

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження моделей управління проектом створення та впровадження програмного забезпечення для аудиту та аналізу маркетингового контенту»

Студентки 2-го курсу групи УПз-21

Аліни ТУРЧЕНКО
(прізвище, ім'я)

Науковий керівник:

кандидат технічних наук, професор
(науковий ступінь, вчене звання)
Віктор МОРОЗОВ
(прізвище, ім'я)

(підпис студента)

(дата)

(підпис)

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач

кафедри технологій

управління

(підпис)

Віктор МОРОЗОВ
(прізвище, ім'я)

(дата)

Київ — 2024

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій**

Кафедра технологій управління
Освітній рівень Магістр
Спеціальність 122 Комп'ютерні науки
Освітня програма Управління проектами

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Віктор МОРОЗОВ

« » 2024 року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студентка Аліна ТУРЧЕНКО

Група УПз-21

1. Тема кваліфікаційної роботи:

«Дослідження моделей управління проектом створення та впровадження програмного забезпечення для аудиту та аналізу маркетингового контенту»

Затверджена наказом від «29» травня 2024 р. № 12.

2. Строк подання студентом готової роботи — «16» грудня 2024 р.

3. Цільова установка та вхідні дані до роботи:

Мета дослідження полягає у розробці концепції програмного забезпечення для аудиту та аналізу маркетингового контенту, що сприятиме підвищенню ефективності маркетингової діяльності.

Об'єкт дослідження — процеси аудиту та аналізу маркетингового контенту.

Предмет дослідження — процеси управління проектами та методи розробки програмного забезпечення для автоматизації аудиту та аналізу маркетингового контенту.

4. Зміст роботи:

Аналіз актуального стану та проблем сфери аудиту та аналізу маркетингового контенту; огляд літературних та інформаційних джерел щодо

можливостей вирішення виявлених проблем; обґрунтування доцільності розробки та впровадження нового ПЗ для аудиту та аналізу контенту; формулювання технічного завдання на розробку у вигляді паспорту проєкту; розробка концептуальних та математичних моделей інформаційної системи; розрахунок економічної ефективності та прогнозування впливу впровадження ПЗ; побудова концептуальної та логічної моделей бази даних проєкту; розробка структури програмного забезпечення та інтерфейсів; планування організаційної структури та формування команди проєкту; декомпозиція робіт, розробка WBS та календарного плану проєкту; управління ресурсами, ризиками та ключовими показниками ефективності.

5. Перелік графічного матеріалу (слайдів): титульний слайд, актуальність дослідження та постановка проблеми, дерево причин та наслідків, об'єкт, предмет, мета та завдання роботи, концептуальна модель інформаційної системи, формалізація математичних моделей, концептуальна та логічна моделі бази даних, алгоритми та інтерфейси програмного забезпечення, діаграма Ганта та WBS проєкту, організаційна структура команди, календарний план, ключові показники ефективності (KPI), висновки.

6. Календарний план виконання роботи:

№ п/п	Назва частини роботи	План виконання роботи
1.	Вивчення літературних джерел з предмету дослідження	02.09.2024 — 22.09.2024
2.	Збір і вивчення матеріалів	23.09.2024 — 23.10.2024
3.	Складання розгорнутого плану кваліфікаційної роботи	24.10.2024 — 30.10.2024
4.	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін	31.10.2024

5.	Підготовка розділу 1 «Дослідження та обґрунтування доцільності та життєздатності проєкту»	01.11.2024 — 08.11.2024
6.	Підготовка розділу 2 «Математична постановка задачі дослідження»	09.11.2024 — 15.11.2024
7.	Підготовка розділу 3 «Розробка інформаційного та програмного забезпечення проєкту»	16.11.2024 — 22.11.2024
8.	Підготовка розділу 4 «Планування елементів управління проєктом»	23.11.2024 — 29.11.2024
9.	Оформлення кваліфікаційної роботи	30.11.2024 — 03.12.2024
10.	Передача кваліфікаційної роботи рецензенту для рецензування	16.12.24
11.	Передача кваліфікаційної роботи науковому керівникові	16.12.24
12.	Попередній захист кваліфікаційної роботи	16.12.24
13.	Перевірка роботи на плагіат	16.12.24
14.	Захист роботи	23.12.24

Дата видачі завдання «29» травня 2024 р.

Керівник роботи к.т.н., професор Віктор МОРОЗОВ

(посада, прізвище, ім'я)

(підпис)

Завдання прийняла до виконання студентка групи УПз-21

Аліна ТУРЧЕНКО

(прізвище, ім'я)

(підпис)

ЗМІСТ

АНОТАЦІЯ.....	7
ВСТУП.....	8
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ	13
1.1 Аналіз актуального стану та проблем сфери аудиту та аналізу маркетингового контенту	13
1.2 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем	17
1.3 Обґрунтування доцільності та економічної ефективності розробки та впровадження ПЗ для аудиту та аналізу контенту	23
1.4 Постановка задачі дослідження та формулювання технічного завдання на розробку у вигляді паспорту проєкту.....	29
РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	35
2.1 Розробка концептуальних моделей інформаційної системи	35
2.2 Формалізація математичних моделей для автоматизації процесів аудиту .	40
2.2.1 Математична модель парсингу вмісту сайту	44
2.2.2 Математична модель інтеграції з аналітичними інструментами	47
2.2.3 Математична модель оновлення даних сайту	49
2.2.4 Математична модель визначення цільової аудиторії сторінки.....	51
2.2.5 Математична модель прогнозування трафіку	52
РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ	55
3.1 Розробка концептуальної моделі бази даних проєкту	55
3.2 Побудова логічної моделі бази даних проєкту.....	58
3.3 Опис структури програмного забезпечення проєкту.....	61

3.4 Розробка алгоритмів та інтерфейсів програмного забезпечення проєкту ..	64
3.5 Опис технічного стеку програмного забезпечення проєкту.....	72
РОЗДІЛ 4. ПЛАНУВАННЯ ЕЛЕМЕНТІВ УПРАВЛІННЯ ПРОЄКТОМ	77
4.1 Розробка організаційної структури управління проєктом. Формування команди проєкту	77
4.2 Розробка ієрархічної структури робіт проєкту та матриці відповідальностей	83
4.3 Розробка календарного плану. Планування термінів проєкту.....	90
4.4 Планування ресурсів та бюджету проєкту	95
4.5 Ключові метрики ефективності у разі впровадження ПЗ	102
ВИСНОВКИ	108
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....	111
ДОДАТКИ.....	116

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

«Дослідження моделей управління проектом створення та впровадження програмного забезпечення для аудиту та аналізу маркетингового контенту»

Студентка: Турченко Аліна Сергіївна

Науковий керівник: Морозов Віктор Володимирович

Рік захисту — 2024.

Темою даної роботи було обрано «Дослідження моделей управління проектом створення та впровадження програмного забезпечення для аудиту та аналізу маркетингового контенту», предметною областю якої є управління проектами у сфері розробки інформаційних систем для автоматизації процесів аудиту та аналізу маркетингового контенту.

Метою підготовки роботи є розробка концепції ПЗ для автоматизованого аудиту та аналізу маркетингового контенту, а також обґрунтування можливостей його впровадження для підвищення ефективності бізнес-процесів.

Ціль проекту — розробити концептуальні та математичні моделі програмного забезпечення та сформулювати рекомендації щодо його реалізації та впровадження в бізнес-середовище.

Практична цінність роботи полягає у можливості застосування розроблених підходів до управління проектами в реальних умовах. Впровадження отриманих результатів дозволяє створити ПЗ для глибокого аналізу маркетингового контенту, а також підвищити оперативність прийняття управлінських рішень на основі автоматизованого збору і обробки даних.

Наукова новизна роботи полягає у розробці нових наукових моделей та рішень, використанні математичних моделей для оптимізації процесів.

