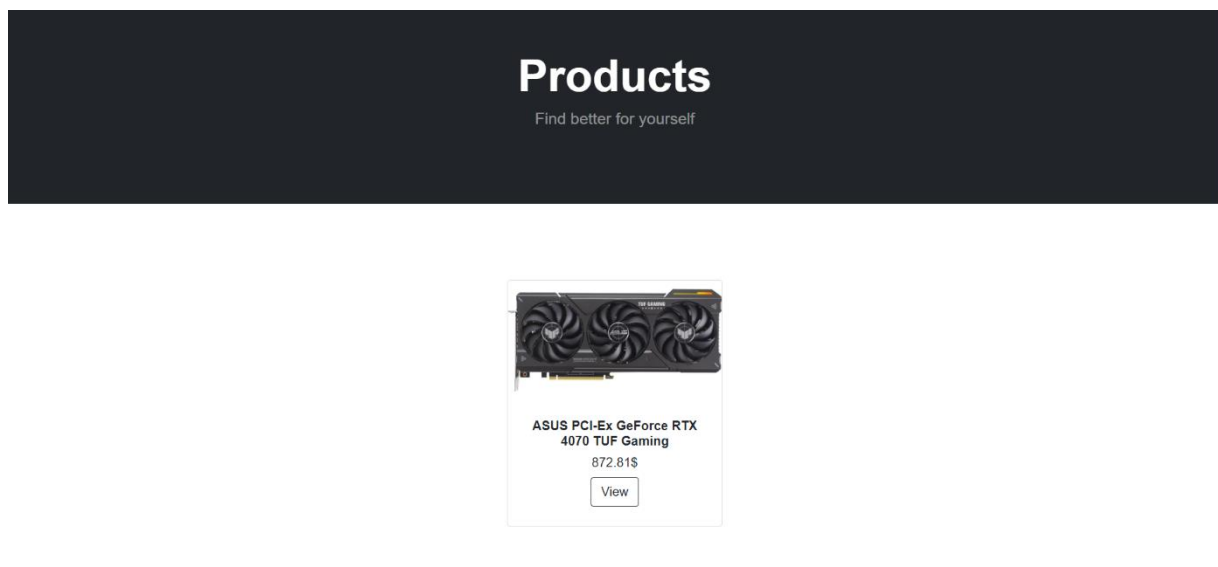


Рис. 3.7. Вигляд магазину товарів

Якщо ви знаходитесь у ролі supplier'a, ваш вигляд сторінки змінюється (рис. 3.8), так як ви входите у групу постачальників, ви можете додати новий товар.

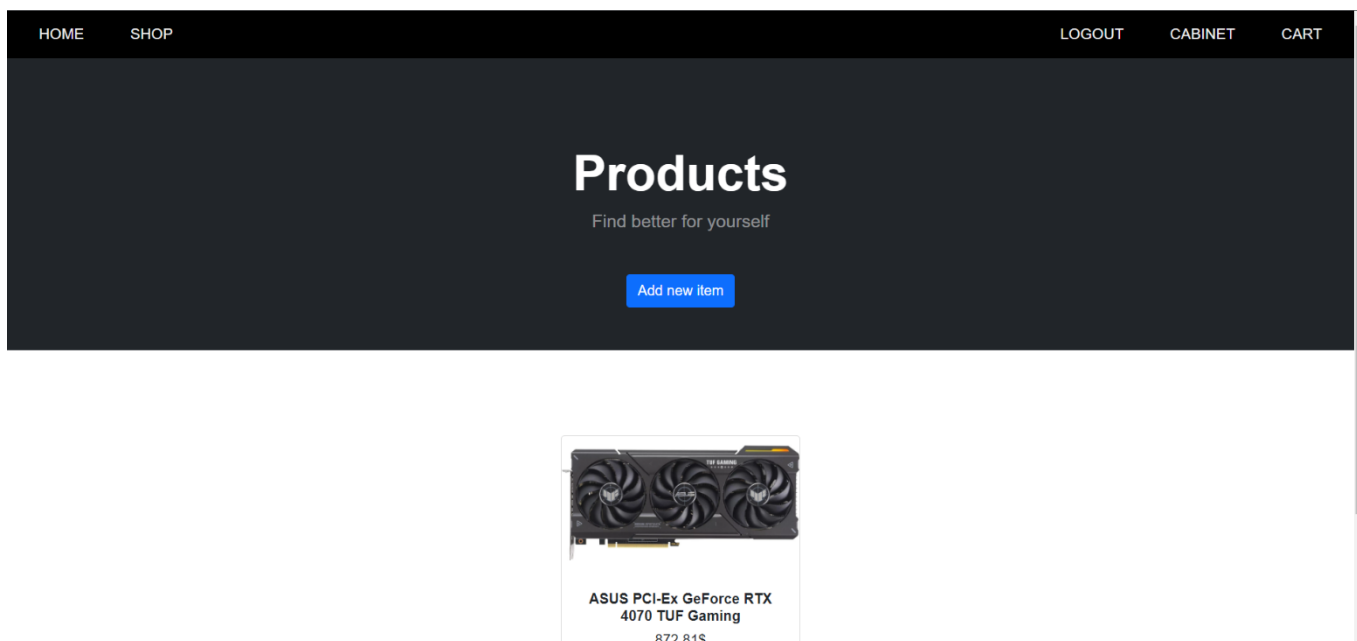
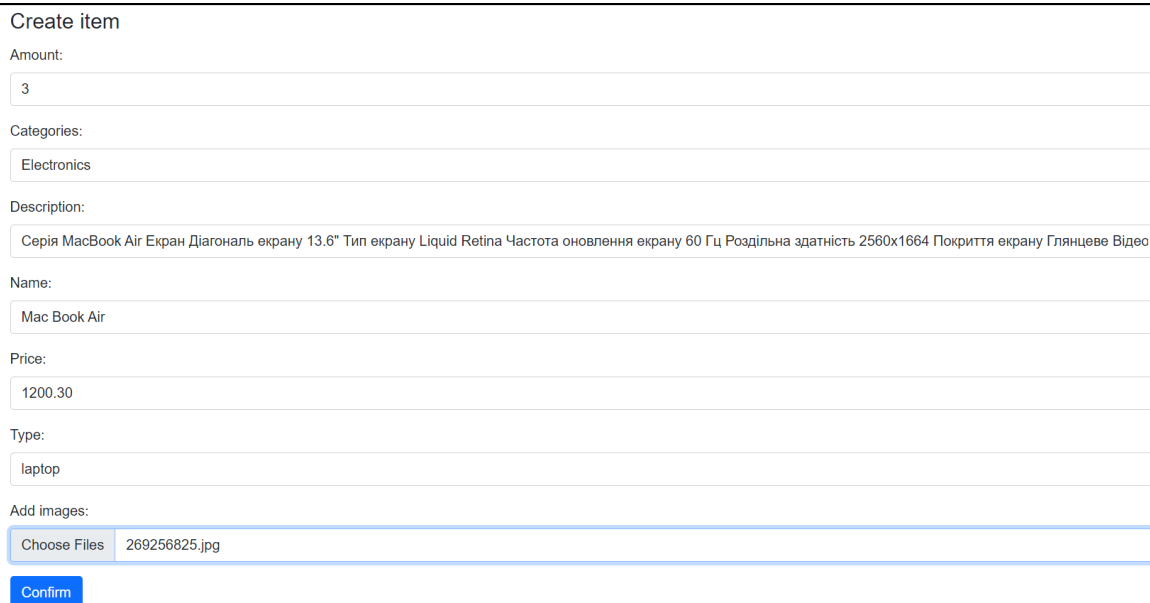


Рис. 3.8. Вигляд магазину товарів для постачальника

При натисканні на кнопку `add new item`, відкриється форма `create_item.vue` (окремий компонент), за допомогою якої можна внести певний товар. Заповнена форма (рис. 3.9).



