

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

перший (бакалаврський)

(рівень вищої освіти)

Розробка автоматизованої системи управління контентом для малого бізнесу  
(тема)

Виконав:

здобувач 4 року навчання,  
групи АКТАКІТ-21-2

Данило Махов

(власне ім'я, прізвище)

Спеціальність 151 Автоматизація,  
комп'ютерно-інтегровані технології та  
робототехніка

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Автоматизація та  
комп'ютерно-інтегровані технології

(повна назва освітньої програми)

Керівник проф. каф. КІТАР, Юрій Ромашов

(посада, власне ім'я, прізвище)

Допускається до захисту  
Зав. кафедри КІТАР

\_\_\_\_\_

(підпис)

Ігор НЕВЛЮДОВ

(власне ім'я, прізвище)

2025 р.

Я, Махов Данило Романович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

"17" червня 2025 р.



Данило Махов

# ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет \_\_\_\_\_ АКТ  
Кафедра \_\_\_\_\_ КІТАР  
Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський)  
Спеціальність \_\_\_\_\_ 151 Автоматизація та комп'ютерно-інтегровані технології  
(код і повна назва)  
Тип програми \_\_\_\_\_ Освітньо-професійна  
Освітня програма \_\_\_\_\_ Автоматизація та комп'ютерно-інтегровані технології  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Махову Данилу Романовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Розробка автоматизованої системи управління контентом для малого бізнеса

Затверджена наказом по університету від 19.05.2025 р. №390

2. Термін подання здобувачем роботи до екзаменаційної комісії 26.06.2025 р.

3. Вихідні дані до роботи \_\_\_\_\_ React, Typescript, Eslint, Vite, Rechart, Redux, Antd, Material UI, Nest, Prisma, PostgreSQL, JWT, Visual Studio Code, документація з відкритих джерел

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

4.1 Аналіз предметної області

4.2 Проектування та розробка

4.3 Реалізація та тестування

4.4 Подальший розвиток та вдосконалення системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

---



---



---



---

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

#### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	19.05.2025	виконано
2	Аналіз предметної області	20.05-23.05	виконано
3	Проектування та розробка	24.05-26.05	виконано
4	Реалізація та тестування	27.05-30.05	виконано
5	Подальший розвиток та вдосконалення системи	31.05-06.06	виконано
6	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	07.07-18.06	виконано
7	Оформлення пояснювальної записки	19.06-20.06	виконано
8	Подання роботи на рецензію	21.06-22.06	виконано
9	Подання роботи на підпис зав. кафедри	23.06-24.06	виконано
11	Подання кваліфікаційної роботи в ЕК	25.06	виконано

Дата видачі завдання 19 травня 2025 р.

Здобувач



(підпис)

Данило Махов

(власне ім'я, прізвище)

Керівник роботи

(підпис)

Юрій Ромашов

(власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка: 95 с., 4 табл., 9 рис., 35 джерел

АВТОМАТИЗОВАНА СИСТЕМА, УПРАВЛІННЯ КОНТЕНТОМ, МАЛИЙ БІЗНЕС, CRM, NESTJS, REACT, TYPESCRIPT, POSTGRESQL, PRISMA ORM, МІКРОСЕРВІСНА АРХІТЕКТУРА, МАШИННЕ НАВЧАННЯ, ХМАРНІ ТЕХНОЛОГІЇ, ЕЛЕКТРОННА КОМЕРЦІЯ, АНАЛІТИКА.

Об'єктом розробки є бізнес-процеси малих підприємств, що потребують автоматизації та централізованого управління для підвищення ефективності діяльності.

Предметом розробки є методи та технології створення автоматизованих систем управління контентом з використанням сучасних веб-технологій та архітектурних підходів.

Мета роботи – розробка комплексної автоматизованої системи управління контентом для малого бізнесу, яка поєднує функціональність CRM, системи управління проектами, платформи електронної комерції та аналітичних інструментів у єдиному, зручному для використання рішенні з використанням сучасних технологій NestJS, React, PostgreSQL та Prisma ORM.

Методи дослідження – аналіз предметної області та існуючих CRM систем, проектування модульної архітектури системи, практична реалізація з застосуванням сучасних веб-технологій, тестування та документування розробленого рішення.

У результаті дослідження було створено повнофункціональну автоматизовану систему управління контентом, що включає модулі автентифікації та авторизації, управління користувачами та організаціями, управління клієнтами та угодами, управління проектами та завданнями, електронної комерції та аналітики. Система забезпечує типобезпечну роботу

з даними, модульну архітектуру для легкого масштабування та сучасний адаптивний користувацький інтерфейс.

Розроблена система демонструє ефективне застосування сучасних технологій веб-розробки для створення масштабованих корпоративних рішень та може бути адаптована під специфічні потреби різних галузей малого бізнесу.

## ABSTRACT

Bachelor's thesis contains: 95 pp., 9 fig., 4 tabl., 35 references.

AUTOMATED SYSTEM, CONTENT MANAGEMENT, SMALL BUSINESS, CRM, NESTJS, REACT, TYPESCRIPT, POSTGRESQL, PRISMA ORM, MICROSERVICE ARCHITECTURE, MACHINE LEARNING, CLOUD TECHNOLOGIES, E-COMMERCE, ANALYTICS.

The object of the study is business processes of small enterprises that require automation and centralized management to improve operational efficiency.

The subject of the study is methods and technologies for creating automated content management systems using modern web technologies and architectural approaches.

The purpose of the work is developing a comprehensive automated content management system for small business that combines CRM functionality, project management system, e-commerce platform and analytical tools in a single, user-friendly solution using modern technologies NestJS, React, PostgreSQL and Prisma ORM.

The research methods are: analysis of the subject area and existing CRM systems, designing modular system architecture, practical implementation using modern web technologies, testing and documentation of the developed solution.

As a result of the study, a fully functional automated content management system was created, including modules for authentication and authorization, user and organization management, customer and deal management, project and task management, e-commerce and analytics. The system provides type-safe data operations, modular architecture for easy scaling and modern adaptive user interface.

The developed system demonstrates effective application of modern web development technologies for creating scalable enterprise solutions and can be

adapted to specific needs of various small business industries.

## ЗМІСТ

Перелік скорочень .....	10
Вступ.....	12
1 Аналіз предметної області.....	14
1.1 Аналіз предметної області.....	14
1.2 Огляд та аналіз аналогічних систем.....	16
1.3 Постановка задачі.....	18
2. Проектування та розробка.....	24
2.1 Специфікація вимог до системи .....	24
2.2 Проектування архітектури системи .....	26
2.3 Вибір технологій для розробки.....	29
2.4 Проектування бази даних .....	32
3. Реалізація та тестування .....	36
3.1 Опис модулів та компонентів серверної частини.....	36
3.2 Опис модулів та компонентів клієнтської частини .....	41
3.3 Інструкція до розробленої системи управління контентом.....	46
4 Подальший розвиток та вдосконалення системи.....	52
4.1 Концепція еволюційного розвитку автоматизованої системи .....	52
4.2 Інтеграція технологій машинного навчання та штучного інтелекту.....	53
4.3 Розширення інтеграційних можливостей та API екосистеми .....	55
4.4 Мобільна платформа та кросплатформенні рішення .....	57
4.5 Хмарна інфраструктура та технології розподіленого розгортання .....	59
Висновки .....	62
Перелік джерел посилання .....	64
Додаток А Програмний код вебдодатка .....	67
Додаток Б Демонстраційний матеріал .....	83

## ПЕРЕЛІК СКОРОЧЕНЬ

- ACID – Atomicity, Consistency, Isolation, Durability – атомарність, узгодженість, ізольованість, довговічність;
- API – Application Programming Interface – програмний інтерфейс;
- AR – Augmented Reality – доповнена реальність;
- AWS – Amazon Web Services – хмарна платформа Amazon;
- CDN – Content Delivery Network – мережа доставки контенту;
- CRM – Customer Relationship Management – система управління взаємовідносинами з клієнтами;
- CRUD – Create, Read, Update, Delete – створення, читання, оновлення, видалення;
- DDD – Domain-Driven Design – предметно-орієнтоване проектування;
- E2E – End-to-End – наскрізне тестування;
- ESB – Enterprise Service Bus – корпоративна сервісна шина;
- GDPR – General Data Protection Regulation – загальний регламент про захист даних;
- GPS – Global Positioning System – система глобального позиціонування;
- HMR – Hot Module Replacement – гаряча заміна модулів;
- HPA – Horizontal Pod Autoscaler – горизонтальний автомасштабувальник подів;
- HTTP – Hypertext Transfer Protocol – протокол передачі гіпертексту;
- IaC – Infrastructure as Code – інфраструктура як код;
- IoT – Internet of Things – інтернет речей;
- IP – Internet Protocol – інтернет протокол;
- JSON – JavaScript Object Notation – запис об'єктів JavaScript;
- JWT – JSON Web Token – веб-токен у форматі JSON;
- KPI – Key Performance Indicators – ключові показники ефективності;
- NFC – Near Field Communication – ближній безконтактний зв'язок;
- NLP – Natural Language Processing – обробка природної мови;

ORM – Object-Relational Mapping – об'єктно-реляційне відображення;  
PWA – Progressive Web Application – прогресивний веб-додаток;  
REST – Representational State Transfer – передача репрезентативного стану;  
ROI – Return on Investment – рентабельність інвестицій;  
RTO – Recovery Time Objective – цільовий час відновлення;  
SDK – Software Development Kit – набір засобів розробки;  
SMTP – Simple Mail Transfer Protocol – простий протокол передачі пошти;  
SQL – Structured Query Language – мова структурованих запитів;  
SSL – Secure Sockets Layer – рівень захищених сокетів;  
TOTP – Time-based One-Time Password – одноразовий пароль на основі часу;  
UI – User Interface – користувацький інтерфейс;  
UPS – United Parcel Service – служба доставки посилок;  
VPA – Vertical Pod Autoscaler – вертикальний автомасштабувальник подів;

## ВСТУП

У сучасному швидкоплинному бізнес-середовищі ефективне управління внутрішніми процесами та взаємовідносинами з клієнтами є критично важливим фактором успіху для підприємств малого бізнесу. Якість організації робочих процесів, своєчасність обробки клієнтських запитів та ефективність координації між різними відділами мають безпосередній вплив на конкурентоспроможність компанії, задоволеність клієнтів та фінансові результати діяльності.

Особливої актуальності ця проблема набуває у контексті цифрової трансформації бізнесу. Малі підприємства часто стикаються з викликами децентралізованого управління інформацією, неефективної координації між продажами, маркетингом та обслуговуванням клієнтів, а також відсутності єдиної системи для відстеження бізнес-процесів. Традиційні методи управління з використанням розрізаних інструментів – електронних таблиць, окремих додатків для різних завдань, паперового документообігу – часто призводять до втрати критичної інформації, дублювання зусиль та неможливості отримання цілісного бачення стану бізнесу.

Розвиток сучасних веб-технологій, зокрема фреймворків NestJS та React, систем управління базами даних PostgreSQL, а також інструментів для створення типобезпечних додатків на базі TypeScript, відкрив нові можливості для створення комплексних систем управління бізнес-процесами. Проте більшість існуючих CRM (Customer Relationship Management) рішень орієнтовані на великі корпорації з відповідними бюджетами та IT-ресурсами, або ж пропонують обмежений функціонал, який не покриває всі потреби сучасного малого бізнесу.

Популярні рішення, такі як HubSpot, Pipedrive або Zoho CRM, часто мають високу вартість володіння для малого бізнесу, складні процеси налаштування або обмежені можливості кастомізації під специфічні потреби конкретної організації. Крім того, більшість таких систем не

інтегрують повний спектр бізнес-процесів – від управління клієнтами до проектного менеджменту та електронної комерції – в єдиній платформі, що вимагає використання декількох різних інструментів.

У цьому контексті виникає потреба у створенні комплексної автоматизованої системи управління контентом, яка поєднає функціональність CRM, системи управління проектами, платформи електронної комерції та аналітичних інструментів у єдиному, зручному для використання рішенні. Така система повинна бути доступною для малого бізнесу, легко налаштовуватися під різні сфери діяльності та забезпечувати високий рівень безпеки і масштабованості.