Кваліфікаційна робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, списку використаних джерел та додатків.

У *першому розділі* проведено аналіз предметної області, визначено проблеми та обґрунтовано доцільність проекту і економічну ефективність.

Другий розділ містить концептуальні та математичні моделі інформаційної системи.

У *третьому розділі* деталізовано структуру ПЗ, створено концептуальні та логічні моделі бази даних, а також розроблено алгоритми та інтерфейс.

Четвертий розділ описує управління проектом, включаючи ієрархічну структуру робіт, календарне планування, управління ресурсами та бюджетом.

За результатами роботи зроблено *висновки* про доцільність розробки та впровадження досліджуваного ПЗ.

Робота містить 110 сторінок без додатків, 17 рисунків та 5 таблиць. Додатки складають 10 сторінок.

Ключові слова: управління проектами, програмне забезпечення, маркетинговий контент, аудит контенту, аналіз контенту, автоматизація, математична модель, концептуальна модель, ефективність, економічна доцільність, база даних, бізнес-процеси.

ВСТУП

Актуальність теми. Розробка нових програмних засобів для аудиту та аналізу маркетингового контенту є важливим завданням у сучасних умовах. Зі зростанням ролі цифрового маркетингу, актуальність комплексного підходу до оцінювання ефективності контенту стає дедалі очевиднішою. На сьогодні більшість існуючих інструментів, таких як HubSpot, Ahrefs, Semrush або Google Analytics, забезпечують лише часткову функціональність і вимагають значних зусиль для інтеграції та аналізу даних, що збільшує трудові витрати та ризики припущення помилок.

У цьому контексті є необхідність у створенні інтегрованого рішення, яке об'єднувало б усі функції аналізу маркетингового контенту в єдиному продукті, дозволяло автоматизувати процеси збору та обробки даних і надавало користувачам інструменти для глибокого аудиту. Такі рішення сприятимуть підвищенню ефективності бізнес-процесів, зменшенню помилок та економії ресурсів, що є стратегічно важливим у висококонкурентному середовищі.

Дане дослідження базується на розробці програмного забезпечення для аудиту та аналізу контенту, що передбачає використання системного підходу до створення концептуальних, математичних та інформаційних моделей, а також інтеграцію з ключовими аналітичними платформами та технологіями штучного інтелекту. Це дозволить створити інноваційне програмне рішення, здатне якісно змінити підхід до аналізу контенту та впровадження маркетингових стратегій.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана у рамках наукової діяльності кафедри технологій управління факультету інформаційних технологій Київського національного університету імені Тараса Шевченка. Тема дослідження відповідає загальному напрямку наукових досліджень кафедри, зокрема розробці інноваційних підходів до управління проектами в інформаційних системах.

Дослідження здійснювалося в контексті освітньо-професійної програми «Управління проектами» за спеціальністю 122 «Комп'ютерні науки», яка

передбачає впровадження сучасних методів управління та розробки інформаційних систем, орієнтованих на потреби бізнесу.

Робота пов'язана з виконанням завдань, передбачених навчальною та науковою програмами, що спрямовані на вдосконалення процесів автоматизації управління та аналізу в проєктах. Тематика дослідження була узгоджена із завданнями кафедри та підтримана науковим керівником, що забезпечує інтеграцію дослідження в загальний науковий контекст університету.

Отримані результати можуть бути використані як у подальших наукових дослідженнях, так і в практичній діяльності організацій, які займаються розробкою програмного забезпечення для маркетингової аналітики.

Мета дослідження — розробка унікального програмного забезпечення для аудиту та аналізу маркетингового контенту, яке забезпечить автоматизацію процесів збору, обробки та аналізу даних з використанням сучасних технологій, включаючи штучний інтелект та інтеграцію з ключовими аналітичними платформами (Google Analytics, HubSpot тощо). Це дозволить маркетологам та бізнес-аналітикам підвищити ефективність управління маркетинговими кампаніями, зменшити трудові витрати та підвищити якість прийняття рішень на основі отриманих даних.

Завдання дослідження:

1) Дослідження предметної області:

1. Дослідити сучасні підходи до автоматизації аудиту та аналізу маркетингового контенту.
2. Вивчити функціональні можливості та обмеження ключових аналітичних платформ.

2) Обґрунтування доцільності проєкту:

1. Визначити ключові проблеми, які виникають при використанні існуючих рішень для аналізу контенту.
2. Оцінити потребу в інтегрованому рішенні, яке б забезпечувало комплексний підхід до автоматизації процесів аналізу.

3) Розробка концепції проєкту:

1. Сформулювати технічне завдання на проєкт розробки ПЗ.
 2. Розробити концептуальну модель інформаційної системи, моделі бази даних та основні алгоритми.
- 4) Аналіз математичних моделей:
1. Формалізувати математичні моделі, які можуть бути застосовані для аналізу маркетингових даних.
- 5) Планування проєкту розробки програмного забезпечення:
1. Розробити ієрархічну структуру робіт (WBS).
 2. Скласти календарний план реалізації проєкту, включаючи основні етапи розробки та тестування.
 3. Описати організаційну структуру команди проєкту та розподіл ролей у проєкті.
- 6) Економічне обґрунтування проєкту:
1. Провести аналіз витрат і оцінити економічну ефективність проєкту.
 2. Спрогнозувати вплив запропонованого рішення на ефективність бізнес-процесів компаній-користувачів.
- 7) Рекомендації щодо впровадження:
1. Надати рекомендації щодо подальшої реалізації проєкту та інтеграції розробленого ПЗ в роботу організацій.

Об'єктом дослідження є процеси аудиту і аналізу маркетингового контенту в рамках вебсайтів.

Предметом дослідження виступають процеси управління проєктом із розробки програмного забезпечення для автоматизації аудиту та аналізу маркетингового контенту, зокрема управління ризиками, часом і ресурсами.

У процесі виконання магістерської роботи застосовувались різноманітні **методи дослідження** для досягнення поставленої мети та вирішення завдань:

- Аналіз літературних джерел — метод використано для вивчення існуючих підходів до управління проєктами, методик розробки програмного забезпечення та інструментів для аудиту і аналізу маркетингового

контенту. На основі аналізу сформовано базу теоретичних знань і виділено актуальні проблеми предметної області.

- Системний аналіз — застосовувався для розгляду складових частин проєкту, їх взаємозв'язків, структури та функціональності, а також для визначення ключових параметрів і вимог до розробки ПЗ.
- Метод моделювання — використовувався для побудови концептуальної та логічної моделі інформаційної системи, а також для формалізації математичних моделей, необхідних для автоматизації процесів аналізу маркетингових даних.
- Метод економічного аналізу — застосовувався для оцінки витрат на розробку та впровадження програмного забезпечення, розрахунку показників економічної ефективності проєкту і прогнозування впливу запропонованого рішення на бізнес-процеси.
- Проєктний метод — використовувався для розробки структури проєкту, планування його основних етапів (декомпозиція робіт, розробка календарного плану), визначення ресурсів і управління ризиками.
- Порівняльний аналіз — застосовувався для зіставлення можливостей існуючих аналітичних платформ з метою виявлення їх переваг, недоліків та формування вимог до нового рішення.

Новизна отриманих результатів полягає у тому, що було розроблено концептуальну модель програмного забезпечення для аудиту та аналізу маркетингового контенту. Вперше запропоновано комплексний підхід до проєктування інформаційної системи, що об'єднує математичні, концептуальні та інформаційні моделі, з урахуванням специфіки маркетингового контенту. Було формалізовано математичні моделі і удосконалено підходи до автоматизації процесів аналізу маркетингових даних шляхом використання адаптивних алгоритмів, здатних враховувати змінність ринкових умов і поведінки користувачів. Отримало подальший розвиток методологічне забезпечення управління проєктом розробки програмного забезпечення, включаючи деталізовану декомпозицію робіт, створення календарного плану та

оцінку ризиків. Вперше розроблено комплексну методику прогнозування економічної ефективності впровадження ПЗ, яка враховує як економію трудових ресурсів, так і підвищення якості маркетингових стратегій.