Create item

Amount:
3

Categories:
Electronics

Description:
Серія MacBook Air Екран Діагональ екрану 13.6" Тип екрану Liquid Retina Частота оновлення екрану 60 Гц Роздільна здатність 2560x1664 Покриття екрану Глянцеве Відео

Name:
Mac Book Air

Price:
1200.30

Type:
laptop

Add images:
Choose Files 269256825.jpg

Confirm

Рис. 3.9. Форма заповнення нового товару

При натисканні на кнопку `confirm`, можна побачити новий товар на екрані магазину (рис. 3.10).

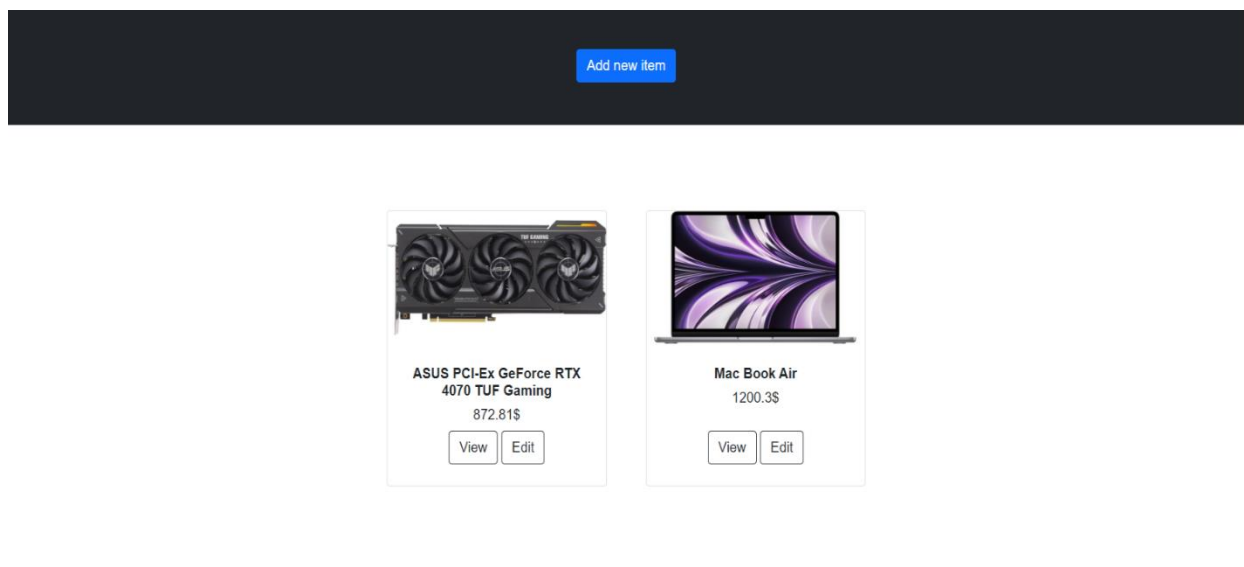


Рис. 3.10. Результат додавання товару до магазину

Якщо цей товар належить до постачальника, який його надає то йому доступна функція оновлення товару (Edit). При натисканні на неї відкриється вікно з поточною інформацією про товар, який можна оновити (рис. 3.11).

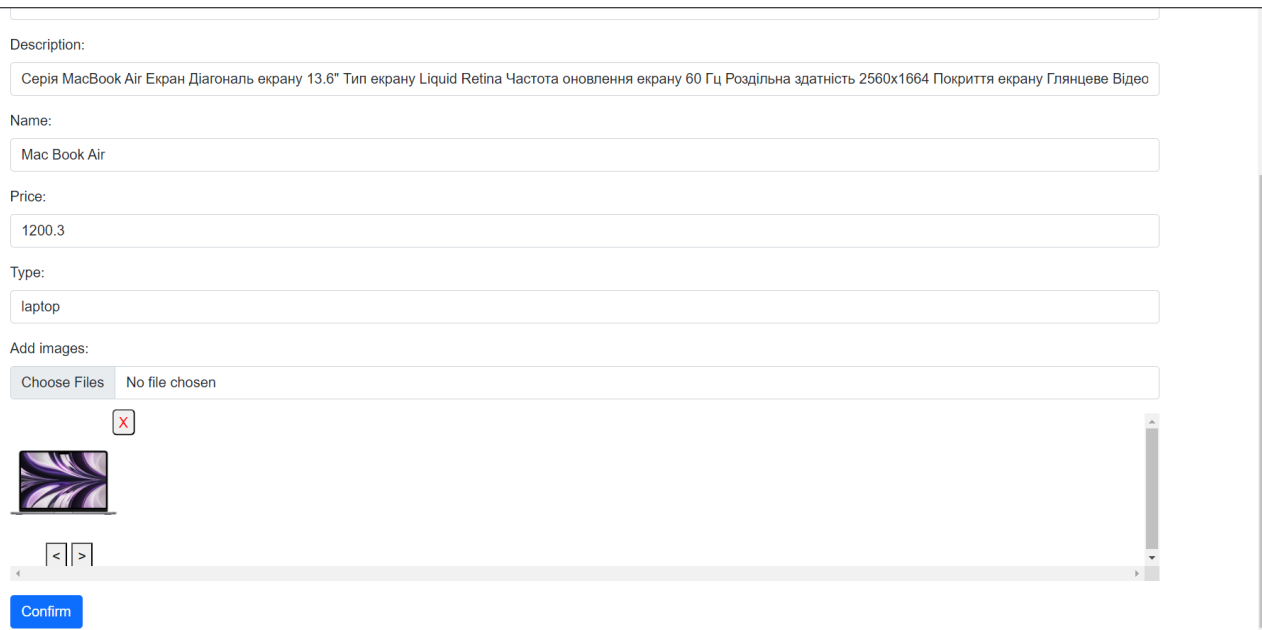


Рис. 3.11. Форма для оновлення товару

Також для всіх користувачів доступна опція перегляду товару, в ній користувач може ознайомитись з інформацією про товар (його опис, які його кількість в наявності, а також зображення товару) (рис. 3.12).