Дане дослідження присвячене розробці автоматизованої системи управління контентом для малого бізнесу, яка використовує сучасні технології для забезпечення комплексного управління всіма аспектами діяльності організації. У межах проєкту було розроблено модульну систему на основі NestJS backend архітектури з використанням PostgreSQL та Prisma ORM (Object-Relational Mapping) для надійного зберігання даних, а також React frontend з Material-UI (User Interface) для створення інтуїтивного користувацького інтерфейсу. Такий підхід забезпечує не лише технічну ефективність, але й доступність системи для користувачів без глибоких технічних знань, сприяючи підвищенню продуктивності малого бізнесу в умовах сучасної цифрової економіки.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Аналіз предметної області

У сучасних ринкових умовах малий бізнес зіштовкується зі значними викликами в організації управлінських процесів з чого випливає що ефективність діяльності підприємства залежить від його здатності оптимізувати взаємодію з клієнтами, адмініструванням проєктів, обробкою замовлень та проведенням аналітичної роботи. Застосування автоматизованих систем, зокрема систем управління взаємовідносинами з клієнтами (Customer Relationship Management, CRM), сприяє систематизації та вдосконаленню цих процесів, що, у свою чергу, призводить до зростання продуктивності та посилення ринкових позицій компанії.

Уявімо ситуацію, коли власник невеликої ІТ-компанії намагається відстежувати прогрес декількох проєктів одночасно, керувати командою розробників, обробляти замовлення клієнтів та аналізувати показники продажів. Без автоматизованої системи це перетворюється на хаотичний процес, де легко втратити важливу інформацію або пропустити критичні терміни. Саме тут на допомогу приходить CRM система, яка централізує всі бізнес-процеси в одному місці.

Інший приклад: маркетингове агентство працює з десятками клієнтів, кожен з яких має свої унікальні потреби та проєкти. Менеджери потребують швидкого доступу до історії взаємодії з клієнтами, статусу поточних угод, завдань команди та аналітичних звітів. Без централізованої системи координація між відділами стає складною, що може призвести до втрати клієнтів або зниження якості послуг.

Автоматизовані системи управління контентом для малого бізнесу надають широкий спектр функцій, таких як управління клієнтською базою, відстеження угод, планування та контроль проєктів, управління завданнями команди, обробка замовлень та товарів, а також аналітика продажів та

поведінки клієнтів. Такі системи також забезпечують різні рівні доступу для співробітників залежно від їх посад у компанії, що гарантує безпеку даних та ефективний розподіл відповідальності.

Сучасні CRM системи інтегруються з електронною комерцією, дозволяючи керувати каталогом товарів, обробляти замовлення та відстежувати склад. Це особливо важливо для підприємств, які поєднують надання послуг з продажем товарів. Аналітичні можливості таких систем дозволяють власникам бізнесу приймати обґрунтовані рішення на основі даних про продаж товарів або послуг та поведінки клієнтів.

Таким чином, автоматизовані системи управління контентом є необхідним інструментом для ефективного ведення малого бізнесу в умовах сучасної ринкової економіки.

Важливим аспектом сучасних автоматизованих систем є їх здатність до інтеграції з зовнішніми сервісами та платформами. Для малого бізнесу це означає можливість синхронізації з банківськими системами для автоматичного імпорту фінансових операцій, інтеграцію з електронною поштою для автоматичного ведення листування з клієнтами, підключення до соціальних мереж для маркетингових цілей та взаємодію з платіжними системами для обробки онлайн-платежів.

Окремої уваги заслуговує питання масштабованості таких систем. Малий бізнес часто характеризується швидкими темпами зростання, що вимагає від автоматизованих систем здатності до горизонтального та вертикального масштабування без значних змін в архітектурі. Сучасні рішення повинні підтримувати мультитенантну архітектуру, яка дозволяє одній системі обслуговувати декілька організацій з повним розмежуванням їх даних та налаштувань.

Безпека даних у таких системах набуває критичного значення, особливо з урахуванням вимог сучасного законодавства про захист персональних даних. Системи повинні включати багаторівневу автентифікацію, шифрування даних як при зберіганні, так і при передачі,

регулярні резервні копії з можливістю швидкого відновлення, а також детальне логування всіх дій користувачів для забезпечення аудиту безпеки.

## 1.2 Огляд та аналіз аналогічних систем

Перед розробкою власної системи було проаналізовано три провідні CRM системи, які орієнтовані на малий та середній бізнес.

Першою для аналізу була обрана HubSpot CRM – це комплексна платформа для управління клієнтськими відносинами, розроблена компанією HubSpot. Запущена у 2014 році, система пропонує безкоштовну базову версію та платні тарифи з розширеною функціональністю. HubSpot CRM використовується понад 100 000 компаній по всьому світу.

HubSpot пропонує безкоштовний план з базовими функціями управління контактами, угодами та завданнями, а також платні плани з розширеними можливостями автоматизації маркетингу та продажів. Основні функції включають в себе управління контактами, відстеження угод через воронку продажів, управління завданнями та нагадуваннями, електронний маркетинг, аналітику та надання звітів. Система також пропонує інтеграцію з понад 500 сторонніми додатками та сервісами.

HubSpot доступний у веб-браузерах та мобільних додатках для iOS та Android. Платформа забезпечує високий рівень безпеки даних, включаючи SSL-шифрування (Secure Sockets Layer) та регулярні резервні копії. Інтерфейс системи інтуїтивний та зручний для користувачів без технічної підготовки.

Другою системою для аналізу є Pipedrive CRM – естонська CRM платформа, заснована у 2010 році Тімо Рейном та командою колишніх співробітників Skype. Pipedrive орієнтована на малий та середній бізнес і відома своїм простим та ефективним підходом до управління продажами.

Станом на 2024 рік системою користуються понад 100 000 компаній у 179 країнах.

Pipedrive пропонує кілька тарифних планів, починаючи від базового плану для невеликих команд до розширених планів для великих організацій. Ключові функції включають в себе візуальну воронку продажів, автоматизацію процесів, управління контактами та угодами, планувальник активностей, мобільні додатки, інтеграцію з електронною поштою. Система також надає детальну аналітику продажів та можливості їх прогнозування.

Pipedrive доступний через веб-інтерфейс та мобільні додатки для Android та iOS. Безпека забезпечується шифруванням даних, регулярними аудитами безпеки та відповідністю європейським стандартам захисту даних GDPR (General Data Protection Regulation).

Третьою системою для аналізу є Zoho CRM – частина екосистеми Zoho, розробленої індійською компанією Zoho Corporation. Заснована у 1996 році, Zoho CRM є однією з найстаріших та найпопулярніших CRM систем на ринку. Систему використовують понад 250 000 бізнесів по всьому світу.

Zoho CRM пропонує безкоштовний план для команд до 3 користувачів та різні платні плани з додатковим функціоналом. Основні можливості включають в себе управління потенційними клієнтами, контактами та угодами, автоматизацію маркетингу та продажів, управління територіями та командами, інтеграцію з іншими продуктами Zoho, штучний інтелект для прогнозування продажів. Система також підтримує налаштування робочих процесів та бізнес-правил.

Zoho CRM доступний у веб-браузерах та мобільних додатках. Платформа забезпечує високий рівень безпеки, включаючи двофакторну автентифікацію, шифрування даних та відповідність міжнародним стандартам безпеки.

Аналіз цих трьох CRM систем показує важливість інтуїтивного інтерфейсу, гнучкості налаштувань, інтеграційних можливостей та мобільного доступу для успішного впровадження в малому бізнесі.

Детальний аналіз цих систем також показав важливість підтримки різних моделей розгортання. HubSpot CRM працює виключно в хмарі, що забезпечує легкість впровадження але може викликати занепокоєння щодо безпеки даних у деяких організацій. Pipedrive пропонує гібридні рішення, дозволяючи зберігати критичні дані локально. Zoho CRM надає найбільш гнучкі варіанти розгортання, включаючи приватні хмари.

Важливим фактором успіху цих систем є їх здатність до кастомізації під специфічні потреби бізнесу. Всі три системи підтримують створення користувацьких полів, налаштування робочих процесів та створення власних звітів. Однак рівень кастомізації суттєво відрізняється: HubSpot надає візуальні інструменти для створення складних автоматизованих воронки, Pipedrive фокусується на простоті налаштування процесів продажів, а Zoho CRM пропонує найглибші можливості програмування через власну платформу розробки.

Аналіз показників використання цих систем демонструє різні підходи до монетизації: HubSpot використовує freemium модель з обмеженнями на кількість контактів, Pipedrive пропонує фіксовану місячну плату за користувача, а Zoho CRM поєднує безкоштовний план з гнучкими тарифами для команд різного розміру. Це впливає на загальну вартість володіння системою та робить кожне рішення більш підходящим для певних типів організацій.

### 1.3 Постановка задачі

Завдання на розроблення програмного забезпечення: автоматизована система управління контентом для малого бізнесу

Основні модулі системи:

- модуль управління користувачами та організаціями;
- модуль управління клієнтами та контактами;
- модуль управління угодами та продажами;

- модуль управління проектами та завданнями;
- модуль електронної комерції;
- модуль аналітики та звітності.

Для доступу до системи користувачі мають пройти автентифікацію за допомогою електронної пошти та пароля. Система повинна підтримувати різні ролі користувачів з відповідними правами доступу: адміністратор, HR менеджер, контент менеджер, менеджер веб-сайту, менеджер магазину, маркетинг менеджер, аналітик, звичайний менеджер та користувач.

Необхідно розробити інтуїтивний та функціональний користувацький інтерфейс, що забезпечить ефективну роботу з усіма модулями системи. Інтерфейс має бути адаптивним та зручним для використання на різних пристроях. Далі будуть наведено вимоги до кожного модуля.

### 1.3.1 Модуль автентифікації та авторизації

Система повинна забезпечувати надійний механізм автентифікації користувачів через реєстрацію користувача та обов'язковою прив'язкою до організації. Користувачі зможуть входити в систему, використовуючи свою електронну пошту та пароль. Важливою складовою є реалізація розмежування доступу на основі ролей користувачів, що дозволить контролювати, які функції та дані доступні різним категоріям співробітників. Система також має включати ефективне управління сесіями та забезпечувати високий рівень безпеки для захисту корпоративних даних.

### 1.3.2 Модуль управління клієнтами

Модуль надаватиме можливість створювати та редагувати детальні картки клієнтів для фізичних та юридичних осіб. Система має дозволяти ефективно управляти всією контактною інформацією, включаючи номери телефонів, електронні адреси, адреси та іншою важливою інформацією.

Користувачі зможуть додавати теги для розбиття на категорії клієнтів та створювати користувацькі поля для зберігання специфічної інформації. Важливою функцією є ведення повної історії взаємодії з кожним клієнтом, що допоможе відстежувати всі контакти, зустрічі та угоди.

### 1.3.3 Модуль управління угодами

Модуль буде забезпечувати повний цикл управління угодами від створення до успішного завершення. Система включатиме в себе воронку продажів з можливістю налаштування різних стадій відповідно до специфіки бізнесу. Буде реалізовано функціонал прогнозування доходів, що допомагає планувати фінансові результати. Кожна угода може мати призначеного відповідального менеджера, що забезпечує персональну відповідальність за результат. Модуль також буде включати в себе розширену аналітику ефективності продажів для оптимізації процесів та підвищення конверсії.

### 1.3.4 Модуль управління проектами

Система має дозволяти створювати проекти з можливістю залучення команди та розподілу ролей. Необхідно реалізувати комплексне планування та контроль виконання проектів з деталізацією кожного етапу. Користувачі зможуть управляти завданнями з встановленням пріоритетів та термінів виконання. Система забезпечуватиме відстеження бюджету проекту та контроль дотримання встановлених термінів. Буде зроблений функціонал генерації детальної звітності по проектах для аналізу ефективності та виявлення проблем.

### 1.3.5 Модуль електронної комерції

Модуль надаватиме повний функціонал для управління каталогом товарів з можливістю додавання детальних описів, зображень та

характеристик. Система має забезпечувати ефективну обробку замовлень клієнтів від отримання до виконання. Буде реалізовано контроль залишків на складі з автоматичними попередженнями про необхідність поповнення. Система також має підтримувати роботу з різними валютами для міжнародної діяльності через інтеграцію з популярними платіжними системами для забезпечення зручних способів оплати.

### 1.3.6 Модуль аналітики

Система здійснюватиме автоматичний збір та аналіз даних про активність користувачів для оптимізації роботи. Необхідно зробити генерацію детальних звітів з продажів та клієнтів з можливістю фільтрації за різними параметрами. Буде реалізовано аналітику ефективності маркетингових кампаній для оцінки ROI (Return on Investment) та оптимізації стратегій. Кожен користувач зможе створювати персоналізовані дашборди з необхідними віджетами та показниками. Система підтримуватиме експорт даних у різних форматах для подальшої обробки в зовнішніх системах.