Практичне значення отриманих результатів. Результати магістерської роботи мають прикладне значення і можуть бути використані для планування, розробки та впровадження програмного забезпечення для аудиту та аналізу маркетингового контенту. Практичне значення роботи полягає у розроблених концептуальній та математичних моделях, що можуть бути використані як основа для розробки програмного забезпечення. Розроблені рекомендації щодо планування проєкту, управління ризиками, ресурсами та часом можуть бути впроваджені в проєктах з розробки програмного забезпечення, особливо у сфері маркетингових технологій. Методи оцінки економічної ефективності впровадження програмного забезпечення можуть бути застосовані в організаціях, що планують впровадження інформаційних систем, з метою оцінки потенційних вигід і ризиків. Результати дослідження мають високий ступінь готовності до практичного використання, зокрема для адаптації до потреб організацій у сфері автоматизації маркетингових процесів.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

1.1 Аналіз актуального стану та проблем сфери аудиту та аналізу маркетингового контенту

У сучасному цифровому середовищі аудит та аналіз контенту стали невід'ємною частиною успішної стратегії контент-маркетингу. Вони допомагають компаніям розуміти, як їх контент впливає на аудиторію, оцінювати ефективність контент-стратегій і виявляти області для покращення. Глибокий аналіз контенту дозволяє визначити, які матеріали приносять найбільшу користь, підвищують залученість користувачів та сприяють досягненню цілей бізнесу.

Аудит та аналіз контенту також допомагає: виявити теми, які ще не були висвітлені, для створення нового контенту; переконатися, що весь контент відповідає стилю бренду; знайти можливості для покращення взаємодії та залучення користувачів; знайти та втілити ідеї для збільшення кількості конверсій [48].

Ринок контент-маркетингу є одним з найбільш швидкозростаючих сегментів маркетингу. У 2022 році його обсяг оцінювався в 413,2 мільярда доларів США, і, за прогнозами, до 2032 року він досягне 2 трильйонів доларів США, зростаючи на 16,9% щороку з 2023 по 2032 рік [37]. Це підкреслює стратегічне значення контент-маркетингу для бізнесу різних галузей. Велика кількість компаній інвестує в контент-маркетинг, розуміючи його потенціал для залучення клієнтів, підвищення впізнаваності бренду та лояльності аудиторії.

У 2023 році компанія Semrush [44] провела опитування понад 1700 маркетингологів з усього світу, сформувавши по його результатам аналітичний звіт під назвою State of Content Marketing [46]. Фрагмент результатів даного дослідження з порівнянням підходів компаній, успішних та неуспішних у своїх контент-маркетинг активностях, підтверджує, що аналіз показників результативності контенту, відстеження коефіцієнту повернення інвестицій у

контент-маркетинг та проведення регулярних аудитів контенту є невід’ємною частиною успішної контент-стратегії (табл. 1.1).

Таблиця 1.1

Фрагмент результатів опитування Semrush “State of Content Marketing”

Характеристика	Частка респондентів за рівнем успішності контент-маркетингу			
	Дуже успішні	Успішні	Мінімально успішні	Неуспішні
Вимірюють коефіцієнт повернення інвестицій в контент-маркетинг	70%	68%	46%	56%
Вважають, що вимірюють цей показник ефективно	34%	23%	12%	3%
Вимірюють загальну результативність свого контенту	77%	73%	61%	50%
Здійснюють аудит контенту 2 або більше рази на рік	61%	59%	43%	29%
Використовують платформи для автоматизації маркетингу	18%	14%	8%	3%

Крім того, за результатами даного дослідження, 15% маркетологів віддають функцію аудиту та аналізу контенту на аутсорсинг [46], тобто використовують модель віддаленого найму, при якій бізнес-завдання компанії делегуються третім особам, фахівцям у певній сфері. При аутсорсингу компанія-виконавець часто отримує доступ до бізнес-процесів, сервісів, систем, інфраструктури компанії-замовника [8]. Це може нести ризики, адже результати аналізу та аудиту контенту є чутливими даними, що мають комерційну цінність. Також факт аутсорсингу цієї функції може свідчити про небажання або неможливість деяких маркетингових команд інвестувати власні час та ресурси у ці трудомісткі процеси.

Ще одним важливим відкриттям дослідження Semrush є те, що 54% опитаних маркетологів з різних країн світу здійснюють аудит та аналіз контенту регулярно [46], а саме 2 та більше разів на рік (рис. 1.1).

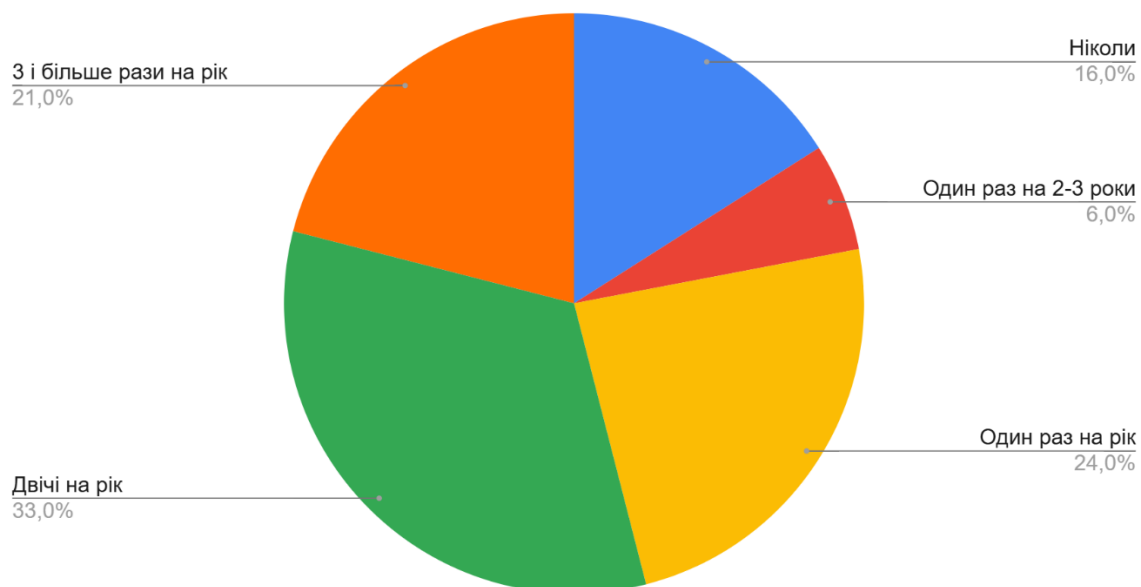


Рис. 1.1. Частота виконання аудиту контенту за даними Semrush

З цих даних можна зробити висновок про те, що переважна більшість компаній стикаються з необхідністю проведення аудиту та аналізу свого маркетингового контенту, тобто усвідомлюють його важливість і ймовірно стикалися з пов'язаними з цим процесом проблемами, а отже можуть потребувати нове рішення, яке допоможе їх вирішити.

Зазвичай проведення аудиту контенту складається з чотирьох основних етапів, на кожному з яких маркетологи можуть зіткнутися з відповідними проблемами:

1. Визначення обсягу та критеріїв аудиту. Необхідно вирішити, які сторінки, розділи або типи контенту слід включити або виключити з аудиту, а також які метрики чи показники використовувати для їх оцінки. Наприклад, це можуть бути трафік, конверсії, показник відмов, читабельність, ключові слова, посилання чи метадані. Залежно від поставлених цілей і доступних ресурсів, може виникнути потреба в пріоритизації чи обмеженні обсягу та критеріїв аудиту.
2. Збір та організація даних. Часто маркетологам доводиться вручну експортувати, очищувати, формувати чи об'єднувати дані з різних

джерел та перевіряти їх коректність. Для збору та організації використовуються платформи аналітики, електронні таблиці тощо. Важливо забезпечити точність, послідовність та актуальність даних. Також необхідно організувати дані у зручному для сортування, фільтрації чи групування за різними критеріями форматі. Таким чином, це перетворюється в довгий трудомісткий процес.