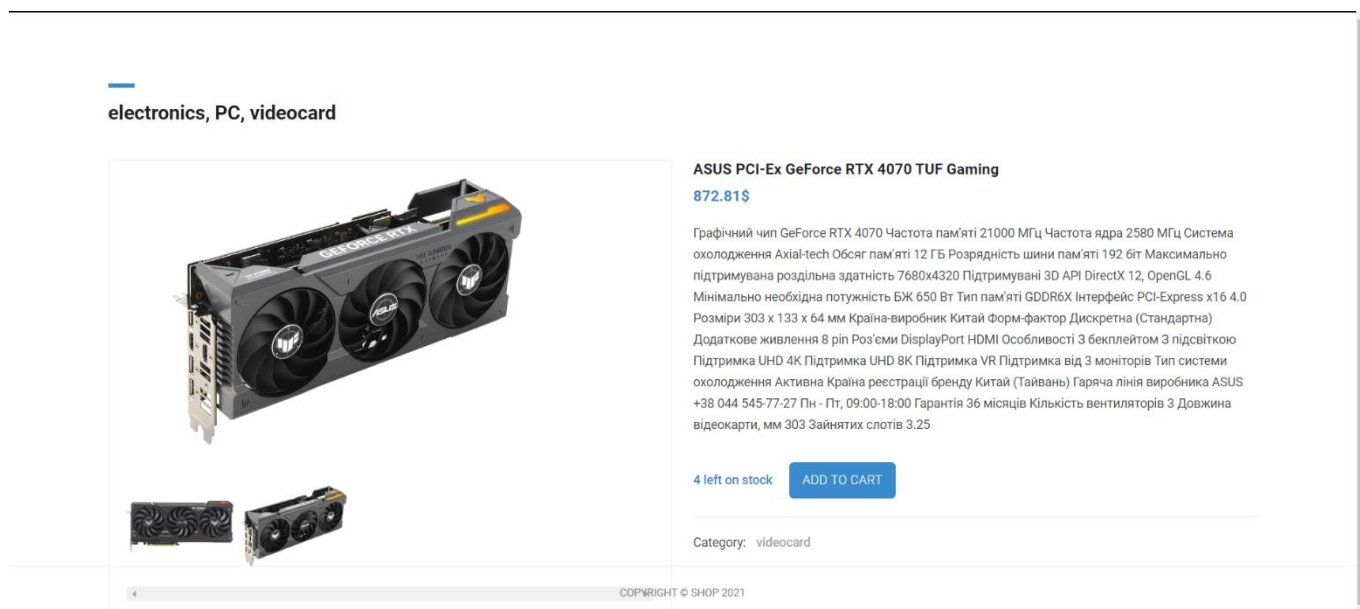


Рис. 3.12. Перегляд товару

Якщо кількість товару зменшиться до нуля або товар буде доданий до корзини, кнопка змінить вигляд в залежності від ситуації (рис. 3.13-3.15).

```
<nonactive_add_button v-if="checkAmount()"/>
<active_add_button v-else-if="!presentInCart"/>
<item-in-the-cart v-else/>
```

Рис. 3.13. Код для зміни кнопки

В залежності від компоненту підгружається певний компонент, який відображає відповідну кнопку користувачу.

підсвіткою клавіатури Порт Thunderbolt Рік випуску 2022 Вид Ноутбуки Додаткові можливості
HD-камера FaceTime HD 1080p Технологія True Tone Touch ID Система з трьох спрямованих
мікрофонів Клавіатура Magic Keyboard з підсвічуванням Датчик зовнішньої освітленості
Трекпад Force Touch Комплект постачання MacBook Air Адаптер живлення USB-C потужністю
30 Вт Кабель USB-C - MagSafe 3 Країна реєстрації бренду США Гарантія 12 місяців офіційної
гарантії від виробника



Рис. 3.14. Товар знаходиться у кошику

мікрофонів Клавіатура Magic Keyboard з підсвічуванням Датчик зовнішньої освітленості
Трекпад Force Touch Комплект постачання MacBook Air Адаптер живлення USB-C потужністю
30 Вт Кабель USB-C - MagSafe 3 Країна реєстрації бренду США Гарантія 12 місяців офіційної
гарантії від виробника



Рис. 3.15. Кількість товару рівна нулю

Cabinet. Показує користувачу його покупки (рис. 3.16). Більш детальну інформацію про покупки можна отримати натиснувши на View items (рис. 3.17).

ID	Date	Total price	Items
6a9cb14f-4b04-442d-9e94-a66ce539520d	2023.12.03 at 18:36:58	3600.9\$	View Items

Get statistical data by products that you supply

Рис. 3.16. Вигляд кабінету користувача з покупками


HOME	SHOP	LOGOUT	WAREHOUSE	CABINET	CART
		<p>Type: Motherboard Name: Материнська плата MSI Z590 Plus Price: 165.7 Quantity: 2</p>			

Рис. 3.17. Перегляд товару, який був замовлений

Кнопка `get statistics data by products you supply` доступна тільки постачальникам, яка надає excel файл, що має контент по статистиці продажу за певний період часу (рис. 3.18).

ID	Date	Total price	Items
6a9cb14f-4b04-442d-9e94-a66ce539520d	2023.12.03 at 18:36:58	3600.9\$	View Items

Get statistical data by products that you supply

Start Date:

End Date:

[Submit](#)

Рис. 3.18. Форма для надання статистики по періоду

Статистика продажу товарів за певним постачальником (рис. 3.19).

	A	B	C	D	E
1	Name	amountStored	quantity	proceeds	profit
2	ASUS PCI-Ex GeForce RTX 4070 TUF Gaming	3	2	1745.62	1 431,41

Рис. 3.19. Статистика продажу товарів

Cart. Кошик, який відображає користувачу його варіанти покупок. Можна змінювати кількість товару в замовленні (рис. 3.20).

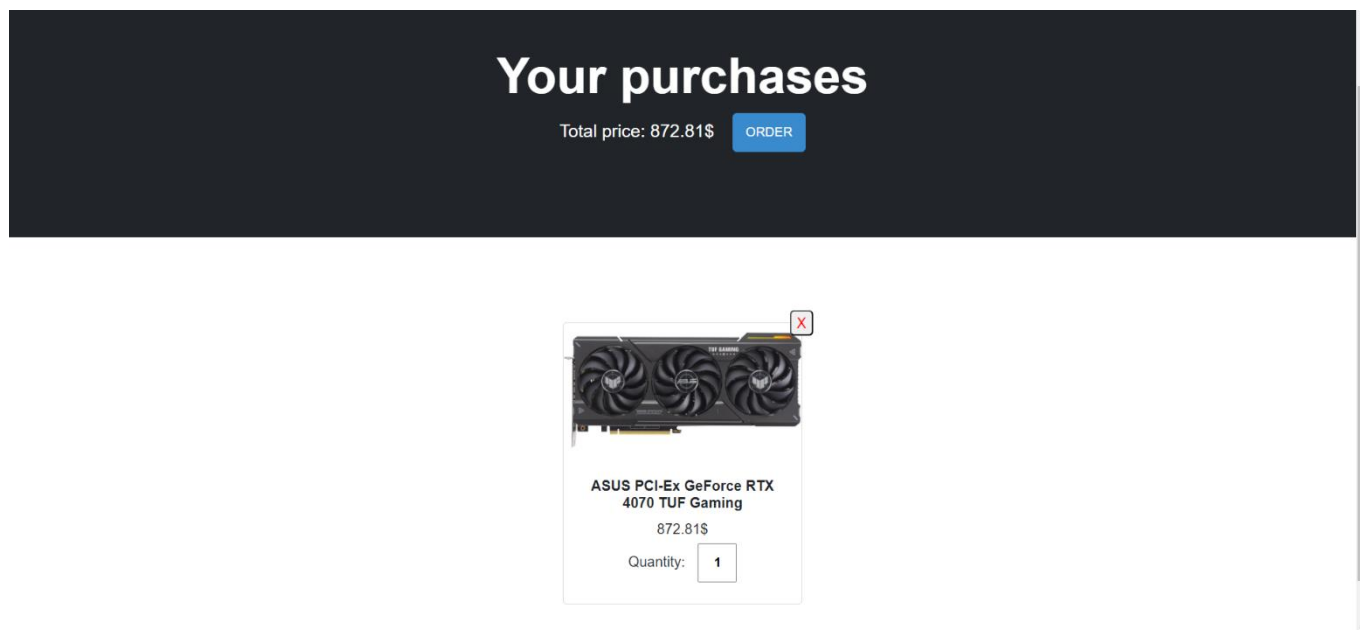


Рис. 3.20. Вигляд кошику

При натисканні на кнопку order, замовлення буде прийняте. Надалі потрібно, щоб storekeeper підтвердив вам відправлення.

Warehouse. Доступний для storekeeper'ів, вони можуть переглянути інформацію по замовленню і підтвердити відправлення даного товару до користувача (рис. 3.21).

HOME	SHOP	LOGOUT	WAREHOUSE	CABINET	CART
ID	Date	Total price	Items	Send	
6a9cb14f-4b04-442d-9e94-a66ce539520d	2023.12.03 at 18:36:58	3600.9\$	View Items	Confirm	

Рис. 3.21. Вигляд сторінки warehouse

Після натискання на кнопку view items, комірник може переглянути інформацію по придбаному товару (рис. 3.22), а кнопкою confirm підтвердити відправку товару до користувача.