### 1.3.6 Основні вимоги до архітектури

Система повинна бути побудована на мікросервісній архітектурі, що забезпечить високу масштабованість та гнучкість розвитку. Взаємодія між компонентами здійснюється через RESTful API (Application Programming Interface) з дотриманням стандартів. Обов'язковою є реалізація типобезпечності та валідації всіх даних на всіх рівнях системи. Система буде включати в себе комплексне логування та моніторинг для відстеження роботи та швидкого виявлення проблем. Передбачено надійне резервне копіювання та процедури відновлення даних. Система забезпечуватиме підтримку транзакцій та консистентність даних у розподіленому середовищі.

### 1.3.7 Основні вимоги до інтерфейсу

Інтерфейс системи повинен мати повністю адаптивний дизайн, що забезпечить комфортну роботу на мобільних пристроях та планшетах. Навігація між модулями має бути інтуїтивно зрозумілою навіть для нових користувачів. Буде реалізований швидкий пошук та розширені можливості фільтрації даних для ефективної роботи з великими обсягами інформації. Користувачі зможуть налаштовувати власні дашборди та віджети відповідно до своїх потреб. Система підтримуватиме як темну, так і світлу тему оформлення для комфорту користувачів. Також, буде передбачена багатомовна підтримка для роботи міжнародних команд.

### 1.3.8 Технічні вимоги

Система розробляється з використанням найсучасніших веб-технологій та фреймворків. Забезпечується повна сумісність з усіма сучасними браузерами та їх останніми версіями. Буде реалізовано найвищий рівень безпеки даних відповідно до міжнародних стандартів. Система оптимізована для високої продуктивності навіть при роботі з великими обсягами даних. Архітектура дозволить легке масштабування системи відповідно до зростання бізнесу та кількості користувачів.

Розробка автоматизованої системи управління контентом має забезпечити комплексне рішення для ефективного ведення малого бізнесу з можливістю адаптації під специфічні потреби різних галузей.

Система передбачає розширену підтримку міжнародних стандартів та регуляторних вимог. Модуль управління користувачами включає функціонал для забезпечення відповідності GDPR з можливістю експорту персональних даних, їх видалення на запит та ведення детального аудиту обробки. Система

автоматично класифікує персональні дані та застосовує відповідні політики безпеки залежно від типу інформації та географічного розташування користувача.

Важливою складовою є підтримка різних часових зон та культурних особливостей. Система автоматично адаптує формати дат, валют та числових значень відповідно до локальних налаштувань користувача. Це включає не тільки відображення інформації, але й правильну обробку вхідних даних з урахуванням локальних форматів. Календарна система враховує національні свята та робочі дні різних країн для коректного планування завдань та розрахунку термінів виконання.

Система також передбачає модульність впровадження, дозволяючи організаціям поетапно активувати необхідні функції. Це особливо важливо для малого бізнесу з обмеженими ресурсами, які можуть почати з базових модулів управління клієнтами та поступово розширювати функціональність. Кожен модуль має власні налаштування доступу та може бути тимчасово відключений без впливу на роботу інших частин системи.

## 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА

### 2.1 Специфікація вимог до системи

Для малого бізнесу автоматизована система управління контентом повинна задовольняти різноманітні функціональні та технічні вимоги, щоб забезпечити ефективну роботу різних відділів компанії. Система розмежовує доступ за допомогою ролей користувачів як модульний веб-додаток.

#### 2.1.1 Загальні характеристики системи

Для того, щоб працівники з різних відділів компанії могли ефективно виконувати свою роботу, інтерфейс системи повинен бути як простим, так і адаптивним. Незалежно від платформи доступу система повинна працювати на всіх сучасних браузерях і мобільних пристроях.

Серверна частина повинна правильно обробляти бізнес-логіку, валідувати вхідні дані та коректно обробляти помилки, надаючи користувачам чіткі повідомлення про них. Для захисту корпоративних даних система розмежовує права доступу залежно від ролі користувача.

#### 2.1.2 Нефункціональні вимоги

Для того, щоб підтримувати розвиток бізнесу та збільшення кількості користувачів, система повинна забезпечувати високу масштабованість і продуктивність. Відповідно до сучасних стандартів кібербезпеки забезпечено найвищий рівень безпеки даних, з підтримкою шифрування та аудиту дій користувачів.

Комплексне тестування, моніторинг системи та резервне копіювання даних гарантують надійність і стабільність роботи. Дизайн системи

передбачає легке адміністрування та можливість інтеграції з популярними зовнішніми сервісами за допомогою API.

### 2.1.3 Функціональні вимоги

Система забезпечує повний цикл управління організацією та користувачами за допомогою ієрархічної структури та розподілу повноважень. Модуль управління базою клієнтів дозволяє вести профілі клієнтів, історії взаємодії та сегментацію аудиторії.

Функціонал ведення угод включає воронку продажів з налаштовуваними етапами, прогнозування доходів та аналітику ефективності. Управління проектами та завданнями реалізовано з підтримкою планування, розподілу ресурсів та контролю термінів виконання. Модуль електронної комерції забезпечує управління каталогом товарів, обробку замовлень та контроль складських залишків. Система аналітики та звітності надає детальні інсайти про роботу всіх напрямків бізнесу з можливістю експорту даних.

Система повинна також забезпечувати розширені можливості персоналізації користувацького досвіду. Кожен користувач має можливість налаштувати власний робочий простір, включаючи вибір віджетів для головної панелі, налаштування фільтрів за замовчуванням, персоналізацію кольорової схеми та розташування елементів інтерфейсу. Ці налаштування зберігаються в профілі користувача та синхронізуються між різними пристроями.

Важливою складовою системи є підтримка багатомовності та локалізації. Інтерфейс повинен автоматично адаптуватися до мови користувача, підтримувати різні формати дат, часу та валют залежно від географічного розташування організації. Система також повинна враховувати локальні особливості ведення бізнесу та вимоги законодавства різних країн.

Для забезпечення безперервності роботи система включає механізми автоматичного збереження даних, відновлення після збоїв та резервного копіювання. Користувачі мають можливість працювати в offline режимі з автоматичною синхронізацією при відновленні з'єднання з інтернетом. Це особливо важливо для мобільних користувачів та співробітників, які часто працюють в умовах нестабільного інтернет-з'єднання.

## 2.2 Проектування архітектури системи

Система використовує мікросервісну архітектуру та має модульну структуру, що забезпечує гнучкість під час розробки та подальшого масштабування. Кожен модуль є окремим функціональним блоком із своєю власною бізнес-логікою, що дозволяє незалежно розвивати та підтримувати окремі частини системи.

Взаємодія між компонентами здійснюється через RESTful API з дотриманням принципів REST (Representational State Transfer) архітектури. Реалізовано централізоване управління конфігурацією, логування та моніторинг для забезпечення прозорості роботи системи та швидкого виявлення проблем.

Архітектура системи побудована з урахуванням принципів DDD (Domain-Driven Design), що дозволяє чітко відокремити бізнес-логіку від технічних деталей реалізації. Кожен модуль інкапсулює власну доменну модель з відповідними бізнес-правилами та валідаціями, що спрощує розуміння та підтримку коду.

Система використовує асинхронну обробку повідомлень через черги завдань для операцій, які не потребують миттєвого виконання. Це включає відправку електронних листів, генерацію звітів, обробку файлів та синхронізацію з зовнішніми системами. Така архітектура забезпечує високу відгукуваність системи навіть при значних навантаженнях.

Для оптимізації продуктивності застосовується багаторівневе кешування: кешування на рівні додатка для часто запитуваних даних, кешування запитів до бази даних та кешування на рівні HTTP (Hypertext Transfer Protocol) для статичних ресурсів. Система також включає механізми інвалідації кешу при оновленні даних для забезпечення консистентності.

Моніторинг та логування реалізовані через централізовану систему збору метрик, що дозволяє відстежувати продуктивність кожного модуля, виявляти вузькі місця та прогнозувати потреби в масштабуванні. Система реалізує розширені принципи мікросервісної архітектури з використанням Domain-Driven Design підходу. Кожен модуль інкапсулює власну доменну логіку та має чітко визначені границі контексту, що дозволяє командам розробників працювати незалежно над різними частинами системи. Взаємодія між модулями здійснюється виключно через добре визначені API інтерфейси, що забезпечує слабку зв'язаність та високу когезію.

AuthModule забезпечує не тільки базову автентифікацію через JWT (JSON Web Token) токени, але й реалізує розширені механізми безпеки. Система підтримує refresh токени з автоматичним оновленням для забезпечення безперервної роботи користувачів, rate limiting для запобігання атакам грубої сили, та session management з можливістю примусового завершення всіх сеансів користувача. Додатково реалізовано механізм збереження контексту користувача для відновлення роботи після переривання з'єднання.

Модуль має власну систему логування безпекових подій, включаючи невдалі спроби входу, підозрілу активність та зміни в правах доступу. Всі дії користувачів фіксуються з відмітками часу, IP (Internet Protocol) адресами та додатковими метаданими для забезпечення повного аудиторського сліду. Система автоматично виявляє аномальну поведінку та може тимчасово блокувати акаунти при підозрі на компрометацію.

UsersModule реалізує складну ієрархічну систему ролей з наслідуванням дозволів. Кожна роль може включати набір базових дозволів

та додаткові специфічні права для конкретних модулів. Система підтримує як статичні ролі (ADMIN, MANAGER, USER), так і динамічні ролі, створені адміністраторами організації для специфічних потреб. Користувачі можуть мати декілька ролей одночасно, при цьому система автоматично об'єднує всі дозволи. Система автоматично генерує сповіщення при виявленні аномалій або перевищенні встановлених порогів продуктивності.

### 2.2.1 Модульна структура

AuthModule відповідає за автентифікацію та авторизацію користувачів з використанням JWT токенів, безпечного хешування паролів і refresh токенів. Модуль дозволяє керувати сесіями та запобігати несанкціонованому доступу.

UsersModule управляє користувачами системи, їх профілями та ролями. Підтримує створення облікових записів, редагування профілів, призначення ролей та налаштування прав доступу відповідно до організаційної структури.

OrganizationsModule забезпечує управління організаціями та їх налаштуваннями, дозволяючи створювати багаторівневу структуру компаній з різними параметрами конфігурації для кожної організаційної одиниці.

CustomersModule реалізує повний функціонал управління клієнтською базою з підтримкою як індивідуальних клієнтів, так і компанії з контактними особами, історією взаємодії та системою тегування.

DealsModule дозволяє вести угоди та воронку продажів з етапами, які можна власноруч налаштувати, а також робити прогнози доходів, аналітику конверсії та звіти по ефективності продажів.

ProjectsModule реалізує керування проектами та завданнями, надаючи підтримку для планування, розподілу ресурсів, контролю за виконанням завдань і створенням детальної звітності по проектах.

OrdersModule виконує обробку замовлень та забезпечує функціонал електронної комерції, включаючи управління каталогом товарів, обробку платежів та відстеження статусів замовлень.

AnalyticsModule збирає, обробляє та аналізує дані з усіх модулів системи, створюючи комплексні звіти та допомагаючи у прийнятті управлінських рішень.

### 2.3 Вибір технологій для розробки

NestJS обрано як основний фреймворк для розробки backend частини завдяки його прогресивному підходу до створення масштабованих серверних додатків. Фреймворк забезпечує модульну архітектуру з використанням декораторів та dependency injection, вбудовану підтримку TypeScript для типобезпечності та автоматичну валідацію даних.

Основними перевагами NestJS є підтримка багатьох транспортних протоколів, інтеграція з поширеними ORM, вбудовані засоби автентифікації та авторизації, middleware та засоби захисту доступу, а також автоматична генерація документації Swagger.

Prisma ORM є сучасним інструментом для роботи з базами даних, який надає безпечний доступ до даних і спрощує управління схемою бази даних. Prisma пропонує декларативну схему з оптимізацією продуктивності, автоматичними міграціями, типобезпечними запитами та автогенерацією TypeScript типів.

PostgreSQL обрано як основну систему управління базами даних завдяки потужним можливостям для зберігання бізнес-критичних даних. СУБД забезпечує повну підтримку ACID (Atomicity, Consistency, Isolation, Durability) транзакцій, розширені типи даних, потужну систему індексування та відповідність SQL (Structured Query Language) стандартам.

### 2.3.1 Клієнтська частина

React використовується для створення динамічного користувацького інтерфейсу, застосовуючи компонентний підхід та ефективно оновлення через віртуальний DOM. Бібліотека забезпечує односпрямований потік даних, підтримку TypeScript та великий екосистему додаткових інструментів.

Redux Toolkit забезпечує централізоване управління станом додатка зі спрощеним синтаксисом для створення actions та reducers, вбудованими middleware для асинхронних операцій та інтеграцією з Redux DevTools для дебагінгу.