3. Аналіз та інтерпретація даних. Це може бути складним і суб'єктивним процесом, адже необхідно враховувати не лише кількісні показники, але й контекст, тенденції та закономірності, що стоять за ними. Для глибшого розуміння ефективності контенту може знадобитися сегментація даних за різними змінними, такими як аудиторія, тематика, формат чи канал. Також необхідно порівнювати отримані дані з поставленими цілями, еталонними значеннями або найкращими практиками, щоб виявити прогалини, можливості чи сфери для покращення.
4. Презентація та впровадження висновків. Необхідно представити результати аудиту в чіткій, лаконічній та придатній до використання формі, використовуючи візуальні елементи, такі як діаграми, таблиці чи схеми. Крім того, потрібно спланувати й реалізувати зміни або дії, що впливають із висновків, наприклад, оновлення, видалення, створення чи оптимізація контенту, а також відстежувати вплив цих зусиль [49].

Серед зазначених проблем чільне місце посідає проблема збору та організації даних. Основним інструментом, з якого беруться дані для проведення аудиту контенту є аналітичні платформи, такі як Google Analytics чи Ahrefs [35]. Вони забезпечують доступ до ключових показників — трафіку та конверсій, що дозволяє розділити якісний контент від менш ефективного [40]. Однак цей інструмент дає обмежене уявлення про результативність контенту, особливо в компаніях зі складними процесами залучення клієнтів.

Так, для всеохопного і якісного аудиту контенту існує потреба збирати та агрегувати інформацію з більш ніж одного джерела — це можуть бути як безкоштовні Google Analytics та Google Search Console, так і платні

інструменти, як-от вебсканер Screaming Frog [43], який дозволяє швидко імпортувати всі URL-адреси статей у таблицю чи аналізувати SEO-оптимізацію сторінок у великому масштабі [40], або HubSpot, за допомогою якого можна відслідкувати роль кожної сторінки сайту на шляху конвертації кожного відвідувача у покупця [39].

Однак використання додаткових інструментів спричинює появу більшої кількості даних і, відповідно, підвищення складності аналізу. В такому випадку обрані показники з кожного інструменту необхідно експортувати й імпортувати їх до електронної таблиці. Постає проблема розрізненості формату та вигляду даних у кожній з систем і виникає потреба проводити їх нормалізацію, що робиться маркетологами зазвичай вручну.

Отже, проведений аналіз актуального стану та проблем сфери аудиту та аналізу маркетингового контенту підкреслив, що існуючі ручні підходи демонструють суттєві обмеження, пов'язані з необхідністю використання багатьох інструментів, що ускладнює збір даних та збільшує витрати часу на аналіз, підвищуючи ймовірність виникнення помилок. А оскільки дані є основою прийняття рішень, низька якість та недостатня актуальність аналітики можуть стати бар'єром для реалізації успішних стратегій.

1.2 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем

Формулювання проблемної області є важливим етапом у проектуванні та реалізації будь-якого програмного продукту. Основна мета цього етапу полягає в ідентифікації ключових проблем, які необхідно вирішити, та визначенні цілей, до яких потрібно прагнути для досягнення успіху. Для структурованого підходу до цього процесу часто використовуються методи створення дерев проблем та дерев цілей.

Дерево проблем допомагає виявити основні проблеми, з якими стикаються користувачі або організації, та розібратися в їхніх першопричинах.

вона допомагає виявити ключові фактори, які впливають на ефективність процесу контент-аудиту, та визначити області, що потребують покращення.

Дерево цілей (рис. 1.3) було створено для визначення шляхів вирішення проблем, виявлених на попередньому етапі.

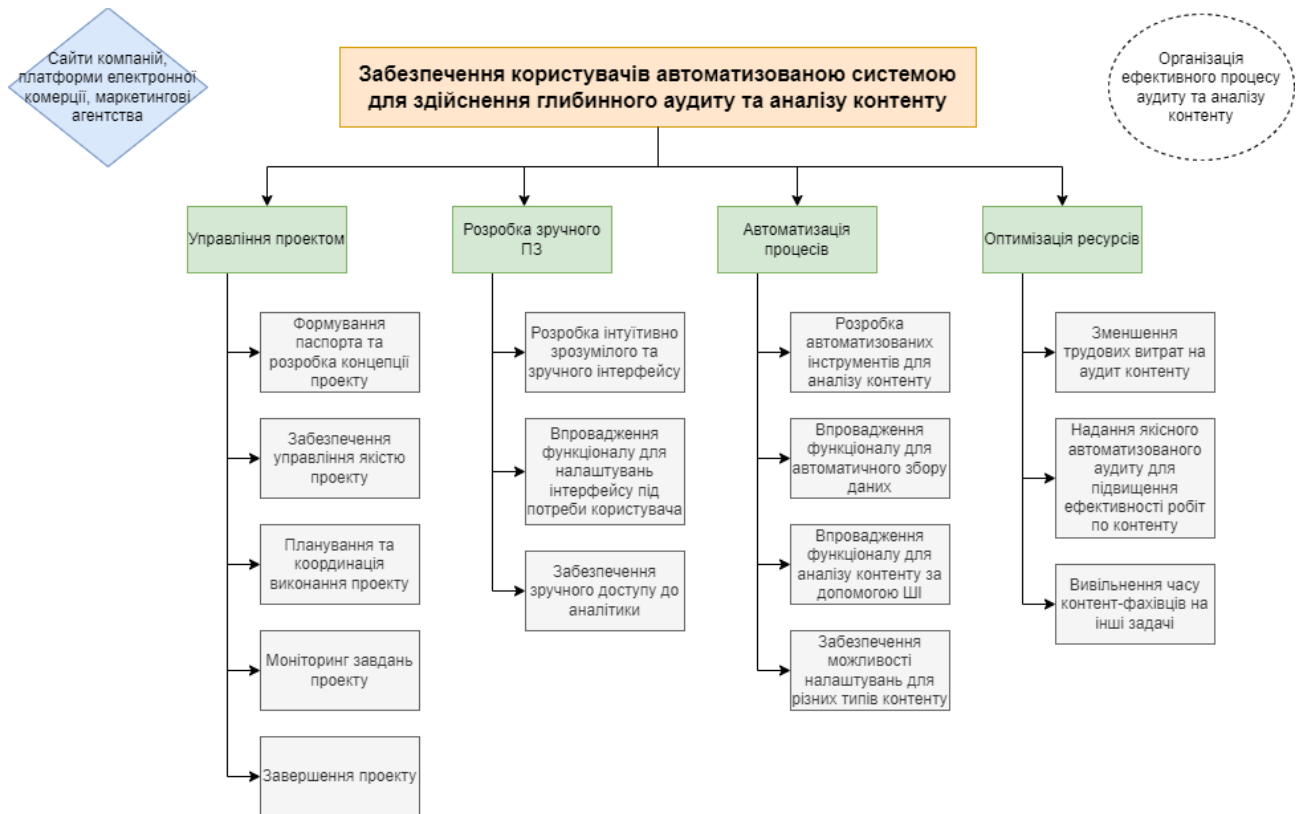


Рис. 1.3. Дерево цілей

Дерево цілей — відображення ідеального бажаного стану, максимально позитивної ситуації, якої можна досягти через вирішення наявних проблем. На цій діаграмі представлені основні цілі, такі як управління проектом, розробка зручного ПЗ, автоматизація процесів та оптимізація ресурсів. Основними завданнями є створення інтуїтивно зрозумілого інтерфейсу, впровадження функціоналу для автоматизованого збору та аналізу даних, а також забезпечення ефективного використання ресурсів. Ця діаграма служить дорожньою картою для реалізації проекту, допомагаючи визначити пріоритети та розробити конкретні кроки для досягнення поставлених цілей.