HOME	SHOP	LOGOUT	WAREHOUSE	CABINET	CART
		Type: laptop Name: Mac Book Air Price: 1200.3 Quantity: 3			

Рис. 3.22. Перегляд придбаного товару

Роутер Vue.js в даному доадутку.

Vue Router — це офіційний маршрутизатор для Vue.js, розроблений для бездоганної роботи з інфраструктурою Vue.js. Він використовується для створення односторінкових програм (SPA), де залежно від URL-адреси відображаються різні види чи компоненти, але без необхідності повного перезавантаження сторінки. Ось ключові функції та поняття, пов'язані з Vue Router:

Кожний компонент підлягає певному посилання (рис. 3.23).

```
routes: [  
  {  
    path: '/',  
    component: home  
  },  
  {  
    path: '/shop',  
    component: shop,  
  },  
  {  
    path: '/cart',  
    component: cart  
  },  
  {  
    path: '/shop/:id',  
    component: commodity  
  },  
  {  
    path: '/cabinet',  
    component: cabinet  
  },  
  {  
    path: '/warehouse',  
    component: warehouse  
  },  
  {  
    path: '/warehouse/:id',  
    component: itemsInTheOrder  
  },  
  {  
    path: '/shop/supply/:id',  
    component: supply  
  },  
  {  
    path: '/shop/supply/item/new',  
    component: create_item  
  },  
  {  
    path: '/cabinet/order/:id',  
    component: itemsInTheOrder  
  }  
]
```

Рис. 3.23. Маршрутизація у фронтенд системі

Дизайн і API Vue Router призначені для забезпечення плавної розробки під час створення SPA за допомогою Vue.js. Його тісна інтеграція з Vue.js означає, що він успадковує реактивність Vue і компонентну архітектуру, що робить його природним вибором для розробників Vue.

Сервіси, які використовує додаток:

- authService
- cabinetService
- cartService
- shopService
- supplyService
- warehouseService

Авторизаційний сервіс – це система, яка використовується для перевірки ідентичності користувачів та надання їм доступу до певних ресурсів чи послуг. Ось основні аспекти авторизаційних сервісів:

Аутентифікація

Це процес перевірки, що користувач є тим, за кого себе видає. Це може включати введення імені користувача та пароля, використання біометричних даних, двофакторної аутентифікації тощо.

Авторизація

Після аутентифікації система визначає, які ресурси або функції доступні для користувача. Це може включати обмеження доступу до певних даних або функцій залежно від ролі користувача.

Управління обліковими записами

Це включає створення, управління та видалення облікових записів користувачів. Авторизаційні сервіси часто включають інструменти для адміністраторів для управління користувацькими обліковими записами.

Функції авторизаційного сервісу:

`handleCallback`. Ця функція обробляє URL-колбек від сервера авторизації. Вона отримує код авторизації з URL, обмінює його на набір токенів і зберігає отримані токени в сесії та локальному сховищі.

`getConfig`. Завантажує конфігурацію сервера авторизації (наприклад, точки кінця токенів), використовуючи HTTP-запит.

`getRole`. Отримує роль користувача з збереженого у сесії токена доступу. Якщо токен недоступний, повертає 'anonymous'.

`getUserId`. Отримує ідентифікатор користувача з токена доступу.

`parseJwt`. Розбирає JWT (JSON Web Token), декодує його з base64 і перетворює в JSON-об'єкт.

`updateAccessToken`. Оновлює токен доступу, використовуючи наявний токен оновлення. Зберігає нові токени в локальному сховищі та сесії.

`isRefreshTokenPresent`. Перевіряє, чи є токен оновлення у локальному сховищі.

`removeAccessAndRefreshToken`. Видаляє токени доступу та оновлення з локального сховища та сесії.

`refreshToken`. Оновлює токен доступу кожні 3 хвилини (з урахуванням 5 секунд до закінчення строку дії) та періодично оновлює його, якщо токен оновлення присутній.

Кожна з цих функцій відіграє важливу роль у забезпеченні безпеки і управлінні доступом користувачів до веб-додатків або послуг.

Функції cabinet сервісу:

Цей JavaScript код визначає клас `WarehouseService` з трьома статичними функціями для взаємодії з сервером через HTTP-запити:

`getList(id)`: Здійснює GET-запит до сервера за адресою, що формується із змінної `VALUES.GATEWAY` і динамічного сегмента URL, який залежить від переданого `id`. Ця функція призначена для отримання списку елементів (наприклад, товарів або документів) із заданого кабінету або розділу.

`getSupplierStatistic (id, period)`: Виконує GET-запит для отримання статистики постачальника. Вона приймає `id` постачальника та об'єкт `period`, який містить дати початку і кінця періоду. Запит повертає дані у форматі `Blob`, які, ймовірно, представляють собою файл електронної таблиці.

`getAdminStatistic(period)`: Аналогічно здійснює GET-запит для отримання адміністративної статистики за заданий період. Параметри `startDate` та `endDate` вказують на період, за який потрібно згенерувати статистику. Результат також повертається у форматі `Blob`, ймовірно як файл електронної таблиці.

Функції cart сервісу:

Цей фрагмент JavaScript коду визначає клас `CartService`, який керує кошиком покупок на веб-сайті:

`getList()`: Повертає список товарів у кошику.

`remove(id)`: Видаляє товар з кошика за його ідентифікатором `id`.

`add(item)`: Додає новий товар у кошик. Якщо товар уже присутній, нічого не робить.

`isPresentInCart(id)`: Перевіряє, чи присутній товар у кошику за його ідентифікатором `id`.

`updateAmount(id, quantity)`: Оновлює кількість конкретного товару в кошику.

`getTotalPrice()`: Обчислює загальну ціну товарів у кошику, відправляючи запит на сервер.

`toOrder(userId)`: Створює замовлення для користувача на основі товарів у кошику, також відправляючи запит на сервер.

Клас `DCartItem` використовується для створення об'єктів з ідентифікатором товару та його кількістю, які потім використовуються у функціях `getTotalPrice` та `toOrder`.

Функції shop сервісу:

Цей фрагмент JavaScript коду описує клас `ShopService`, який містить три методи для взаємодії з сервером:

`getListOfItems()`: Виконує GET-запит до сервера для отримання списку товарів. Повертає дані, отримані у відповіді від сервера.

`getItemById(id)`: Виконує GET-запит до сервера, щоб отримати деталі конкретного товару за його ідентифікатором `id`. Повертає дані, отримані від сервера.

`getPictureUrl(epicture)`: Формує URL для отримання зображення товару. Це не HTTP-запит, а лише утворення URL на основі ідентифікатора зображення `epicture`.

Функції supply сервісу:

Цей JavaScript код описує анонімний клас з методами для управління товарами у веб-додатку:

`updateItem (id, item)`: Оновлює інформацію про товар, використовуючи POST-запит з оновленими даними товару (`DItem`).

`addImages (id, formData)`: Додає зображення для товару, спершу відправляючи зображення на сервер, а потім асоціюючи отримані ідентифікатори зображень з конкретним товаром.

`changePriorities (id, array)`: Змінює порядок зображень товару, відправляючи оновлену послідовність через POST-запит.

`removeImages (pictures)`: Видаляє зображення товару, використовуючи POST-запит для передачі ідентифікаторів зображень, які потрібно видалити.

`createItem (itemProperties)`: Створює новий товар, відправляючи його характеристики через POST-запит.

Клас `DItem` використовується для створення об'єкта товару з набором характеристик, таких як назва, ціна, кількість, опис, зображення, категорії, тип та ідентифікатор постачальника.

Функції warehouse сервісу:

У цьому коді описано клас `WarehouseService` з трьома методами для взаємодії з сервером:

`getList ()`: Виконує GET-запит до сервера для отримання списку об'єктів або даних зі складу. Повертає відповідь у форматі даних, отриманих від сервера.