TypeScript забезпечує статичну типізацію на всіх рівнях додатка, що значно знижує кількість помилок під час виконання та підвищує продуктивність розробки. Використання строгих налаштувань TypeScript гарантує повну типобезпечність API-викликів, валідацію форм та управління станом додатка.

Для оптимізації збірки та завантаження використовується Vite з підтримкою HMR (Hot Module Replacement), що забезпечує миттєве оновлення змін під час розробки. Продакшн збірка включає автоматичну оптимізацію коду, tree-shaking для видалення невикористаного коду, компресію зображень та lazy loading для покращення швидкості завантаження.

Система тестування включає unit тести для бізнес-логіки, integration тести для API endpoints та end-to-end тести для критичних користувацьких сценаріїв. Використовуються Jest для unit тестів, Supertest для тестування API та Playwright для E2E (End-to-End) тестування. Всі тести автоматично виконуються при кожному commit через GitHub Actions.

Для забезпечення якості коду налаштовані ESLint з суворими правилами, Prettier для автоматичного форматування, Husky для pre-commit hooks та SonarQube для статичного аналізу коду. Це гарантує дотримання кодстайлу та виявлення потенційних проблем на ранніх стадіях розробки.

Архітектура клієнтської частини побудована на принципах Clean Architecture з чітким розділенням між presentation, business та data layers. Компоненти React організовані в ієрархічну структуру з використанням композиції для забезпечення переважного використання коду. Кожен компонент має власні типи TypeScript, що забезпечує повну типобезпечність на рівні інтерфейсу.

Redux Toolkit Query забезпечує потужне кешування даних з автоматичною інвалідацією при змінах на сервері. Система використовує оптимістичні оновлення для покращення відгукуваності інтерфейсу - зміни відображаються миттєво, а у випадку помилки автоматично відкочуються до попереднього стану. Middleware для обробки помилок забезпечує централізоване управління станами завантаження та помилок по всій аплікації.

Material-UI компоненти розширені власними стилізованими версіями для забезпечення консистентного дизайну. Система темізації включає не тільки кольори та типографіку, але й анімації, тіні та просторові відношення. Компоненти автоматично адаптуються до розміру екрану, типу пристрою та налаштувань доступності користувача.

Система державного управління включає middleware для синхронізації з localStorage, автоматичного збереження форм та відновлення після перезавантаження сторінки. Складні форми використовують бібліотеку React Hook Form з інтеграцією Yup для валідації, що забезпечує високу продуктивність навіть при роботі з великими формами.

Material-UI надає готові React компоненти відповідно до принципів Material Design, систему темізації, адаптивний дизайн та підтримку доступності з коробки.

## 2.4 Проектування бази даних

База даних спроектована з урахуванням принципів нормалізації та оптимізації для високої продуктивності. Використовується PostgreSQL з підтримкою складних запитів, транзакцій та повнотекстового пошуку.

Таблиця Users збирає та зберігає дані про користувачів системи, такі як унікальний ідентифікатор, електронна адреса для автентифікації, персональні дані, роль у системі та зв'язок з організацією. У цій таблиці передбачено поля для контролю активності та мітки часу створення і оновлення записів.

Таблиця Organizations управляє інформацією про організації з полями для назви, опису діяльності, JSON (JavaScript Object Notation) налаштувань та міток часу. Структура дозволяє створювати ієрархічні зв'язки між організаційними одиницями.

Таблиця Customers містить клієнтську базу з підтримкою як індивідуальних клієнтів, так і компаній. Включає контактну інформацію, тип клієнта, систему тегування та призначення відповідального менеджера.

Таблиця Deals зберігає інформацію про угоди з полями для суми, валюти, етапу в воронці продажів, ймовірності закриття та очікуваної дати завершення. Реалізовано зв'язки з клієнтами та відповідальними менеджерами.

Таблиця Projects управляє проектами з інформацією про назву, опис, статус виконання, терміни та бюджет. Передбачено зв'язок з клієнтами та можливість прикріплення завдань.

Таблиця Tasks зберігає завдання з детальним описом, статусом, пріоритетом, термінами виконання та призначеним виконавцем. Реалізовано ієрархічні зв'язки з проектами.

### 2.4.1 Оптимізація та індексування

База даних була оптимізована за допомогою створення індексів на часто використовуваних полях для пошуку та фільтрації. Реалізовано повнотекстовий пошук для ефективного знаходження інформації в текстових полях. Каскадні операції та обмеження зовнішніх ключів гарантують цілісність даних.

Композитні індекси підвищують продуктивність і оптимізують складні запити. Для забезпечення оптимальної швидкості роботи системи використовується стратегія архівування історичних даних.

Система включає розширений механізм soft delete для збереження цілісності даних. Замість фізичного видалення записів, система позначає їх як видалені з можливістю відновлення протягом певного періоду. Це особливо важливо для критичних бізнес-даних як клієнтські записи, угоди та фінансові транзакції. Адміністратори можуть налаштувати період збереження видалених даних для кожного типу сутності.

Таблиця аудиту (audit\_logs) автоматично створює записи для всіх операцій CRUD (Create, Read, Update, Delete) з детальною інформацією про зміни. Кожен запис включає старе та нове значення поля, timestamp операції, ідентифікатор користувача та додаткові метадані як IP адреса та user agent. Це забезпечує повну трасуєміст всіх змін даних для відповідності регуляторним вимогам.

Система підтримує масові операції для ефективної обробки великих обсягів даних. Batch processing дозволяє імпортувати тисячі записів клієнтів або продуктів за одну операцію з автоматичною валідацією та обробкою помилок. Для особливо великих операцій використовуються фонові задачі з можливістю відстеження прогресу та отримання звітів про результати.

Васкур стратегія включає щоденні інкрементальні та тижневі повні резервні копії з автоматичним тестуванням відновлення. Система автоматично перевіряє цілісність резервних копій та повідомляє про будь-які

проблеми. Для критичних організацій доступне real-time реплікування на резервний сервер з автоматичним failover у випадку відмови основної системи.

База даних також включає спеціальні таблиці для аудиту, які автоматично зберігають історію всіх змін критичних даних. Кожна зміна фіксується з інформацією про користувача, час зміни та тип операції, що забезпечує повний аудиторський слід для відповідності регуляторним вимогам.

Для підвищення продуктивності реалізовано горизонтальне партиціонування великих таблиць за часовими періодами. Історичні дані автоматично переносяться в архівні секції, що зменшує розмір активних таблиць та прискорює виконання запитів. Система також підтримує автоматичне створення та видалення індексів на основі аналізу частоти запитів.

Реплікація бази даних налаштована за схемою master-slave для забезпечення високої доступності та розподілення навантаження читання. Автоматичний failover гарантує переключення на резервний сервер у випадку недоступності основного. Регулярні перевірки цілісності даних та автоматичне відновлення пошкоджених індексів забезпечують надійність системи.

Проектування автоматизованої системи управління контентом здійснено з урахуванням сучасних принципів розробки enterprise рішень. Визначено комплексну специфікацію вимог, що охоплює функціональні та нефункціональні аспекти системи. Розроблено модульну архітектуру з чітким розділенням відповідальності між компонентами.

Обґрунтовано вибір технологічного стеку, що включає NestJS, React, Redux Toolkit, Prisma ORM та PostgreSQL для забезпечення масштабованості, типобезпечності та високої продуктивності. Спроектовано оптимізовану структуру бази даних з урахуванням нормалізації та індексування.

Така архітектура забезпечує можливість незалежного розвитку модулів, спрощує підтримку системи та дозволяє ефективно масштабувати функціонал відповідно до потреб бізнесу. Обрані технології та проектні рішення створюють надійну основу для реалізації конкурентоспроможного продукту для малого бізнесу.

## 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

### 3.1 Опис модулів та компонентів серверної частини

Система побудована за принципами модульної архітектури з використанням NestJS фреймворку. Основний файл запуску системи (лістинг А.1) демонструє конфігурацію CORS, валідацію даних та автоматичну генерацію API документації через Swagger. Кожен модуль відповідає за окрему бізнес-область та має власний набір контролерів, сервісів та моделей даних. Модульна архітектура системи визначається у головному модулі AppModule (лістинг А.2), який імпортує всі функціональні модулі. Такий підхід забезпечує слабку зв'язаність компонентів та високу згуртованість всередині модулів.

Модульна архітектура – це метод розробки програмного забезпечення, який ділить систему на окремі, незалежні частини або модулі. Кожен модуль виконує конкретну функцію та може працювати самостійно, що дозволяє йому бути окремим від інших модулів системи.

Основні переваги модульної архітектури включають:

- легкість підтримки та розширення функціональності;
- можливість незалежної розробки різних модулів;
- покращена тестованість кожного компонента;
- зменшення складності системи через розділення відповідальності.

Prisma схема визначає структуру бази даних та взаємозв'язки між сутностями, як зображено на рис. 3.1. В файлі schema.prisma, визначені основні енуми системи, зокрема UserRole який містить всі можливі ролі користувачів системи. Детальна структура бази даних з усіма моделями, енумами та зв'язками представлена в лістингу А.6.

Основні енуми системи включають:

- UserRole: визначає ролі користувачів від адміністратора до звичайного користувача;
- OrderStatus: статуси замовлень від очікування до доставки;
- CustomerStatus: статуси клієнтів для сегментації бази;
- DealStage: етапи угод у воронці продажів;
- ProjectStatus: статуси проектів для контролю виконання;
- TaskStatus та TaskPriority: для управління завданнями.

```
// This is your Prisma schema file

Generate
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

// Common enums
enum UserRole {
  ADMIN
  HR_MANAGER
  CONTENT_MANAGER
  WEBSITE_MANAGER
  STORE_MANAGER
  MARKETING_MANAGER
  ANALYST
  MANAGER
  USER
}

enum OrderStatus {
  PENDING
  CONFIRMED
  PROCESSING
  SHIPPED
  DELIVERED
  CANCELLED
  REFUNDED
}

// Analytics Module
enum AnalyticsSource {
  USER
  SYSTEM
  SALES
  MARKETING
  WEBSITE
}

enum AnalyticsEventType {
  PAGE_VIEW
  CLICK
  PURCHASE
  SIGNUP
  LOGIN
  CUSTOM
}
```

Рисунок 3.1 – Схема бази даних системи, яка реалізована за допомогою Prisma ORM

DatabaseService, який зображений на рис. 3.2, розширює PrismaClient та реалізує патерн Singleton для забезпечення єдиного екземпляра підключення

до бази даних. Цей сервіс включає обробку помилок та логування для надійної роботи з даними. Централізований доступ до бази даних забезпечується через Prisma ORM, схема якої представлена в лістингу А.6.

```
@Injectable()
export class DatabaseService
  extends PrismaClient
  implements OnModuleInit, OnModuleDestroy
{
  private static instance: DatabaseService;
  private readonly logger = new Logger(DatabaseService.name);

  constructor() {
    if (DatabaseService.instance) {
      return DatabaseService.instance;
    }

    super({
      log: ['query', 'info', 'warn', 'error'],
      errorFormat: 'pretty',
    });

    DatabaseService.instance = this;
  }

  async onModuleInit() {
    try {
      this.logger.log('Connecting to database...');
      await this.$connect();
      this.logger.log('Successfully connected to database');
    } catch (error) {
      this.logger.error('Failed to connect to database', error);
      throw new DatabaseError('Database connection failed', error as Error);
    }
  }

  async onModuleDestroy() {
    try {
      await this.$disconnect();
      this.logger.log('Database disconnected');
    } catch (error) {
      this.logger.error('Error disconnecting from database', error);
    }
  }
}
```

Рисунок 3.2 – DatabaseService, який забезпечує централізований доступ до бази даних.