Використання дерев проблем та цілей дозволяє глибше зрозуміти існуючі проблеми та визначити шляхи їх вирішення. Крім того, проробка дерева

проблем дозволяє перейти до проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем.

У ході аналізу інформаційних джерел було виявлено, що контент-маркетологи, які стикаються з виявленими проблемами і намагаються пришвидшити процеси аудиту, пробують вирішити це завдання шляхом використання систем управління контентом (CMS, Content Management Systems) або інструментів управління проєктами замість звичайних електронних таблиць. Так, контент-фахівець Жанета Стиллова вважає, що найважливіше в процесі аудиту контенту — це консолідувати весь контент, який потрібно регулярно переглядати, в одному місці — наприклад, у CMS або в інструменті управління проєктами, такому як Asana. За її словами, хоча це може зайняти багато часу на початковому етапі, у майбутньому це зекономить контент-маркетологам значну кількість часу, оскільки у них буде єдине «джерело правди», на яке можна спиратися, і їм не доведеться шукати необхідні дані вручну [41].

Також на ринку існує ПЗ, що позиціонується як рішення для автоматизації аудитів контенту. Наприклад, одним з таких рішень є вебдодаток PrioNow [47]. Єдиною підтримуваною ним інтеграцією є Google Search Console, тож дане рішення не можна вважати достатнім для проведення якісного і повного аудиту контенту. При цьому перевагою цього рішення є те, що воно здатне самостійно сформулювати висновки по сторінкам сайту відповідно до інформації з Google Search Console, підсвітивши рекомендовані дії. Таким чином, дане рішення не дає повного уявлення про результативність контенту, зосереджуючись лише на одному аспекті аудиту — оптимізації контенту для пошукової видачі. При цьому підписка на це рішення вартує 13,99 доларів США на місяць.

Існують також рішення для проведення аудиту контенту сайтів, створених на платформі WordPress. Наприклад, MonsterInsights та Analytify, які інтегруються з Google Analytics [47].

Основними перевагами MonsterInsights є простота встановлення, відстеження ефективності електронної комерції та інформація щодо взаємодії

користувачів. Водночас базова версія цього інструменту має обмежений функціонал і вимагає придбання платного плану. Вартість базового плану починається від 199,50 доларів США на місяць, що робить його доволі дорогим варіантом для невеликих компаній чи стартапів. Крім того, цей інструмент підходить виключно для сайтів на платформі WordPress.

Analytify також забезпечує відображення метрик у зручному інтерфейсі, включаючи дані в реальному часі, аналіз кампаній і поведінку відвідувачів. Основними обмеженнями є брак функцій у безкоштовній версії та недостатність можливостей для виконання розширених аналітичних задач. Вартість платного плану починається від 39 доларів США на місяць, що є більш доступним варіантом, але й функціонал такого рішення є значно обмеженим.

Якщо розглядати сегмент більших гравців ринку, варто відзначити рішення на кшталт Zapier — універсальної платформи для автоматизації, яка інтегрує тисячі додатків і сервісів, дозволяючи маркетологам автоматизувати рутинні процеси та оптимізувати робочі потоки [50]. Zapier пропонує функціонал для автоматизації робочих процесів між різними додатками, що може бути корисним для управління контентом. Основні переваги інструменту включають:

- Підтримку тисяч інтеграцій, що дозволяє об'єднати різні системи управління контентом та аналітики.
- Економію часу завдяки автоматизації рутинних завдань, таких як перенесення даних або налаштування сповіщень.
- Простий інтерфейс, який підходить навіть для користувачів без технічного досвіду.

Однак Zapier має ряд суттєвих обмежень, які впливають на його ефективність для проведення аудиту контенту. Наприклад, Zapier не надає власних функцій для аналітики або оцінки ефективності контенту, таких як аналіз ключових показників, пошукова оптимізація або аудит за критеріями якості. План Starter коштує від 19,99 доларів США на місяць, але для більших

проектів можуть знадобитися дорожчі підписки, що створює додатковий фінансовий бар'єр.

Таким чином, хоча Zapier може бути корисним для автоматизації загальних задач, його функціонал обмежений для потреб повноцінного контент-аудиту. Цей інструмент може бути ефективним лише як додатковий елемент для автоматизації рутинних процесів, але він не покриває потреби в аналізі даних чи оптимізації контенту. Для розв'язання таких завдань потрібні спеціалізовані рішення з можливостями інтеграції та автоматизації, які були б розраховані на комплексну роботу з великими обсягами контенту.

Існують також рішення, що позиціонуються як інструменти для інтеграції даних, наприклад Supermetrics, який дозволяє маркетологам збирати дані з різних джерел у зручному форматі. Цей інструмент підтримує інтеграцію з великою кількістю платформ, таких як Google Analytics, Facebook Ads, HubSpot, LinkedIn, що робить його корисним для здійснення аудиту та аналізу маркетингового контенту [45].

Основними перевагами Supermetrics є те, що інструмент дозволяє збирати дані з різних платформ у одному місці, наприклад, в Google Sheets або Excel, що значно спрощує аналіз. Supermetrics автоматично оновлює дані в таблицях, зменшуючи час, витрачений на ручний збір і обробку інформації. Інструмент підходить для великих обсягів даних, що дозволяє використовувати його для роботи з великими проектами.

Проте Supermetrics лише збирає дані, але не надає можливостей для глибокого аналізу або оцінки ефективності контенту, таких як аналіз читабельності, ключових слів чи показників пошукової оптимізації. Інструмент пропонує платні плани, які починаються від 69 доларів США на місяць, але для більш складних завдань потрібні розширені підписки, що може бути фінансово обтяжливим для малих бізнесів.

Проведений аналіз літературних та інформаційних джерел дозволив детально розглянути сучасні підходи до вирішення виявлених проблем у сфері контент-аудиту. Результати дослідження показали, що, незважаючи на наявність

великої кількості рішень на ринку, жоден із них не здатний повністю покрити всі аспекти, необхідні для комплексного контент-аудиту. Це обумовлено як технічними, так і фінансовими обмеженнями.

Це підкреслює необхідність створення універсального рішення, яке б дозволяло автоматизувати весь процес аудиту контенту, включаючи збір, аналіз і оцінку даних з кількох джерел.

1.3 Обґрунтування доцільності та економічної ефективності розробки та впровадження ПЗ для аудиту та аналізу контенту

Аналіз існуючих рішень для аудиту контенту свідчить про те, що доступні інструменти не задовольняють повною мірою потреби сучасного ринку, створюючи тим самим потребу у розробці нового програмного забезпечення.

Одна з основних проблем наявних інструментів для контент-аудиту полягає у їхній вузькій спеціалізації та фрагментарності функціоналу. Платформи аналітики, такі як Google Analytics і Google Search Console, є основними джерелами даних для проведення аудиту контенту. Вони надають важливу інформацію про трафік, поведінку користувачів, кліки та інші метрики, що дозволяють оцінити ефективність контенту. Інструменти такого типу є незамінними для маркетологів, але вони обмежуються збором даних і потребують додаткових кроків для подальшої їхньої обробки та аналізу. Окремо існують рішення, спрямовані на спрощення аудиту контенту, такі як PrioNow, MonsterInsights, Analytify, Zapier та Supermetrics. Вони пропонують функціонал, що дозволяє інтегрувати дані з платформ аналітики та автоматизувати певні аспекти аудиту [47].