`submit(id)`: Відправляє POST-запит на сервер для підтвердження або обробки елемента за його ідентифікатором `id`, вказуючи в URL параметр `sent=true`.

`getItemsInOrder(id)`: Виконує GET-запит до сервера для отримання деталей про конкретний елемент або замовлення за його ідентифікатором `id`. Повертає детальну інформацію про цей елемент або замовлення.

3.2. Розробка `pictures` системи

Структура `pictures` системи (рис. 3.24).

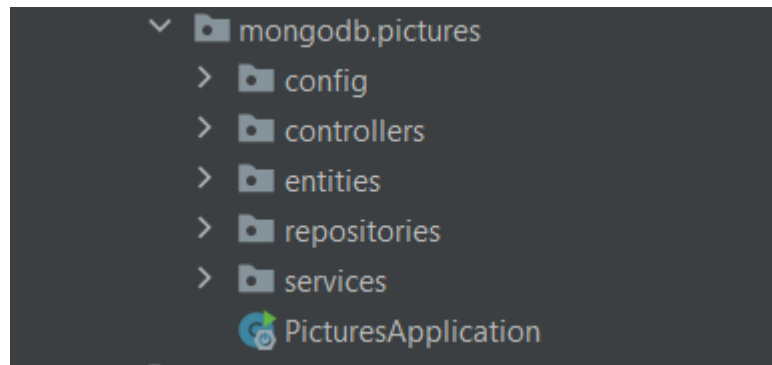


Рис. 3.24. Структура

config: Містить `SecurityConfig`, клас конфігурації, що використовує Spring Security для налаштування безпеки API, включаючи авторизацію на основі JWT та контроль доступу.

controllers: Включає `PictureController`, який управляє HTTP запитамися для зберігання, видалення та отримання зображень. Використовує методи `saveImage`, `removeImage` і `getPictureById` з огляду на ролі користувачів.

entities: Визначає клас `Picture`, що представляє сутність зображення в базі даних, з полями для ідентифікатора та бінарних даних зображення.

repositories: Містить `PictureRepository`, інтерфейс для взаємодії з MongoDB, використовуючи Spring Data MongoDB.

services: Включає `PictureService`, сервіс для обробки бізнес-логіки, пов'язаної зі зображеннями, включаючи зберігання, видалення зображень та отримання зображень за ідентифікатором.

3.3. Розробка gateway системи

Цей проект Spring Boot Gateway служить як центральний точка доступу (gateway) для маршрутизації запитів до різних мікросервісів в системі. Ось його ключові конфігурації:

Сервер: Запускається на порту 8086 за адресою localhost.

Spring Cloud Gateway:

Global CORS Configuration: Встановлює CORS (Cross-Origin Resource Sharing) правила для всіх маршрутів ([/**]), дозволяючи запити від http://localhost:8082 з будь-якими методами та заголовками, а також підтримує передачу облікових даних.

Маршрути:

Backend: Маршрутизує запити з /api/backend/** до сервісу, що працює на http://localhost:8081/.

Pictures: Маршрутизує запити з /api/mongo/** до сервісу, що працює на http://localhost:8087/.

Keucloak: Маршрутизує запити з /auth/** до Keucloak, що працює на http://localhost:8080/.

3.4. Розробка embedded-ldap системи

Цей проект "Embedded-LDAP Spring" використовує LDAP (Lightweight Directory Access Protocol) для створення вбудованої системи керування користувачами та групами. Проект організовано наступним чином:

Основний Домен: Визначає кореневий вузол домену LDAP (dc=embedded, dc=ldap).

Організаційні Одиниці: Структуровано на дві основні одиниці: ou=groups для груп та ou=users для користувачів.

Записи Користувачів і Груп: Визначає різні типи користувачів (наприклад, замовників, постачальників, комірників, адміністраторів) із їхніми відповідними атрибутами (ім'я, прізвище, UID, логін, пароль тощо).

Групи: Створює специфічні групи (наприклад, customers, suppliers, storekeepers, admins) та визначає унікальних членів цих груп.

3.5. Розробка backend системи

Структура backend системи:

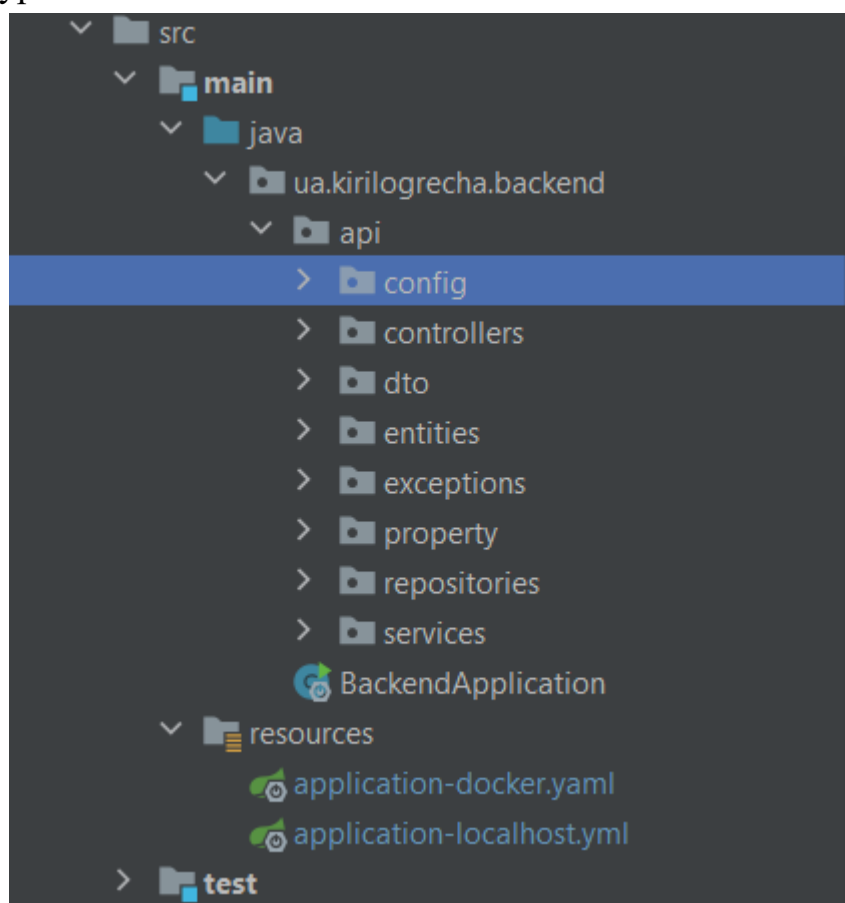


Рис. 3.25. Структура

SecurityConfig: Налаштовує безпеку API, включаючи JWT аутентифікацію та конфігурацію CORS.

CabinetController: Обробляє запити для отримання інформації про замовлення користувачів.

CartController: Управління кошиком покупців, включаючи розрахунок загальної ціни та оформлення замовлень.

ItemController: Керування товарами, включаючи отримання списку товарів, деталей по конкретним товарам та оновлення інформації про товари.

PictureController: Управління зображеннями товарів, включаючи збереження, видалення та зміну пріоритетів зображень.

StatisticController: Генерує статистику для різних ролей (наприклад, постачальників та адміністраторів) у форматі електронних таблиць.

WarehouseController: Управління складом, включаючи перегляд списку замовлень та управління станом замовлень.

Ці контролери взаємодіють з сервісним шаром для виконання бізнес-логіки, використовуючи PostgreSQL для зберігання даних про товари, MongoDB для зберігання зображень і Keycloak для аутентифікації та авторизації.

Сутності (Entities) в проекті Spring Boot відіграють ключову роль у визначенні структури бази даних і відображенні її на об'єкти в Java. Ось деталі кожної сутності:

EItem: Представляє товари, зберігає інформацію про назву, ціну, кількість, опис, категорії, тип та ідентифікатор постачальника. Містить зв'язок зі зображеннями та елементами замовлень.