Система обробки помилок Prisma, яка зображена на рис. 3.3, перетворює специфічні помилки бази даних у зрозумілі для додатка виключення. Система обробки помилок включає:

- P2002: порушення унікальності полів;
- P2025: запис не знайдено;

- P2003: порушення зовнішнього ключа;
- P2014: некоректний ідентифікатор.

```

private handlePrismaError(
  error: any,
  operation: string,
  model: string,
): never {
  this.logger.error(`Database error in ${operation} for ${model}:`, error);

  if (error instanceof Prisma.PrismaClientKnownRequestError) {
    switch (error.code) {
      case 'P2002':
        throw new ValidationError(
          `Unique constraint violation: ${error.meta?.target}`,
        );
      case 'P2025':
        throw new RecordNotFoundError(model, error.meta);
      case 'P2003':
        throw new ValidationError(
          `Foreign key constraint violation: ${error.meta?.field_name}`,
        );
      case 'P2014':
        throw new ValidationError(`Invalid ID: ${error.meta?.target}`);
      default:
        throw new DatabaseError(
          `Database operation failed: ${error.message}`,
          error,
        );
    }
  }

  if (error instanceof Prisma.PrismaClientUnknownRequestError) {
    throw new DatabaseError('Unknown database error occurred', error);
  }

  if (error instanceof Prisma.PrismaClientRustPanicError) {
    throw new DatabaseError('Database engine error occurred', error);
  }

  if (error instanceof Prisma.PrismaClientValidationError) {
    throw new ValidationError(`Validation error: ${error.message}`);
  }

  // Загальна помилка
  throw new DatabaseError(`Unexpected error in ${operation}`, error);
}

```

Рисунок 3.3 – Обробка помилок бази даних

Транзакції, схему реалізації яких зображено на рис. 3.4, забезпечують атомарність операцій з базою даних, що особливо важливо для бізнес-критичних операцій, таких як створення угод або обробка замовлень.

```

280 // Метод для транзакцій з обробкою помилок
281 async executeTransaction<T>(
282   operations: (tx: Prisma.TransactionClient) => Promise<T>,
283 ): Promise<T> {
284   try {
285     this.logger.debug('Starting database transaction');
286     const result = await this.$transaction(operations);
287     this.logger.debug('Transaction completed successfully');
288     return result;
289   } catch (error) {
290     this.logger.error('Transaction failed:', error);
291     this.handlePrismaError(error, 'transaction', 'multiple');
292   }
293 }
294

```

Рисунок 3.4 – Реалізація методу executeTransaction для виконання транзакцій

Основні модулі серверної частини системи:

- AuthModule – це модуль авторизації та автентифікації (лістинг А.3), який дозволяє безпечно отримати доступ до системи. Реалізація автентифікаційного сервісу представлена в лістингу А.4, а контролер для обробки запитів авторизації показано в лістингу А.5;
- UsersModule – управління користувачами та їх ролями. Підтримує створення облікових записів, редагування профілів, призначення ролей та контроль прав доступу відповідно до бізнес-логіки;
- OrganizationsModule – модуль для управління організаціями та їх структурою. Дозволяє створювати ієрархічну структуру компаній з різними рівнями доступу та налаштуваннями;
- CustomersModule – управління клієнтською базою з підтримкою як індивідуальних клієнтів, так і корпоративних. Включає сегментацію клієнтів та ведення історії взаємодії;
- DealsModule – це модуль, який дозволяє керувати воронкою продажів і вести угоди. Він підтримує різні етапи угод, прогнозування та аналітику ефективності продажів;

- `ProjectsModule` – це модуль, який дозволяє керувати проектами та завданнями, планувати ресурси, контролювати терміни та відстежувати прогрес виконання;
- `OrdersModule` – обробка замовлень та управління електронною комерцією. Включає управління каталогом товарів, обробку платежів та відстеження доставки;
- `AnalyticsModule` – це модуль, який дозволяє збирати та аналізувати дані системи, щоб створювати звіти та бізнес-інсайти. Він відстежує KPI (Key Performance Indicators) та надає дані для управління.

### 3.2 Опис модулів та компонентів клієнтської частини

Клієнтська частина системи розроблена з використанням `React` та `Material-UI` для створення сучасного та зручного користувацького інтерфейсу. Система організована за компонентним принципом з використанням `Redux Toolkit` для управління глобальним станом.

Маршрутизація клієнтського додатку та організація компонентів демонструється в лістингу A.7, а конфігурація точки входу `React` додатку з усіма необхідними провайдерами показана в лістингу A.8.

Для забезпечення типізації та підвищення надійності коду застосовується `TypeScript`. Інструменти `ESLint` та `TypeScript ESLint` інтегровані для статичного аналізу та дотримання кодстайлу. Збірка та запуск проєкту здійснюється за допомогою `Vite`, що забезпечує високу продуктивність під час розробки та швидке збирання у продакшн.

Система розроблена як `PWA` (`Progressive Web Application`) з повною підтримкою `offline` функціональності. `Service Worker` забезпечує кешування критичних ресурсів та `API` відповідей, дозволяючи користувачам продовжувати роботу навіть при відсутності інтернет-з'єднання. При відновленні з'єднання всі зміни автоматично синхронізуються з сервером з розумним вирішенням конфліктів.

Компонентна архітектура включає власну систему дизайну на базі Material-UI з розширеними компонентами для бізнес-логіки. Компоненти як DataGrid підтримують віртуалізацію для роботи з великими таблицями, lazy loading для поступового завантаження даних, та вбудовані можливості експорту в різні формати. Всі компоненти повністю типізовані з TypeScript інтерфейсами.

Система управління станом використовує розширені можливості Redux Toolkit з RTK Query для автоматичного кешування та оптимізації мережевих запитів. Стан додатка організований в slice-и з чіткою відповідальністю кожного за певну доменну область. Middleware включає логування всіх дій в development режимі та автоматичне збереження критичного стану в localStorage.

Responsive дизайн реалізований через систему breakpoints з урахуванням не тільки ширини екрану, але й висоти, орієнтації та типу пристрою. Компоненти автоматично адаптують не тільки розмір та розташування, але й спосіб взаємодії - наприклад, hover ефекти відключаються на touch пристроях, а контекстні меню замінюються на bottom sheets.

Головна панель включає персоналізовані віджети для різних ролей користувачів. Адміністратори бачать загальну статистику системи, менеджери з продажів - показники угод, а проект-менеджери - статус поточних проектів. На рис. 3.5 зображено головну панель управління системи, яка адаптується відповідно до ролі користувача.

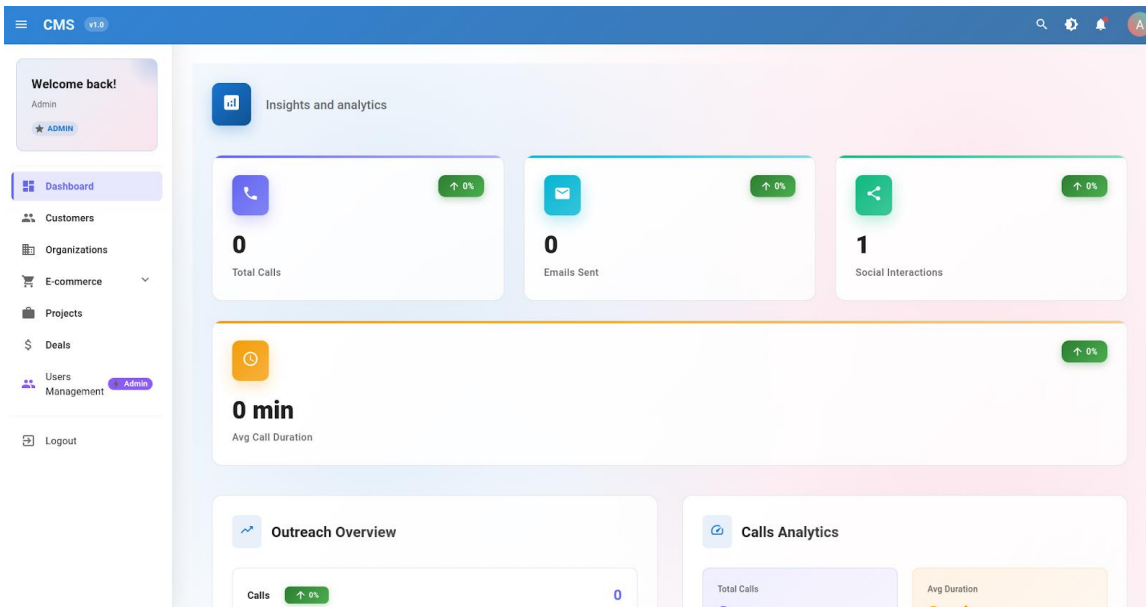


Рисунок 3.5 – Головна панель управління системи

Модуль управління клієнтами, який зображен на рис. 3.6, включає:

- таблицю з пагінацією та сортуванням;
- фільтри за статусом та типом клієнта;
- форми для створення та редагування клієнтських карток;
- історію взаємодії з клієнтами.

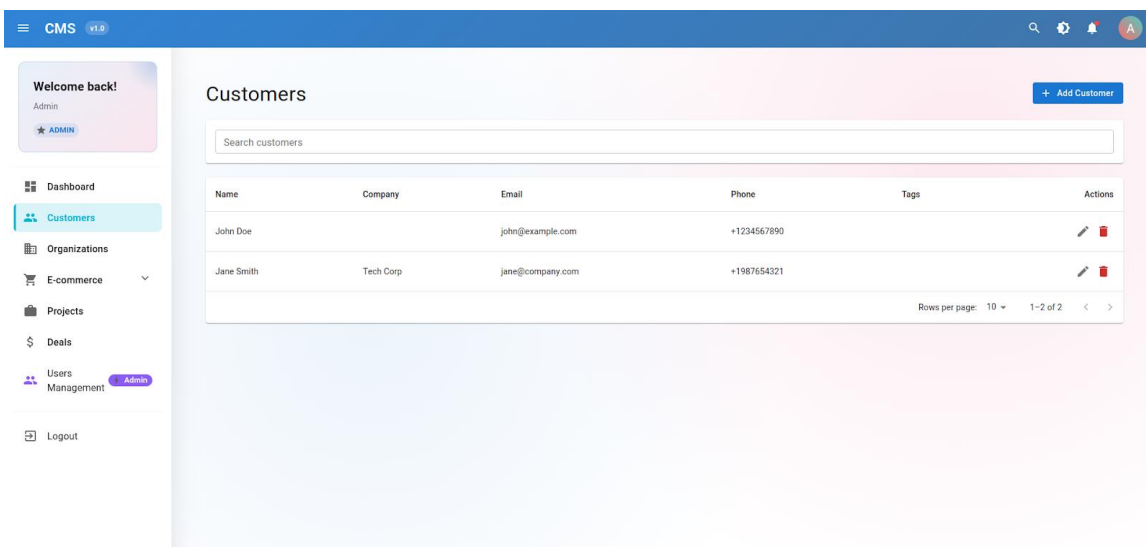


Рисунок 3.6 – Інтерфейс управління клієнтами

Воронка продажів, яка зображена на рис. 3.7, візуалізує етапи угод та дозволяє:

- переміщувати угоди між етапами;
- відстежувати ймовірність закриття;
- прогнозувати доходи;
- аналізувати ефективність продажів.

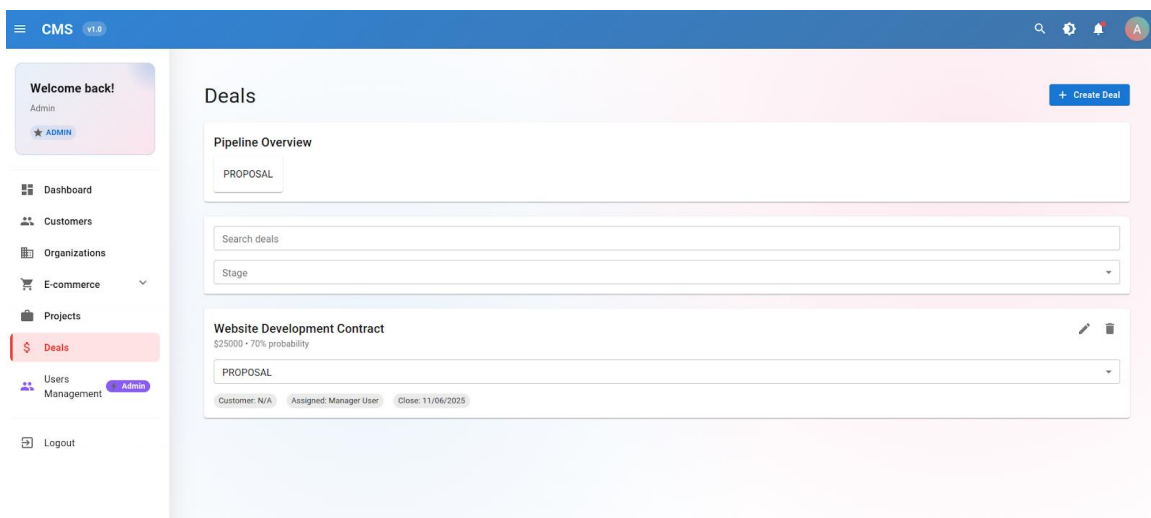


Рисунок. 3.7 – Інтерфейс воронки продажів для відстеження угод

Система управління проектами, яка зображена на рис. 3.8, включає:

- Kanban дошку для візуального управління завданнями;
- календар з термінами виконання;
- розподіл ресурсів та навантаження команди;
- відстеження прогресу та бюджету проектів.