Попри це, як платформи аналітики, так і спеціалізовані рішення мають суттєві обмеження. Перші обмежуються збором даних і не пропонують інструментів для автоматизованого аудиту чи створення висновків. Другі, попри автоматизацію окремих завдань, не забезпечують комплексного підходу до аудиту контенту, оскільки їхній функціонал часто орієнтований на специфічні

задачі або обмежений певними платформами. У результаті маркетологи змушені використовувати кілька інструментів одночасно, що збільшує кількість ручної роботи, ускладнює інтеграцію даних і витрачає значний обсяг часу. Крім того, більшість інструментів не враховують важливість автоматизації. Значна частина завдань, таких як збір даних, їх очищення, формування звітів і проведення первинного аналізу, виконується вручну. Це не лише значно збільшує навантаження на маркетологів та аналітиків, але й підвищує ймовірність виникнення помилок. Ручна обробка великих обсягів даних призводить до недоліків у якості аналізу та втрати цінної інформації. Наприклад, маркетологи, які хочуть включити аналіз цільової аудиторії або короткий опис вмісту сторінок, змушені самостійно вводити ці дані, витрачаючи на це час і ресурси. Висока вартість інструментів також створює бар'єри для багатьох компаній, особливо для малого та середнього бізнесу.

Особливу увагу слід приділити відсутності інтеграції інструментів з можливостями штучного інтелекту (ШІ). Жоден із популярних інструментів не пропонує функцій автоматичного аналізу контенту за допомогою ШІ [42], таких як оцінка цільової аудиторії, побудова персоналізованих рекомендацій або автоматизоване визначення слабких місць у контенті.

Запропоноване ПЗ для аудиту та аналізу контенту має значний конкурентний потенціал, що базується на його унікальних функціональних можливостях, орієнтованих на усунення недоліків існуючих рішень. Однак для повноти оцінки доцільності його розробки та впровадження необхідно проаналізувати його економічну ефективність.

Оскільки тема дослідження передбачає вивчення як процесів створення, так і впровадження ПЗ для аудиту та аналізу маркетингового контенту, варто розглянути економічну ефективність як створення цього проекту з точки зору стейкхолдерів, так і економічну ефективність для компаній-клієнтів, які будуть впроваджувати це рішення на своїх підприємствах.

Розпочнемо з розрахунку економічної ефективності проекту створення даного ПЗ. Для цього можна використати підхід, що базується на оцінці витрат

та очікуваних вигод для стейкхолдерів. Першим кроком є визначення витрат на створення ПЗ. Для оцінки витрат на розробку ПЗ слід врахувати такі компоненти:

- Прямі витрати: заробітна плата членів команди; ліцензії (наприклад, ПЗ для розробки); обчислювальні ресурси (сервери, хмарна інфраструктура).
- Непрямі витрати: витрати на управління проектом, маркетинг, операційні витрати (офісні витрати, підтримка технічної інфраструктури).
- Одноразові витрати: реєстрація інтелектуальної власності тощо.

Для проведення точного розрахунку необхідно розуміти ціну планів підписки та кількість користувачів, яких необхідно залучити. За SaaS бізнес-моделі існують такі підходи до ціноутворення: ціноутворення на основі використання (Usage Pricing); ціноутворення за кількістю користувачів (User-Count Pricing); модель фіксованої ставки (Flat-Rate Pricing); модель за функціоналом (Per-Feature Pricing); модель Freemium; проникнення на ринок (Penetration Pricing); ціноутворення на основі конкуренції (Competitor-Based Pricing); ціноутворення на основі цінності (Value-Based Pricing) [38].

Оптимальною для даного ПЗ була б модель ціноутворення на основі використання — це дозволило б створити прайсінг плани і передбачити умови, за яких за використання платформи у більшому обсязі, ніж передбачено обраним планом, знімається додаткова плата. Однак для розрахунку ціни за цією моделлю необхідно мати готове рішення і показники його використання. Тому наразі для потреб дослідження застосуємо модель ціноутворення на основі конкуренції.

Ціни на контент-маркетингові програмні рішення коливаються від 14 доларів США за найпримітивнішу функціональність до 200 і більше доларів США за продукти, наближені за можливостями до досліджуваного ПЗ. Таким чином можна припустити, що ціна за щомісячну підписку у розмірі 65 доларів США є адекватною і конкурентоспроможною. Встановимо також ціль здобути 1000 користувачів продукту після його реалізації та релізу.

Тепер ми можемо приблизно оцінити економічну ефективність від створення ПЗ шляхом розрахунку коефіцієнта рентабельності інвестицій (ROI) [33] за наступною формулою:

$$ROI = \frac{Rev - Exp}{Exp} * 100\% \quad (1.1)$$

де ROI — коефіцієнт рентабельності інвестицій;

Rev — очікуваний дохід;

Exp — витрати.

Очікуваний дохід складе 65 (ціна підписки) * 1000 (кількість користувачів) * 41.50 (приблизний курс долара США) = 2,697,500 грн. Витрати відповідно до бюджету, закладеного у паспорті проєкту, складуть 2,173,000 грн. Тоді коефіцієнт рентабельності інвестицій складатиме 24.13%. Цей позитивний і досить високий показник свідчить про рентабельність розробки даного програмного забезпечення.

Розглянемо ще один, дещо складніший спосіб оцінки економічної ефективності проєктів — розрахунок чистої приведеної вартості.

Чиста приведена вартість (NPV, Net Present Value) — це показник економічної ефективності інвестиційного проєкту, який розраховується шляхом дисконтування (приведення до поточної вартості, тобто на момент інвестування) очікуваних грошових потоків (як доходів, так і витрат). Вона відображає прибуток інвестора після того, як грошові притоки окуплять його початкові інвестиційні витрати та відтоки, пов'язані зі здійсненням такого проєкту [28].

NPV розраховується за формулою:

$$NPV = \sum_{t=1}^n \frac{CF_t}{(1+r)^t} \quad (1.2)$$

де NPV (Net Present Value) — чиста приведена вартість;

CF (Cash Flow) — грошовий потік;

r — ставка дисконтування;

n — загальна кількість періодів (інтервалів, кроків);

t — часовий період (у роках).

Для розрахунку NPV необхідно скласти прогнозний графік грошових потоків за інвестиційним проектом у періоді. Потоки повинні включати як доходи, так і витрати (інвестиції та інші витрати з реалізації проекту) [28].

Зробимо приблизний прогноз грошових потоків на 5 років (табл. 1.2). Припустимо, що постійні витрати у перший рік складають приблизно 1,303,800 грн щорічно (60% від витрат на створення ПЗ) і поступово наростають через збільшення кількості клієнтів, а отже і витрат на інфраструктуру. Ставка дисконту — відсоток, що враховує вартість грошей у часі (зазвичай береться від 10% до 20% для технологічних проектів) — використаємо середнє значення (15%). Якщо для інвестування буде використано позикові кошти банку, то ставкою дисконтування буде ефективна ставка відсотка за кредитом. Якщо ж буде використано власні кошти інвестора, то за ставку дисконтування може бути взята ставка відсотка за банківським депозитом тощо [28].

Таблиця 1.2

Прогноз грошових потоків на 5 років

Рік	Очікувані доходи (грн)	Очікувані витрати (грн)	Чистий потік (грн)	Дисконтований потік (грн)
1	32,370,000 (1000 користувачів)	1,303,800	31,066,200	$\frac{31,066,200}{1.15^1} = 26,973,217$
2	38,844,000 (1200 користувачів)	1,402,000	37,442,000	$\frac{37,442,000}{1.15^2} = 28,362,139$
3	46,612,800 (1440 користувачів)	1,609,500	45,003,300	$\frac{45,003,300}{1.15^3} = 29,604,185$
4	55,935,360 (1728 користувачів)	1,800,600	54,134,760	$\frac{54,134,760}{1.15^4} = 31,222,719$
5	67,122,432 (2073 користувачі)	1,900,800	65,221,632	$\frac{65,221,632}{1.15^5} = 33,495,238$

Таким чином, сума дисконтованих потоків складе 149,657,498 грн. Вирахуємо з цієї суми початкові інвестиції у 2,173,000 грн і отримуємо чисту приведену вартість (NPV) у розмірі 147,484,498 грн. Цей високий показник підтверджує економічну ефективність розробки даного рішення, хоча звісно дані розрахунки дуже приблизні та грубі, і потребують уточнення як у прибутковій, так і у витратній частині.