EOrder: Відображає замовлення, включаючи дату, загальну ціну, статус відправлення та зв'язок з елементами замовлення.

EOrderItem: Представляє окремі елементи в замовленні, зв'язані з конкретними товарами та замовленнями.

EPicture: Зберігає інформацію про зображення товарів, включаючи назву, пріоритет та зв'язок з товаром.

Репозиторії в проекті Spring Boot використовуються для взаємодії з базою даних через JPA (Java Persistence API). Ось опис кожного репозиторію:

ItemRepository: Надає функціональність для роботи з ентиті EItem, включаючи зміну кількості товару за допомогою кастомного запиту.

OrderItemRepository: Призначений для роботи з ентиті EOrderItem, дозволяє виконувати базові CRUD-операції.

OrderRepository: Використовується для управління замовленнями (EOrder), містить кастомні запити для оновлення статусу замовлення, отримання списку замовлень та фільтрації замовлень за періодом часу та ідентифікатором користувача.

PictureRepository: Дозволяє виконувати операції з ентиті EPicture, включаючи зберігання та отримання зображень товарів.

Сервіси у Spring Boot проекті виконують різноманітні бізнес-функції:

CabinetService: Управляє операціями, пов'язаними із замовленнями користувачів, використовуючи OrderRepository.

CartService: Відповідає за роботу з кошиком покупців, зокрема розрахунок загальної ціни кошика та обробку замовлень.

ItemService: Керує товарами, включаючи отримання інформації про товари, їх оновлення та створення нових записів товарів.

PictureService: Управління зображеннями товарів, включаючи зміну пріоритетів зображень та їх видалення.

StatisticService: Генерація статистики за певний період, використовуючи дані із замовлень і товарів.

WarehouseService: Управління складськими операціями, включаючи отримання списку замовлень та оновлення статусу замовлень.

3.6. Розгортання Keycloak та баз даних на Docker

Для більш простого розгортання баз даних та Keycloak у докері ми використаємо такий інструмент як `docker-compose`. За допомогою `yaml` коду ми пропишемо логіку при розгортанні наших `images` та їх `volumes`.

Вигляд `yaml` коду для розгортання (рис. 3.26):

```
version: '3'

services:
  keycloak:
    image: bitnami/keycloak:16.1.1
    environment:
      - KEYCLOAK_DATABASE_HOST=keycloak-postgres
      - KEYCLOAK_DATABASE_USER=postgres
      - KEYCLOAK_DATABASE_PASSWORD=postgres
      - KEYCLOAK_DATABASE_NAME=keycloak
      - KEYCLOAK_ADMIN_USER=admin
      - KEYCLOAK_ADMIN_PASSWORD=admin
      - KEYCLOAK_IMPORT=/tmp/my-realm.json
    volumes:
      - C:\Users\Kyrylo\Desktop\graduation\realm-export.json:/tmp/my-realm.json
    ports:
      - '8080:8080'
    depends_on:
      - keycloak-postgres

  keycloak-postgres:
    image: bitnami/postgresql:latest
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres
      - POSTGRES_DB=keycloak
    volumes:
      - keycloak_data:/var/lib/postgresql/data
    ports:
      - "5433:5432"

  goods-postgres:
    image: bitnami/postgresql:latest
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres
      - POSTGRES_DB=project
    volumes:
      - goods_data:/var/lib/postgresql/data
    ports:
      - "5432:5432"

  pictures-mongodb:
    image: bitnami/mongodb:latest
    environment:
      - MONGODB_ROOT_PASSWORD=mongo
      - MONGODB_USERNAME=mongo
      - MONGODB_PASSWORD=mongo
      - MONGODB_DATABASE=pictures
    volumes:
      - pictures_data:/data/db
    ports:
      - "27017:27017"

volumes:
  goods_data:
  keycloak_data:
  pictures_data:
```

Рис. 3.26. Docker-compose

Пройдемося по коду. Як видно вище код розбитий на 4 сервіси, а саме keycloak, keycloak-postgres, goods-postgres, pictures-mongodb. А також на volumes goods_data, keycloak_data та pictures_data, за допомогою яких наші дані не втрачаються після кожного запуску контейнера.

Keycloak сервіс. Надає нам увесь необхідний функціонал keycloak, який можна знайти за посиланням по localhost:8080 (рис. 3.27).

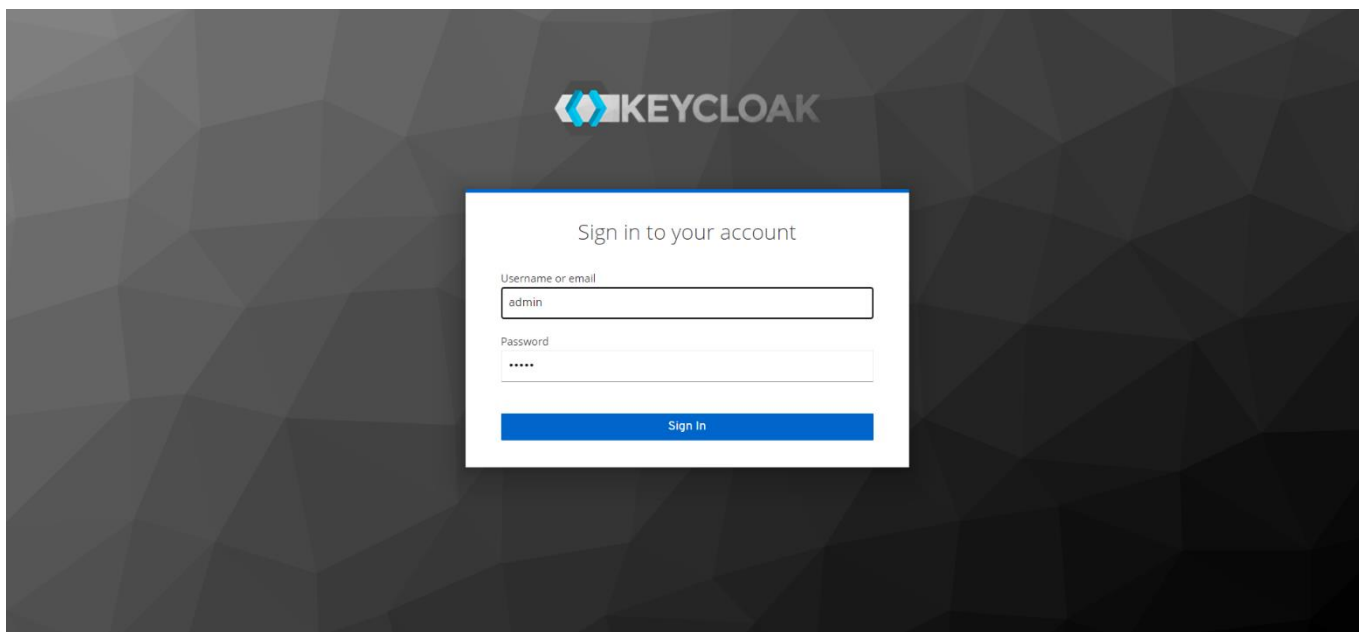


Рис. 3.27. Логін сторінка для доступу в keycloak

Для авторизації наших користувачів у наш додаток нам потрібно налаштувати keycloak на коректну роботу, а саме додати новий realm (рис. 3.28) та client (рис. 3.29). А також підключити користувачів через LDAP (рис. 3.30).