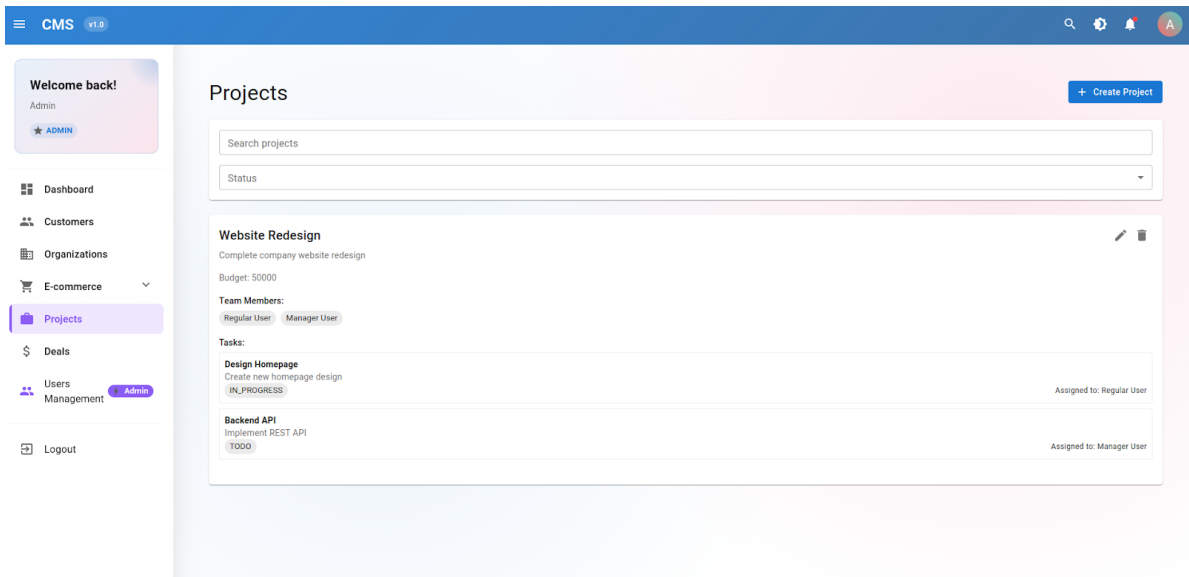


Рисунок 3.8 – Модуль управління проектами з Kanban дошкою

Система електронної комерції, яка зображена на рис. 3.9, включає:

- каталог товарів з можливістю редагування;
- обробку замовлень та відстеження статусів;
- управління складськими залишками.

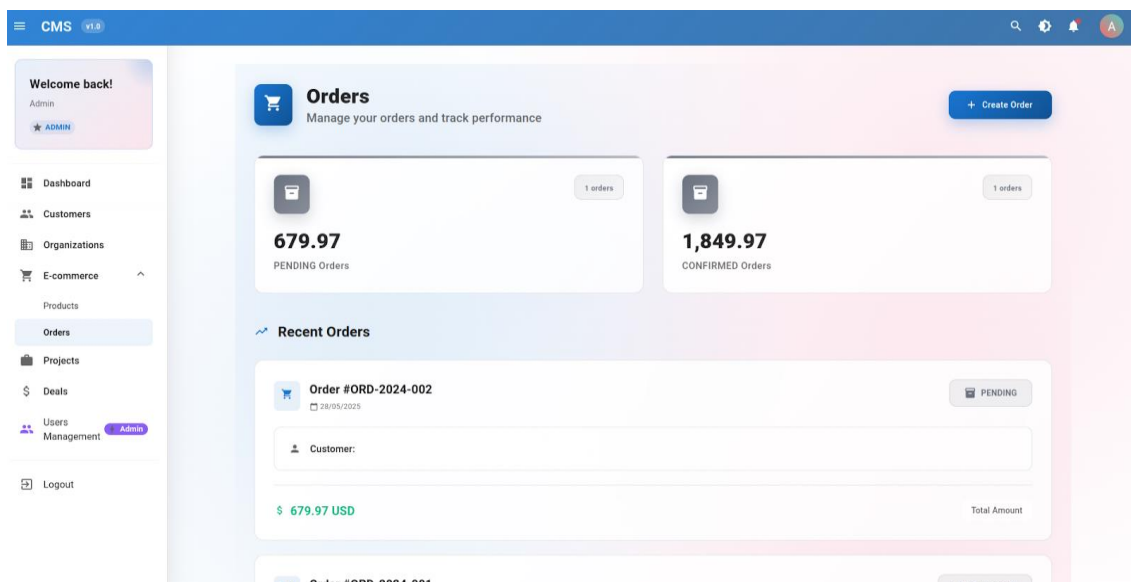


Рисунок 3.9 – Модуль електронної комерції

Основні компоненти клієнтської частини:

- Header Component – це верхня навігація, яка містить меню користувача, сповіщення та швидкий пошук по системі;
- Sidebar Navigation – це бічне меню, яке дозволяє отримати доступ до всіх модулів системи. Воно адаптується залежно від ролі користувача;
- Dashboard Widgets – це настроювані віджети для головної панелі, які показують важливі метрики та дозволяють швидко отримати доступ до важливих функцій;
- Data Tables – це універсальні компоненти таблиць, які підтримують пагінацію, сортування, фільтрацію та експорт даних;
- Form Components – це набір форм, які дозволяють створювати та змінювати записи з валідацією та обробкою помилок;
- Modal Windows – модальні вікна для швидких дій без переходу на інші сторінки;
- Chart Components – це компоненти, які можна використовувати для представлення даних у вигляді графіків, діаграм та інших інтерактивних елементів.

### 3.3 Інструкція до розробленої системи управління контентом

Для того, щоб автоматизована система управління контентом ефективно працювала, користувач повинен виконати кілька основних кроків. Спочатку необхідно отримати доступ до системи від адміністратора, який створить обліковий запис і призначить відповідну роль. Технічна реалізація системи автентифікації та авторизації детально представлена в лістингах А.3-А.5, що включають налаштування модуля, бізнес-логіку сервісу та REST API контролер.

Початкові кроки роботи з системою:

- вхід до системи: користувач вводить свої облікові дані (email та пароль) у формі авторизації. Система перевіряє дані та надає доступ відповідно до призначеної ролі;
- ознайомлення з інтерфейсом: після входу користувач потрапляє на персоналізовану головну панель (Dashboard), яка надає швидкий доступ до основних функцій;
- навігація по системі: використовуючи бічне меню, користувач може переходити між різними модулями системи. Доступні модулі залежать від ролі користувача.

Робота з основними модулями.

Управління клієнтами:

- створення нових клієнтських карток через форму "Додати клієнта";
- редагування існуючих записів через контекстне меню;
- використання фільтрів для пошуку потрібних клієнтів;
- перегляд історії взаємодії з кожним клієнтом.

Ведення угод:

- створення нових угод з прив'язкою до клієнтів;
- переміщення угод між етапами воронки продажів;
- встановлення ймовірності закриття та очікуваних сум;
- відстеження прогресу та прогнозування доходів.

Управління проектами:

- створення проектів з визначенням цілей та термінів;
- розподіл завдань між членами команди;
- відстеження прогресу через Kanban дошку;
- контроль бюджету та ресурсів проекту.

Електронна комерція:

- додавання товарів до каталогу з описами та цінами;
- обробка вхідних замовлень та зміна їх статусів;
- контроль складських залишків;

- аналіз продажів по товарних категоріях.

Аналітика та звітність:

- перегляд готових дашбордів з ключовими метриками;
- створення власних звітів за обраними параметрами;
- експорт даних у форматах Excel та PDF;
- налаштування автоматичних сповіщень по KPI.

Додаткові можливості системи:

- персоналізація: користувачі можуть налаштувати власні дашборди, додаючи або видаляючи віджети відповідно до своїх потреб;
- пошук та фільтрація: система надає потужні можливості пошуку по всіх модулях з використанням різних критеріїв та фільтрів;
- сповіщення: автоматичні сповіщення про важливі події, терміни та зміни статусів допомагають не пропустити критичні моменти;
- співпраця: можливість залишати коментарі, призначати завдання та спільно працювати над проектами;
- мобільна версія: адаптивний дизайн дозволяє користуватися системою на мобільних пристроях без втрати функціональності;
- безпека: система автоматично завершує сеанс після періоду неактивності та вимагає підтвердження для критичних операцій.

Для завершення роботи з системою користувач повинен натиснути кнопку "Вихід" у меню користувача. Це забезпечить безпечне завершення сеансу та зберігатиме всі зміни, які були внесені.

Система аналітики використовує машинне навчання для виявлення patterns в поведінці користувачів та прогнозування бізнес-показників. Алгоритми автоматично аналізують історичні дані для виявлення найбільш ефективних каналів залучення клієнтів, оптимальних часів для контакту, та прогнозування ймовірності закриття угод. Результати представляються у вигляді інтуїтивних візуалізацій з рекомендаціями для покращення показників.

Інтеграційна платформа підтримує широкий спектр зовнішніх сервісів через стандартизовані API та webhook механізми. Система може автоматично синхронізуватися з платіжними системами Stripe та PayPal, імпортувати контакти з Google Workspace та Microsoft 365, відправляти повідомлення через Slack та Microsoft Teams. Всі інтеграції налаштовуються через графічний інтерфейс без необхідності програмування.

Система звітності включає конструктор звітів з drag-and-drop інтерфейсом, що дозволяє користувачам створювати власні дашборди та звіти без технічних знань. Звіти можуть включати дані з декількох модулів з автоматичними розрахунками метрик, тенденцій та прогнозів. Система підтримує планування автоматичного відправлення звітів електронною поштою з налаштованою періодичністю.

Мобільні додатки розроблені з використанням React Native для забезпечення native experience на iOS та Android платформах. Додатки включають push сповіщення про важливі події, біометричну автентифікацію для швидкого входу, та offline синхронізацію для роботи без інтернету. Специфічні функції мобільних пристроїв як GPS (Global Positioning System) використовуються для геолокації клієнтів та автоматичного відстеження візитів.

Система безпеки включає розширені механізми захисту як двофакторна автентифікація через SMS або TOTP (Time-based One-Time Password) додатки, шифрування sensitive даних на рівні application, та регулярні penetration testing для виявлення вразливостей. Всі дані передаються через HTTPS з додатковим шифруванням critical інформації. Система автоматично виявляє та блокує підозрілу активність з детальним логуванням для security audit.

Система підтримує розширені можливості імпорту та експорту даних для полегшення міграції з інших систем та резервного копіювання. Користувачі можуть імпортувати дані з Excel, CSV файлів або через API з інших CRM систем. Експорт підтримує різні формати включаючи PDF звіти з

брендингом компанії, Excel таблиці для подальшого аналізу та JSON для технічної інтеграції.

Автоматизація робочих процесів дозволяє створювати складні сценарії без програмування. Користувачі можуть налаштувати автоматичне призначення задач при створенні угод, відправку email нагадувань за день до дедлайну проекту, автоматичне переведення клієнтів в інший статус після певних дій. Система підтримує умовну логіку, затримки та інтеграцію з зовнішніми сервісами.

Мобільні додатки (iOS та Android) забезпечують повний функціонал системи з оптимізованим інтерфейсом для мобільних пристроїв. Push сповіщення інформують про нові задачі, терміни виконання та важливі зміни в проектах. Офлайн режим дозволяє переглядати та редагувати дані без інтернету з автоматичною синхронізацією при підключенні.

Система аналітики включає машинне навчання для прогнозування поведінки клієнтів, оптимізації цін та виявлення трендів. Штучний інтелект аналізує історичні дані для надання рекомендацій щодо найкращого часу для контакту з клієнтами, ймовірності закриття угод та оптимальних маркетингових каналів.

### Висновок до розділу 3

У цьому розділі було детально розглянуто реалізацію автоматизованої системи управління контентом для малого бізнесу. Описано архітектуру серверної частини, яка побудована на основі NestJS фреймворку з використанням модульного підходу. Кожен модуль системи відповідає за окрему бізнес-область та забезпечує слабку зв'язаність компонентів.

Реалізація системи демонструється через ключові компоненти коду: конфігурацію NestJS додатку (лістинг А.1), організацію модульної архітектури (лістинг А.2), систему автентифікації (лістинги А.3-А.5), структуру бази даних (лістинг А.6) та клієнтську частину (лістинги А.7-А.8).

Особливу увагу приділено реалізації DatabaseService з використанням Prisma ORM, що забезпечує типобезпечну роботу з базою даних PostgreSQL. Система включає надійну обробку помилок та підтримку транзакцій для забезпечення цілісності бізнес-критичних операцій.