Для розрахунку економічної ефективності впровадження ПЗ слід проаналізувати середні поточні витрати часу та вартість таких процесів, порівнявши з потенційними заощадженнями після автоматизації.

Для цього скористаємося прикладом з бази виробничої практики. Щомісяця на ручну агрегацію та нормалізацію даних та інші процеси, пов'язані з аудитом та аналізом контенту без спеціального інструменту, витрачається приблизно 22 годин ліда команди контенту (близько 13% щомісячного робочого часу) та 12 годин кожного з двох контент спеціалістів (близько 7% щомісячного робочого часу). Для розрахунків можемо припустити, що заробітна плата ліда команди становить 80000 грн на місяць, а спеціалістів — по 55000 грн. Тоді компанія витрачає на виконання цієї роботи лідом команди контенту 10400 грн на місяць ($80000 * 0.13$), а спеціалістами — 7700 грн на місяць ($55000 * 0.07 * 2$). У такому випадку щомісячні витрати на ручний аудит становлять 18100 грн.

Запропоноване ПЗ автоматизує більшу частину процесів, пов'язаних з аудитом та аналізом контенту, та залучає ІІІ для виконання дій, які зараз виконують фахівці вручну, тож можна припустити, що щонайменше 70% цього часу вказаних трьох спеціалістів вдасться заощадити для інших, більш стратегічно важливих та економічно вигідних ініціатив.

Тоді компанія витрататиме на виконання аудиту лідом команди контенту 3120 грн на місяць ($80000 * (0.13 - 70\%)$), а спеціалістами — 2310 грн на місяць ($55000 * (0.07 - 70\%) * 2$). Отже, загальні щомісячні витрати після впровадження ПЗ знизяться до 5430 грн. Щомісячна економія складе 12670 грн, однак від цієї суми варто також відняти вартість щомісячної підписки на ПЗ для аудиту та аналізу маркетингового контенту. Для розрахунків вище ми використовували

вартість підписки у 65 доларів США, тобто близько 2697.5 грн. Таким чином, компанія економитиме 9972.5 грн щомісяця.

Отже, очікувана економічна ефективність може бути обумовлена такими факторами вигоди:

- Економія часу — автоматизація збору та аналізу даних щонайменше на 70% знизить час, необхідний для цих процесів.
- Підвищення точності та якості аналізу — уникнення помилок при ручному введенні даних та підвищення якості аналітичних висновків можуть призвести до кращих маркетингових рішень та підвищення ROI (Return on Investment) — показника рентабельності інвестицій, який використовується для оцінки їх ефективності [33].
- Оптимізація маркетингових витрат — більш точний аналіз ефективності різних каналів та кампаній дозволить оптимізувати маркетинговий бюджет, перерозподіляючи кошти в найбільш ефективні канали.

Таким чином, хоча впровадження ПЗ вимагатиме додаткових витрат від компаній у розмірі вартості щомісячної підписки, потенційна економія часу та підвищення ефективності маркетингових процесів можуть значно перевищити ці витрати. Покращення точності даних, оптимізація маркетингових витрат, та здатність швидко приймати обґрунтовані рішення можуть призвести до підвищення загального ROI маркетингових кампаній.

1.4 Постановка задачі дослідження та формулювання технічного завдання на розробку у вигляді паспорту проєкту

Проєкт розробки ПЗ для аудиту та аналізу маркетингового контенту є важливим кроком у вирішенні існуючих проблем контент-маркетологів та власників вебсайтів. Цей розділ спрямований на формулювання основних задач дослідження, які ляжуть в основу технічного завдання, та визначення структурованого підходу до створення унікального продукту.

Основною метою дослідження є створення технічних рішень для автоматизації аудиту контенту, інтеграції з популярними аналітичними

платформами та впровадження можливостей штучного інтелекту для аналізу контенту.

У межах цього проєкту передбачається виконання таких ключових завдань, як:

1. Ідентифікація функціональних та нефункціональних вимог. Дослідження потреб користувачів та ринку дозволило визначити вимоги до функціоналу: автоматичний парсинг контенту, інтеграція з Google Analytics та HubSpot, кастомізація даних користувачами, а також сповіщення про важливі події. Не менш важливими є нефункціональні аспекти: зручний інтерфейс, безпека даних і масштабованість.
2. Розробка технічного завдання. Формулювання чітких технічних вимог забезпечує структурованість у реалізації проєкту. Технічне завдання передбачає створення хмарного Software as a Service (SaaS) рішення, доступного через веббраузер, з можливістю роботи на будь-яких сучасних пристроях.
3. Впровадження автоматизації аналізу даних. Розробка алгоритмів для автоматичного збору та обробки даних забезпечить мінімізацію ручної роботи, що зменшить ризики людських помилок та підвищить ефективність процесів аудиту.
4. Інтеграція з аналітичними платформами та штучним інтелектом. Програмне забезпечення має підтримувати інтеграцію з Google Analytics, HubSpot і штучним інтелектом для автоматичного аналізу змісту сторінок, визначення цільової аудиторії та генерації персоналізованих рекомендацій.
5. Оптимізація інтерфейсу. Створення зручного та інтуїтивно зрозумілого інтерфейсу, що дозволить користувачам легко взаємодіяти з продуктом незалежно від їхнього рівня технічної підготовки.
6. Проведення тестування та впровадження. Продукт буде протестований із залученням бета-тестерів, що забезпечить зворотний зв'язок і дозволить усунути можливі недоліки перед впровадженням.

Для забезпечення успішної реалізації проєкту важливим етапом є визначення функціональних та нефункціональних вимог до ПЗ. Ці вимоги є основою для формування технічного завдання та забезпечують чітке розуміння ключових характеристик і можливостей продукту.

Функціональні вимоги — це певна функціональність, яку має виконувати система. Ці вимоги описують вхідні й вихідні дані та взаємодію з користувачем. Нефункціональні вимоги відомі також як атрибути якості — це характеристики системи, які не пов'язані безпосередньо з певною функціональністю. Ці вимоги описують, як має поводитися і функціонувати система, щоб задовольняти потреби користувачів і стейкхолдерів. Функціональні вимоги описують, що система повинна робити, тоді як нефункціональні вимоги описують, як саме вона повинна це робити. Ці вимоги є ключовими для підтвердження, що система відповідає своєму призначенню та забезпечує позитивний досвід користування. Вони також відіграють важливу роль у визначенні загального успіху проєкту [10].

Отже, функціональні вимоги описують, що саме повинно виконувати програмне забезпечення, наприклад, автоматичний парсинг контенту, інтеграція з Google Analytics та HubSpot, або можливість створення кастомізованих звітів. Нефункціональні вимоги, натомість, фокусуються на якісних характеристиках продукту, таких як зручність використання, безпека, масштабованість та швидкодія. Ці вимоги є не менш важливими, адже саме вони забезпечують стабільність роботи системи та комфорт для кінцевих користувачів.

Перехід від загального бачення проєкту до детальних функціональних і нефункціональних вимог (див. Додаток А) дозволяє структурувати підхід до розробки програмного забезпечення.

Список вимог, сформованих на основі дослідження потреб ринку та потенційних користувачів, було деталізовано під час проходження виробничої практики, а саме:

- Парсинг контенту та метаданих. ПЗ має забезпечувати можливість автоматизованого збору метаданих із вебсторінок, включно з заголовками,

ключовими словами, зображеннями, а також інтерпретувати HTML-код для отримання цільової інформації.