Add realm

Import	<input type="button" value="Select file"/>
Name *	<input type="text"/>
Enabled	<input checked="" type="checkbox"/>
	<input type="button" value="Create"/> <input type="button" value="Cancel"/>

Рис. 3.28. Створення нового realm'у

Always Display in Console ?	<input type="checkbox"/> OFF
Consent Required ?	<input type="checkbox"/> OFF
Login Theme ?	<input type="text" value=""/>
Client Protocol ?	<input type="text" value="openid-connect"/>
Access Type ?	<input type="text" value="public"/>
Standard Flow Enabled ?	<input checked="" type="checkbox"/> ON
Implicit Flow Enabled ?	<input type="checkbox"/> OFF
Direct Access Grants Enabled ?	<input checked="" type="checkbox"/> ON
OAuth 2.0 Device Authorization Grant Enabled ?	<input type="checkbox"/> OFF
Front Channel Logout ?	<input type="checkbox"/> OFF
Root URL ?	<input type="text" value="http://localhost:8082"/>
* Valid Redirect URIs ?	<input type="text" value="http://localhost:8082/*"/> <input type="button" value="-"/> <input type="text" value=""/> <input type="button" value="+"/>
Base URL ?	<input type="text" value=""/>
Admin URL ?	<input type="text" value="https://www.keycloak.org/app/"/>
Logo URL ?	<input type="text" value=""/>
Policy URL ?	<input type="text" value=""/>
Terms of service URL ?	<input type="text" value=""/>
Web Origins ?	<input type="text" value="http://localhost:8082"/> <input type="button" value="-"/> <input type="text" value="https://www.keycloak.org"/> <input type="button" value="-"/> <input type="text" value=""/> <input type="button" value="+"/>

Рис. 3.29. Налаштування client'у

* Edit Mode ?	<input type="text" value="READ_ONLY"/>
Sync Registrations ?	<input type="checkbox"/> OFF
* Vendor ?	<input type="text" value="Active Directory"/>
* Username LDAP attribute ?	<input type="text" value="login"/>
* RDN LDAP attribute ?	<input type="text" value="login"/>
* UUID LDAP attribute ?	<input type="text" value="uid"/>
* User Object Classes ?	<input type="text" value="top, person, organizationalPerson, inetOrgPerson"/>
* Connection URL ?	<input type="text" value="ldap://host.docker.internal:8084"/>
* Users DN ?	<input type="text" value="ou=users,dc=embedded,dc=ldap"/>
Custom User LDAP Filter ?	<input type="text" value="LDAP Filter"/>
Search Scope ?	<input type="text" value="One Level"/>
* Bind Type ?	<input type="text" value="simple"/>
* Bind DN ?	<input type="text" value="uid=admin"/>
* Bind Credential ?	<input type="text" value="....."/> <input type="button" value="eye"/>

Рис. 3.30. Налаштування LDAP доступу користувачів

Keycloak-postgres. Так як keycloak підкапотно використовує базу даних для зберігання своїх налаштувань, то потрібно надати йому доступ до цієї бази. Деякі таблиці keycloak, які він зберігає (рис. 3.31).

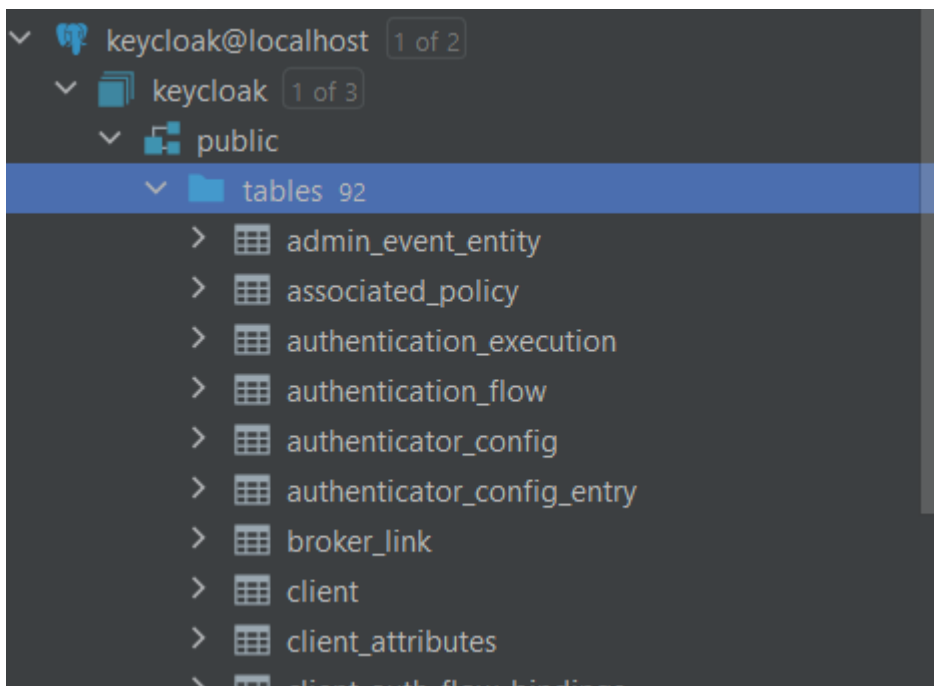


Рис. 3.31. Таблиці Keycloak

Goods-postgres. Використовується для зберігання товарів. Схема бази даних для товарів (рис 3.32).

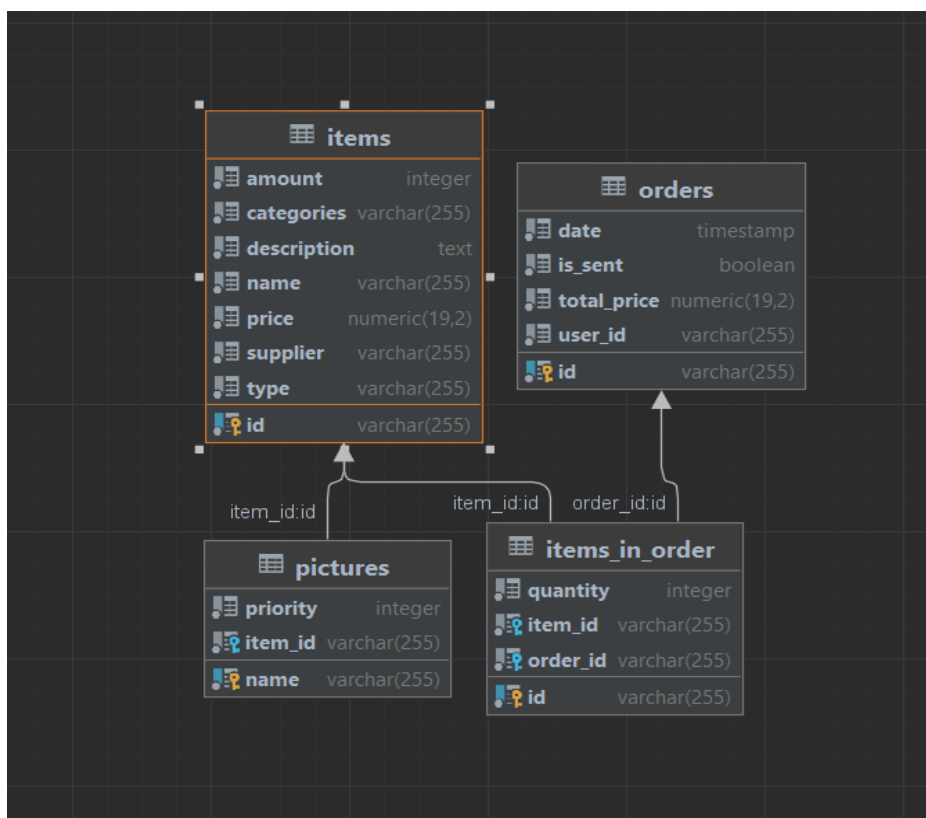


Рис. 3.32. Схема-діаграма бази даних товарів

Pictures-mongodb. Окрема база даних, яка використовується для зберігання зображень товару (рис. 3.33). MongoDB більше підходить для зберігання зображень у двійковому форматі.