Клієнтська частина системи реалізована з використанням React та Material-UI, що забезпечує сучасний та зручний користувацький інтерфейс. Модульна структура frontend частини дозволяє створювати персоналізовані інтерфейси для різних ролей користувачів.

Розроблено детальну інструкцію користувача, яка охоплює всі основні функції системи від базової навігації до роботи зі складними аналітичними звітами. Система забезпечує комплексне рішення для управління всіма аспектами діяльності малого бізнесу.

У результаті створено повнофункціональну систему управління контентом, яка відповідає сучасним вимогам до безпеки, масштабованості та зручності використання. Модульна архітектура дозволяє легко розширювати функціональність системи відповідно до зростаючих потреб бізнесу.

## 4 ПОДАЛЬШИЙ РОЗВИТОК ТА ВДОСКОНАЛЕННЯ СИСТЕМИ

### 4.1 Концепція еволюційного розвитку автоматизованої системи

Сучасний динамічний бізнес-ландшафт вимагає від корпоративних систем управління не лише відповідності поточним потребам організацій, але й здатності до постійної еволюції та адаптації до змінних вимог ринку. Автоматизована система управління контентом для малого бізнесу, розроблена в межах даного дослідження, закладає потужний фундамент для подальшого розширення функціональності та впровадження інноваційних технологічних рішень.

Стратегія подальшого розвитку системи базується на принципах масштабованості, модульності та інтеграційної сумісності. Поточна архітектура системи, побудована на основі NestJS backend та React frontend з використанням TypeScript, створює необхідні передумови для безперервного вдосконалення без порушення існуючої функціональності. Модульна структура дозволяє поетапне впровадження нових можливостей з мінімальним впливом на роботу системи в цілому.

Аналіз існуючої реалізації демонструє, що система вже включає базові компоненти для аналітики даних, управління організаціями, роботи з клієнтською базою та електронної комерції. Проте потенціал подальшого розвитку значно перевищує поточні можливості. Впровадження технологій штучного інтелекту, розширення інтеграційних можливостей та створення екосистеми додаткових сервісів відкриває нові горизонти для трансформації системи у комплексну платформу цифровізації бізнесу.

Важливим аспектом еволюційного розвитку є збереження зворотної сумісності та забезпечення плавного переходу існуючих користувачів на нові версії системи. Це вимагає ретельного планування архітектурних змін, впровадження механізмів версіонування API та створення комплексної системи міграції даних. Кожне нововведення має ґрунтуватися на глибокому

аналізі потреб користувачів та тенденцій розвитку індустрії.

Концепція подальшого розвитку також передбачає створення відкритої екосистеми, яка дозволить стороннім розробникам створювати додаткові модулі та розширення. Це забезпечить швидке реагування на специфічні потреби різних галузей бізнесу та сприятиме формуванню активної спільноти навколо платформи.

#### 4.2 Інтеграція технологій машинного навчання та штучного інтелекту

Впровадження технологій машинного навчання у систему управління контентом відкриває нові можливості для автоматизації процесів прийняття рішень та надання інтелектуальних рекомендацій користувачам. Поточна реалізація системи аналітики, яка базується на модулі AnalyticsService, створює необхідну інфраструктуру для збору та обробки даних, що є фундаментом для впровадження алгоритмів машинного навчання.

Система прогнозування продажів стане одним з ключових компонентів інтелектуального модуля. Використовуючи історичні дані угод, сезонні коливання та поведінкові паттерни клієнтів, алгоритми зможуть генерувати точні прогнози майбутніх продажів з урахуванням ймовірності закриття угод. Модель прогнозування інтегрується з існуючим DealsModule та використовуватиме дані з таблиць deals, customers та analytics для навчання та валідації.

Рекомендаційна система для клієнтів базуватиметься на аналізі історії покупок, взаємодії з контентом та демографічних характеристиках. Система автоматично ідентифікуватиме схожих клієнтів та пропонуватиме персоналізовані рекомендації товарів або послуг. Це особливо актуально для модуля електронної комерції, де персоналізація безпосередньо впливає на конверсію та середній чек.

Інтелектуальна система класифікації клієнтів автоматично сегментуватиме клієнтську базу на основі поведінкових паттернів, цінності

для бізнесу та потенціалу росту. Алгоритми кластеризації аналізуватимуть дані з CustomerModule та створюватимуть динамічні сегменти, які автоматично оновлюються при надходженні нових даних.

Система виявлення аномалій моніторитиме бізнес-процеси та автоматично виявлятиме відхилення від нормальних паттернів роботи. Це включає виявлення підозрілих транзакцій, аномальної поведінки користувачів та потенційних проблем в операційних процесах. Модуль інтегрується з існуючою системою аналітики та використовує дані з усіх модулів системи.

Чатбот з природним мовним інтерфейсом забезпечить автоматизовану підтримку клієнтів та допомогу співробітникам у роботі з системою. Використовуючи технології обробки природної мови NLP (Natural Language Processing), чатбот розумітиме контекст запитів та надаватиме релевантні відповіді на основі документації системи та історії взаємодії.

Система автоматичного генерування звітів використовуватиме алгоритми для створення інсайтів та рекомендацій на основі аналізу бізнес-даних. Замість статичних звітів, система генеруватиме динамічні аналітичні огляди з виявленими трендами, аномаліями та рекомендаціями для покращення показників.

Модуль оптимізації цін застосовуватиме алгоритми динамічного ціноутворення на основі попиту, конкурентного середовища та характеристик клієнтів. Система автоматично корегуватиме ціни для максимізації прибутку або досягнення інших бізнес-цілей.

Технічна реалізація модулів машинного навчання передбачає створення окремого MLModule, який інтегрується з існуючою архітектурою через REST API та асинхронні черги завдань. Для обробки великих обсягів даних впроваджується система потокової обробки на базі Apache Kafka або Redis Streams. Навчені моделі зберігаються у спеціалізованих сховищах як MLflow або власному model registry з підтримкою версіонування та A/B тестування.

### 4.3 Розширення інтеграційних можливостей та API екосистеми

Сучасні бізнес-процеси характеризуються використанням множини спеціалізованих інструментів та сервісів, що робить інтеграційні можливості критично важливими для успішного впровадження будь-якої корпоративної системи. Розширення інтеграційної платформи автоматизованої системи управління контентом передбачає створення комплексної екосистеми взаємодії з зовнішніми сервісами та формування власного API marketplace.

Фінансова інтеграція стане одним з пріоритетних напрямків розвитку. Система інтегрується з провідними банківськими API для автоматичного імпорту транзакцій, верифікації платежів та управління корпоративними рахунками. Інтеграція з такими системами як Plaid, Yodlee або відкритими банківськими API дозволить автоматично синхронізувати фінансові дані та забезпечить реальний час відстеження грошових потоків. Додатково, система підтримуватиме інтеграцію з криптовалютними платіжними шлюзами для організацій, що працюють з цифровими активами.

Інтеграція з провідними CRM платформами забезпечить можливість міграції даних та синхронізації інформації про клієнтів. Система підтримуватиме двосторонню синхронізацію з Salesforce, HubSpot, Pipedrive та іншими популярними CRM рішеннями. Це особливо важливо для організацій, які поступово переходять на нову платформу або використовують гібридний підхід до управління клієнтськими відносинами.

Комунікаційна інтеграція охоплюватиме широкий спектр каналів взаємодії. Система інтегрується з електронною поштою через Gmail API, Outlook API та SMTP (Simple Mail Transfer Protocol) протоколи для автоматичного ведення листування. Інтеграція з месенджерами як Slack, Microsoft Teams, Discord дозволить отримувати сповіщення та виконувати базові операції прямо з робочих каналів комунікації. Додатково, впровадження SMS та голосових повідомлень через Twilio, Nexmo або

локальних провайдерів розширить можливості багатоканального маркетингу. Маркетингова автоматизація передбачає інтеграцію з провідними платформами email маркетингу як Mailchimp, Constant Contact, SendGrid. Система автоматично синхронізуватиме списки контактів, результати кампаній та поведінкові дані для створення єдиного профілю клієнта. Інтеграція з соціальними мережами через Facebook API, Instagram API, LinkedIn API забезпечить автоматичну публікацію контенту та аналіз ефективності соціальних медіа кампаній.

E-commerce інтеграція включатиме підключення до головних маркетплейсів як Amazon, eBay, Etsy для автоматичної синхронізації каталогу товарів та управління замовленнями. Система підтримуватиме інтеграцію з платіжними системами Stripe, PayPal, Square та локальними платіжними провайдерами для забезпечення максимальної гнучкості у виборі методів оплати. Інтеграція з логістичними компаніями UPS (United Parcel Service), FedEx, DHL дозволить автоматично розраховувати вартість доставки та відслідковувати статус відправлень.

Бухгалтерська інтеграція забезпечить синхронізацію з провідними системами обліку як QuickBooks, Xero, FreshBooks. Система автоматично експортуватиме рахунки, імпортуватиме платежі та синхронізуватиме податкову звітність. Інтеграція з податковими сервісами TaxJar, Avalara забезпечить автоматичний розрахунок податків для різних юрисдикцій.

Технічна реалізація інтеграційної платформи базується на створенні універсального Integration Gateway, який централізовано управляє всіма зовнішніми підключеннями. Система використовує архітектурний паттерн ESB (Enterprise Service Bus) для забезпечення надійної та масштабованої взаємодії з зовнішніми сервісами. Кожна інтеграція реалізується як окремий мікросервіс з власними механізмами аутентифікації, ретраїв та обробки помилок.

Webhook система забезпечить реальний час синхронізацію даних між системами. Створення GraphQL API поряд з існуючим REST API дозволить

клієнтам ефективно запитувати лише необхідні дані. Система підтримуватиме OpenAPI стандарт для автоматичної генерації документації та SDK (Software Development Kit) для різних мов програмування.

#### 4.4 Мобільна платформа та кросплатформенні рішення

Розвиток мобільних технологій та зміна робочих звичок у бік віддаленої роботи роблять мобільну доступність корпоративних систем не просто бажаною функцією, а необхідністю для конкурентоспроможності. Створення комплексної мобільної платформи для автоматизованої системи управління контентом передбачає розробку нативних додатків, прогресивних веб-додатків (PWA) та забезпечення безшовної синхронізації між різними пристроями.

Архітектура мобільної платформи базується на принципах responsive-first design з адаптацією під різні форм-фактори пристроїв. Система підтримуватиме роботу на смартфонах, планшетах, розкладних пристроях та навіть розумних годинниках. Кожна категорія пристроїв матиме оптимізований інтерфейс, що враховує особливості взаємодії та технічні обмеження.

Нативні мобільні додатки розроблятимуться з використанням React Native для забезпечення максимальної сумісності коду з веб-платформою. Це дозволить підтримувати єдину кодову базу для більшості компонентів бізнес-логіки, одночасно забезпечуючи нативну продуктивність та доступ до специфічних функцій пристроїв. iOS додаток використовуватиме можливості Swift для критичних компонентів продуктивності, а Android версія інтегруватиме Kotlin код для оптимізації роботи з системними API.

Офлайн-функціональність стане ключовою особливістю мобільної платформи. Система реалізуватиме інтелектуальне кешування даних з пріоритизацією найбільш важливої інформації для кожного користувача. Механізм offline-first дозволить користувачам продовжувати роботу навіть

при відсутності інтернет-з'єднання, з автоматичною синхронізацією при відновленні зв'язку. Конфліктне розв'язання забезпечить коректну обробку ситуацій, коли один і той же об'єкт модифікується різними користувачами в офлайн режимі.

Геолокаційні можливості інтегрують GPS функціональність для автоматичного відстеження місцезнаходження співробітників під час візитів до клієнтів, логування робочого часу на різних об'єктах та оптимізації маршрутів для мобільних робітників. Система підтримуватиме геофенсинг для автоматичного відмічання приходу/відходу з робочих локацій та сповіщення про важливі події в певних географічних зонах.

Push-сповіщення забезпечать своєчасне інформування користувачів про критичні події бізнесу. Система підтримуватиме персоналізовані налаштування сповіщень для різних типів подій, часових зон та робочих графіків. Інтелектуальне планування сповіщень враховуватиме активність користувача та важливість повідомлення для мінімізації порушень робочого процесу.

Біометрична автентифікація використовуватиме Touch ID, Face ID, відбитки пальців та розпізнавання голосу для забезпечення безпечного та зручного доступу до системи. Система підтримуватиме багатофакторну автентифікацію з поєднанням біометричних та традиційних методів верифікації.

Доповнена реальність AR (Augmented Reality) інтегрується для візуалізації бізнес-даних в контексті реального світу. Складські працівники зможуть сканувати QR-коди товарів для миттєвого отримання інформації про залишки та характеристики. Менеджери з продажів матимуть можливість візуалізувати інформацію про клієнтів при наближенні до їх офісів або магазинів.

Голосовий інтерфейс забезпечить hands-free взаємодію з системою через інтеграцію з Siri, Google Assistant та власним голосовим помічником. Користувачі зможуть створювати завдання, отримувати звіти та виконувати

прості операції за допомогою голосових команд.

Адаптивне кешування даних оптимізуватиме використання мобільного трафіку та сховища пристрою. Система автоматично аналізуватиме паттерни використання та передзавантажуватиме найбільш релевантні дані в оптимальний час. Компресія даних та інкрементальна синхронізація мінімізуватимуть споживання трафіку.

Інтеграція з мобільними платіжними системами як Apple Pay, Google Pay, Samsung Pay дозволить здійснювати швидкі B2B платежі та експенс-менеджмент прямо з мобільного додатку. NFC (Near Field Communication) технологія забезпечить безконтактні платежі та обмін контактною інформацією.

#### 4.5 Хмарна інфраструктура та технології розподіленого розгортання

Еволюція хмарних технологій створює нові можливості для масштабування корпоративних систем та забезпечення їх глобальної доступності. Міграція автоматизованої системи управління контентом у хмарне середовище передбачає не просто зміну місця розташування серверів, а фундаментальну трансформацію архітектури для максимального використання переваг cloud-native підходу.

Мульти-хмарна стратегія розгортання забезпечить високу доступність та мінімізацію ризиків vendor lock-in. Система підтримуватиме розгортання на AWS, Google Cloud Platform, Microsoft Azure з можливістю гібридного використання ресурсів різних провайдерів. Кожен хмарний провайдер пропонує унікальні сервіси, які оптимізовані для конкретних завдань: AWS (Amazon Web Services) для машинного навчання та аналітики, GCP для big data обробки, Azure для інтеграції з Microsoft екосистемою.

Kubernetes оркестрація стане основою для управління контейнеризованими додатками. Система використовуватиме Helm charts для стандартизованого розгортання в різних середовищах. Автоматичне

масштабування на базі HPA (Horizontal Pod Autoscaler) та VPA (Vertical Pod Autoscaler) забезпечить оптимальне використання ресурсів відповідно до навантаження. Istio service mesh додасть додатковий рівень безпеки та спостережуваності для мікросервісної архітектури.

Serverless обчислення інтегруються для обробки спорадичних завдань та піків навантаження. AWS Lambda, Google Cloud Functions та Azure Functions використовуватимуться для фонових завдань як генерація звітів, обробка зображень, відправка email сповіщень. Це дозволить оптимізувати витрати на інфраструктуру, оплачуючи лише фактично використані ресурси.

Глобальна мережа доставки контенту (CDN) забезпечить швидкий доступ до системи з будь-якої точки світу. CloudFlare, AWS CloudFront або Google Cloud CDN кешуватимуть статичні ресурси та API відповіді у географічно розподілених точках присутності. Інтелектуальна маршрутизація направлятиме користувачів до найближчого дата-центру для мінімізації латентності.

Edge computing розширить можливості системи для роботи з IoT (Internet of Things) пристроями та обробки даних на периферії мережі. AWS IoT Greengrass, Azure IoT Edge дозволять виконувати частину бізнес-логіки безпосередньо на локальних пристроях, зменшуючи залежність від центральної інфраструктури та покращуючи час відгуку для критичних операцій.

Розподілена база даних реалізується через database sharding та master-slave реплікацію для забезпечення високої доступності та продуктивності. PostgreSQL cluster з автоматичним failover гарантуватиме безперервну роботу навіть при відмові окремих вузлів. Для аналітичних завдань використовуватиметься окремий data warehouse на базі Amazon Redshift, Google BigQuery або Azure Synapse Analytics.

Containerization стратегія охопить всі компоненти системи. Docker контейнери забезпечать консистентність середовища розгортання від розробки до продакшену. Multi-stage builds оптимізуватимуть розмір образів,

а distroless базові образи підвищать безпеку. Container registry з автоматичним сканування вразливостей гарантуватиме якість образів.

IaC (Infrastructure as Code) автоматизує управління інфраструктурою через Terraform, AWS CloudFormation або ARM templates. Версіонування інфраструктурного коду дозволить відстежувати зміни та швидко відкочуватися до попередніх станів. GitOps підхід інтегрує управління інфраструктурою з процесами розробки через pull request та code review.

Disaster Recovery план включатиме автоматичне резервне копіювання в географічно розподілені локації, реплікацію даних у реальному часі та можливість швидкого відновлення роботи у резервному дата-центрі. RTO (Recovery Time Objective) цільовий показник складатиме менше 15 хвилин для критичних компонентів системи.

Моніторинг та observability реалізовуватимуться через Prometheus для збору метрик, Grafana для візуалізації, Jaeger для трасування запитів та ELK stack (Elasticsearch, Logstash, Kibana) для централізованого логування. Machine learning алгоритми аналізуватимуть метрики для прогнозування проблем та автоматичної оптимізації продуктивності.

## ВИСНОВКИ

Впродовж роботи над кваліфікаційною роботою було розроблено автоматизовану систему управління контентом для малого бізнесу.

Розробка сучасних корпоративних систем управління вимагає глибокого аналізу та використання найкращих доступних технологій. Зокрема, для створення автоматизованої системи управління контентом для малого бізнесу було обрано ряд сучасних технологій і бібліотек. Кожна з цих технологій забезпечує високий рівень функціональності, масштабованості та зручності використання, що є ключовими факторами для успішної розробки корпоративних рішень.

Було розглянуто процес проектування та розробки комплексної системи управління бізнес-процесами. Перш за все, було визначено специфікацію функціональних та нефункціональних вимог до системи, що слугувала основою для подальшої розробки. Далі було створено UML діаграми варіантів використання та структури бази даних, які допомогли візуалізувати взаємодію акторів системи та архітектуру даних.

Після цього був проведений детальний аналіз і вибір технологій для розробки як серверної частини з використанням NestJS фреймворку, так і клієнтського інтерфейсу на базі React та Material-UI. Особливу увагу було приділено вибору PostgreSQL як основної бази даних та Prisma ORM для типобезпечної роботи з даними.

Було детально описано розробку модульної архітектури системи, включаючи ключові компоненти серверної частини та принципи організації бізнес-логіки. Реалізовано восьмимодульну структуру, що охоплює управління користувачами, організаціями, клієнтами, угодами, проектами, замовленнями, товарами та аналітикою. Останнім етапом стало проектування нормалізованої структури бази даних з оптимізованими індексами для забезпечення високої продуктивності системи.

Було розглянуто модулі та компоненти як серверної частини на базі NestJS, так і клієнтської частини веб-додатку на React. Описано їх функціональні можливості та взаємодію між собою для забезпечення належної роботи комплексної системи управління. Особливу увагу приділено реалізації системи ролей та прав доступу, що дозволяє персоналізувати інтерфейс відповідно до посадових обов'язків користувачів.

Також було підготовано детальну інструкцію до розробленої системи управління контентом, яка дозволить користувачам різних ролей ефективно використовувати всю функціональність системи. Ця інструкція охоплює кроки автентифікації, навігації по модулях системи, роботи з клієнтською базою, веденням угод, управлінням проектами, обробкою замовлень та аналізом бізнес-метрик.

У результаті було створено повнофункціональну автоматизовану систему управління контентом, яка забезпечує комплексне рішення для координації всіх аспектів діяльності малого бізнесу. Модульна архітектура системи гарантує можливість подальшого розширення функціональності та адаптації до специфічних потреб різних типів організацій, що робить систему готовою до ефективного використання в реальних бізнес-процесах.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. NestJS - A progressive Node.js framework. NestJS. URL: <https://nestjs.com/> (дата звернення: 15.12.2024).
2. React – A JavaScript library for building user interfaces. React. URL: <https://react.dev/> (дата звернення: 15.12.2024).
3. TypeScript: JavaScript With Syntax For Types. TypeScript. URL: <https://www.typescriptlang.org/> (дата звернення: 15.12.2024).
4. PostgreSQL: The World's Most Advanced Open Source Relational Database. PostgreSQL. URL: <https://www.postgresql.org/> (дата звернення: 16.12.2024).
5. Prisma | Next-generation ORM for Node.js & TypeScript. Prisma. URL: <https://www.prisma.io/> (дата звернення: 16.12.2024).
6. Redux Toolkit | Redux Toolkit. Redux Toolkit. URL: <https://redux-toolkit.js.org/> (дата звернення: 16.12.2024).
7. MUI: The React component library you always wanted. Material-UI. URL: <https://mui.com/> (дата звернення: 17.12.2024).
8. Vite | Next Generation Frontend Tooling. Vite. URL: <https://vitejs.dev/> (дата звернення: 17.12.2024).
9. HubSpot | Inbound Marketing, Sales, and Service Software. HubSpot. URL: <https://www.hubspot.com/> (дата звернення: 18.12.2024).
10. Pipedrive: Sales CRM & Pipeline Management Software. Pipedrive. URL: <https://www.pipedrive.com/> (дата звернення: 18.12.2024).
11. Zoho CRM - Customer Relationship Management Software. Zoho CRM. URL: <https://www.zoho.com/crm/> (дата звернення: 18.12.2024).
12. Domain-Driven Design: Tackling Complexity in the Heart of Software / E. Evans. Addison-Wesley Professional, 2003. 560 с.
13. Building Microservices: Designing Fine-Grained Systems / S. Newman. O'Reilly Media, 2021. 616 с.
14. Clean Architecture: A Craftsman's Guide to Software Structure and

Design / R. Martin. Prentice Hall, 2017. 432 с.

15. Docker: Accelerated Container Application Development. Docker. URL: <https://www.docker.com/> (дата звернення: 19.12.2024).

16. Kubernetes. Kubernetes. URL: <https://kubernetes.io/> (дата звернення: 19.12.2024).

17. Amazon Web Services (AWS) - Cloud Computing Services. AWS. URL: <https://aws.amazon.com/> (дата звернення: 20.12.2024).

18. Google Cloud Platform. Google Cloud. URL: <https://cloud.google.com/> (дата звернення: 20.12.2024).

19. Microsoft Azure Cloud Computing Services. Microsoft Azure. URL: <https://azure.microsoft.com/> (дата звернення: 20.12.2024).

20. JWT.IO - JSON Web Tokens Introduction. JWT.IO. URL: <https://jwt.io/> (дата звернення: 21.12.2024).

21. REST API Tutorial. REST API Tutorial. URL: <https://restfulapi.net/> (дата звернення: 21.12.2024).

22. Progressive Web Apps | MDN. MDN Web Docs. URL: [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps) (дата звернення: 22.12.2024).

23. React Native · Learn once, write anywhere. React Native. URL: <https://reactnative.dev/> (дата звернення: 22.12.2024).

24. Stripe | Payment Processing Platform for the Internet. Stripe. URL: <https://stripe.com/> (дата звернення: 23.12.2024).

25. PayPal Developer. PayPal. URL: <https://developer.paypal.com/> (дата звернення: 23.12.2024).

26. Terraform by HashiCorp. Terraform. URL: <https://www.terraform.io/> (дата звернення: 24.12.2024).

27. Prometheus - Monitoring system & time series database. Prometheus. URL: <https://prometheus.io/> (дата звернення: 24.12.2024).

28. Grafana: The open observability platform. Grafana. URL: <https://grafana.com/> (дата звернення: 24.12.2024).

29. Elasticsearch: The Official Distributed Search & Analytics Engine. Elastic. URL: <https://www.elastic.co/elasticsearch/> (дата звернення: 25.12.2024).
30. General Data Protection Regulation (GDPR). European Commission. URL: <https://gdpr.eu/> (дата звернення: 25.12.2024).
31. Machine Learning Yearning / A. Ng. 2018. URL: <https://www.mlyearning.org/> (дата звернення: 26.12.2024).
32. Designing Data-Intensive Applications / М. Kleppmann. O'Reilly Media, 2017. 616 с.
33. The Pragmatic Programmer: Your Journey to Mastery / D. Thomas, A. Hunt. Addison-Wesley Professional, 2019. 352 с.
34. Microservices Patterns: With examples in Java / C. Richardson. Manning Publications, 2018. 520 с.
35. Site Reliability Engineering: How Google Runs Production Systems / N. Murphy та ін. O'Reilly Media, 2016. 552 с.