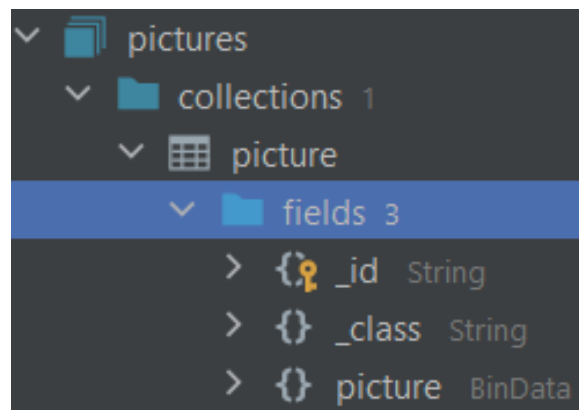


Рис. 3.33. Структура бази даних для зображень

Docker ефективно трансформував процеси розробки, тестування та розгортання програм, надаючи рішення, яке прискорює ці процеси та робить їх більш ефективними і послідовними. Ця технологія контейнеризації здобула широке визнання серед розробників і стала важливою складовою в сфері DevOps, оскільки вона інтегрується з численними інструментами цієї області, підвищуючи ефективність управління програмними проектами та інфраструктурою.

Завдяки Docker, команди розробників можуть легко пакувати додатки разом з усіма їх залежностями в стандартизовані контейнери. Це не тільки зменшує ризики, пов'язані з різницею в середовищах розробки та продукції, але й забезпечує гнучкість у розгортанні на будь-якій платформі. Таким чином, Docker сприяє більшій уніфікації та стандартизації в процесі розробки, що є важливим для сучасних методологій розробки програмного забезпечення.

ВИСНОВКИ

Дана кваліфікаційна робота демонструє розробку високопродуктивного вебзастосунку для обліку товарів і продажів на основі мікросервісної архітектури. Основною метою роботи є розробка незалежних, але взаємопов'язаних мікросервісів, які разом утворюють єдину систему обробки замовлень та обліку продажів.

Використання Spring Boot як основи для побудови окремих служб забезпечує гнучкість у розробці та розгортанні, дозволяючи інтегрувати сучасні підходи до програмування та впроваджувати зміни без поломки системи. PostgreSQL використовується для забезпечення надійного та ефективного зберігання даних, тоді як MongoDB використовується для керування зображеннями продуктів, забезпечуючи оптимальну гнучкість і швидкість обробки. Додаток Keycloak дозволяє реалізувати надійний механізм аутентифікації та авторизації, підвищуючи безпеку системи.

Архітектура мікросервісів сприяє не тільки масштабованості та високій доступності системи, але й забезпечує її стійкість до збоїв, оскільки проблеми в одному сервісі не впливають на роботу інших.

Використання архітектури мікросервісів забезпечує більш високий рівень гнучкості та спрощує впровадження окремих сервісів. Основною концепцією мікросервісів є можливість самостійного розгортання та оновлення окремих компонентів програми, що дозволяє вносити зміни в окремі частини системи, не зачіпаючи решту. Це також полегшує масштабування окремих служб без масштабування всієї системи. Використання REST із підходом без збереження стану забезпечує додаткову безпеку, оскільки кожен запит HTTP обробляється окремо, незалежно від попередніх запитів, і вся необхідна інформація передається в межах поточного запиту.

Загалом результати цієї роботи підкреслюють переваги архітектури мікросервісів у розробці сучасних веб-додатків, особливо у сфері електронної комерції, де важливі гнучкість, ефективність та безпека системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стаття: Md Halimuzzaman, Jaideep Sharma. The evolution of accounting information systems (AIS) and enterprise resource planning (ERP): a review of literature. Bharat College of Arts and Commerce, Badlapur, MMR, India. 2023. Volume 12, Issue 1. С. 170 – 181.
2. Ресурс інтернету: How Innovative Accounting Inventory Methods Can Boost Efficiency. URL: <https://oboloo.com/blog/revolutionizing-your-supply-chain-how-innovative-accounting-inventory-methods-can-boost-efficiency/> (04.10.2023)
3. Ресурс інтернету: The Evolving Role of IoT in Supply Chain and Logistics. URL: <https://biztechmagazine.com/article/2021/05/evolving-role-iot-supply-chain-and-logistics-perfcon> (06.10.2023)
4. Стаття: Stefan Wengler, Gabriele Hildmann, Ulrich Vossebein. Digital transformation in sales as an evolving process. Journal of Business & Industrial Marketing. 2021. С. 599 – 600.
5. Ресурс інтернету. The Role of Technology in Inventory Management. URL: <https://www.lengow.com/get-to-know-more/the-role-of-technology-in-inventory-management/> (08.10.2023)
6. Ресурс інтернету. Evolving a microservice architecture: how to right-size your services. URL: <https://microservices.io/post/architecture/2023/05/17/evolving-microservices-sfjug.html> (17.10.2023)
7. Ресурс інтернету. The microservice architecture is meant to simplify and accelerate development, but only when done correctly. URL: <https://microservices.io/post/architecture/2023/07/06/msa-is-meant-to-simplify-development.md.html> (18.10.2023)
8. Ресурс інтернету. DDD, necessary but insufficient: physical design principles for microservices. URL: <https://microservices.io/post/architecture/2023/04/26/microservices-physical-design-ddd-socal.html> (18.10.2023)
9. Ресурс інтернету. When to use the microservice architecture: part 5 - the monolithic architecture and rapid, frequent, reliable and sustainable software delivery. URL:

- <https://microservices.io/post/microservices/2021/02/14/why-microservices-part-5-monolith.html> (19.10.2023)
10. Ресурс інтернету. Identity and user management in modern applications with Keycloak URL: <https://zenitech.co.uk/insights/articles/identity-and-user-management-in-modern-applications-with-keycloak/> (25.10.2023)
11. Ресурс інтернету. Keycloak SSO – advantages of Single Sign-On and a ready-made access management system. URL: <https://pretius.com/blog/keycloak-sso/> (25.10.2023)
12. Ресурс інтернету. User Authentication with OAuth 2.0. URL: <https://oauth.net/articles/authentication/> (28.10.2023)
13. Ресурс інтернету. Just Use Postgres for Everything. URL: <https://www.amazingcto.com/postgres-for-everything?ref=refind> (30.10.2023)
14. Ресурс інтернету. MongoDB Features. URL: <https://www.mongodb.com/features> (01.11.2023)
15. Ресурс інтернету. LDAP vs. Active Directory: Top 14 Differences You Should Know. URL: <https://www.spiceworks.com/tech/data-management/articles/differences-active-directory-ldap/> (02.11.2023)
16. Ресурс інтернету. Helidon flies faster with GraalVM. URL: <https://medium.com/helidon/helidon-flies-faster-with-graalvm-eea85287d2dc> (02.11.2023)
17. Ресурс інтернету. What are they building—and why? Questions for the top Java architects URL: <https://blogs.oracle.com/javamagazine/post/what-are-they-building-and-why-6-questions-for-the-top-java-architects> (05.11.2023)
18. Ресурс інтернету. How to Avoid Object Mutation in JavaScript. URL: <https://javascript.plainenglish.io/javascript-how-to-avoid-object-mutation-7cd733913a9f> (08.11.2023)
19. Ресурс інтернету. THE ROLE OF API GATEWAYS IN MODERN ENTERPRISE URL: <https://www.smartwaves.com/blog-articles/what-is-an-api-gateway/> (09.11.2023)

- 20.Ресурс інтернету. Overview of best practices for writing Dockerfiles. URL:
https://docs.docker.com/develop/develop-images/dockerfile_best-practices/
(15.11.2023)
- 21.Ресурс інтернету. Why you should use Docker and containers. URL:
<https://www.infoworld.com/article/3310941/why-you-should-use-docker-and-containers.html> (15.11.2023)