- Інтеграція з аналітичними платформами. Маркетинговою командою бази практики було підкреслено важливість інтеграції з Google Analytics і HubSpot як необхідного мінімуму для отримання ключових маркетингових метрик. Завдяки цим інтеграціям користувачі зможуть автоматично отримувати показники трафіку, конверсій, MQL та SQL, показник відмов і середній час взаємодії з контентом.
- Аналіз контенту з використанням ШІ. Спеціалісти маркетингової команди відзначили цінність використання штучного інтелекту для розпізнавання вмісту сторінок, визначення цільової аудиторії, ключових слів і сегментів аудиторії, які найбільше підходять для даного контенту. На сьогодні подібна робота здійснюється вручну, що займає багато часу та ресурсів.
- Гнучкі налаштування звітності. Маркетингова команда підкреслила необхідність мати можливість самостійно налаштовувати звіти у ПЗ. Це передбачає включення кастомних колонок, додавання нотаток, застосування фільтрів для сегментації даних, що відповідає специфічним аналітичним запитам кожного користувача.
- Сповіщення про зміну показників. Одним із пріоритетів стало впровадження функціоналу сповіщень, який інформуватиме команду про суттєві зміни в ефективності контенту, зокрема про падіння трафіку чи зменшення рівня взаємодії. Команда також зазначила важливість можливості налаштовувати параметри таких сповіщень, аби система надсилала повідомлення лише за умови досягнення певних показників.

Оцінка функціональних вимог у співпраці з командою маркетингу дозволила деталізувати найважливіші аспекти майбутнього ПЗ, враховуючи специфіку його використання та потреби кінцевих користувачів.

Ще одним важливим документом, що формує основу для реалізації проєкту, є паспорт проєкту. Він містить структуровану інформацію про всі аспекти реалізації — від цілей до обмежень та очікуваних результатів. Паспорт

проєкту слугує основою для управління проєктом, визначаючи його ключові параметри та забезпечуючи комунікацію між зацікавленими сторонами [2].

Паспорт проєкту є важливим інструментом на етапі планування, адже він не лише документує цілі та задачі, але й встановлює обмеження щодо бюджету, часу та ресурсів. Окрім цього, документ описує унікальність продукту, його конкурентні переваги та визначає зацікавлені сторони, кожна з яких має свої інтереси в успішній реалізації проєкту. Визначення очікуваних результатів у паспорті проєкту дозволяє оцінити ефективність досягнення поставлених цілей.

Паспорт проєкту виконує декілька важливих функцій:

- Ініціювання проєкту. Паспорт допомагає офіційно розпочати проєкт, чітко сформулювавши його рамки.
- Комунікація. Документ забезпечує ефективну комунікацію між усіма зацікавленими сторонами, створюючи єдине бачення.
- Контроль та управління. Паспорт є основою для моніторингу та управління проєктом, дозволяючи оцінювати його прогрес і відповідність початковим параметрам.
- Прийняття рішень. Паспорт містить важливу інформацію для прийняття управлінських рішень на різних етапах реалізації [2].

У проєкті створення та впровадження програмного забезпечення для аудиту та аналізу маркетингового контенту паспорт проєкту (див. Додаток Б) слугує дорожньою картою для реалізації інноваційного програмного забезпечення, деталізуючи всі аспекти його створення та впровадження. Він дозволяє:

- Зафіксувати основні характеристики продукту, такі як унікальність і орієнтованість на кінцевого користувача.
- Чітко сформулювати цілі: автоматизація аудиту, інтеграція з аналітичними платформами, забезпечення кастомізації тощо.
- Координувати роботу міждисциплінарної команди спеціалістів.

- Сформувати реалістичний план реалізації, враховуючи часові та фінансові обмеження.

Таким чином, паспорт проекту є основоположним документом, який забезпечує системний підхід до планування, реалізації та управління проектом. Його правильне оформлення сприяє успішному виконанню поставлених задач і досягненню стратегічних цілей.

Детальний опис функціональних і нефункціональних вимог, а також формулювання паспорту проекту дозволяє структурувати процес розробки та забезпечити його відповідність як технічним стандартам, так і потребам кінцевих користувачів. Ці аспекти є ключовими для створення успішного та конкурентоспроможного продукту. Отже, розробка програмного забезпечення базуватиметься на чітко сформульованих задачах і технічному завданні.

РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1 Розробка концептуальних моделей інформаційної системи

Концептуальна модель проєкту є основою для розробки інформаційної системи для аудиту маркетингового контенту. Вона включає основні компоненти та взаємозв'язки між ними, що допомагає візуалізувати структуру системи та зрозуміти, як різні частини проєкту співпрацюють один з одним для досягнення кінцевої мети [2].

Ця модель визначає ключові компоненти системи, їх взаємозв'язки та функції, що дозволяє формалізувати підхід до створення програмного забезпечення. Вона слугує орієнтиром для всіх учасників розробки на етапах планування, впровадження та тестування системи.

Концептуальна модель має кілька важливих функцій:

- Візуалізація проєкту — концептуальна модель допомагає ідентифікувати ключові елементи системи, забезпечуючи всім зацікавленим сторонам (розробникам, аналітикам, менеджерам проєктів) зрозуміле представлення загальної структури продукту. Це дає змогу уникнути помилок і дублювання функцій на етапі проєктування.
- Планування та управління — модель слугує основою для планування ресурсів, розподілу завдань та управління проєктом. На основі моделі визначається обсяг необхідних ресурсів, розподіл функціональності між компонентами та оптимізація технологічних рішень.
- Документація — концептуальна модель є частиною документації проєкту, яка використовується для подальшої роботи над проєктом. Концептуальна модель забезпечує основу для створення математичних моделей, алгоритмів обробки даних та оптимізації функціональних процесів. Завдяки чіткій структуризації елементів і зв'язків модель допомагає контролювати, чи відповідає реалізація початково визначеним цілям.

Концептуальна модель є ключовим елементом процесу розробки інформаційної системи, оскільки вона забезпечує системний підхід до

створення програмного продукту, який враховує всі аспекти його функціонування. Модель виконує роль дорожньої карти, що дозволяє учасникам проєкту ефективно координувати свої дії, зосереджувати увагу на ключових завданнях та уникати можливих помилок на етапах проєктування, впровадження і тестування.

Окрім цього, концептуальна модель забезпечує чітке розуміння структури проєкту та його функціональних можливостей для всіх зацікавлених сторін, можливість адаптації до змін у вимогах або бізнес-цілях завдяки модульному підходу, ефективний розподіл ресурсів, що сприяє оптимізації процесів розробки та зменшенню ризиків перевищення бюджету чи часових рамок.

Таким чином, розробка концептуальної моделі є основоположним кроком у реалізації проєкту, який визначає його технічний напрямок, формує основу для подальших етапів розробки і сприяє створенню конкурентоспроможного продукту, що відповідає потребам кінцевих користувачів та вимогам ринку.

Концептуальна модель програмного забезпечення для аудиту та аналізу маркетингового контенту (рис. 2.1) визначає основну архітектуру системи, включаючи ключові компоненти, взаємозв'язки між ними та їхні функції. Вона виступає основою для створення інтегрованого та ефективного продукту, який відповідає потребам користувачів і сучасним технічним стандартам.

Ключові елементи моделі охоплюють надсистему, зовнішнє середовище, вхідні дані та вихідні результати, програмне забезпечення та його компоненти. Наприклад, модуль автоматичного парсингу забезпечує збір контенту та метаданих з вебсайтів, тоді як модуль штучного інтелекту проводить аналіз вмісту, генерує описи сторінок та визначає цільову аудиторію. Модуль «Інтеграції» підтримує взаємодію з платформами Google Analytics та HubSpot, що дозволяє отримувати ключові маркетингові метрики, необхідні для оцінки ефективності контенту.

Результатом роботи системи є зручний звіт про аудит контенту, який стає основою для прийняття управлінських рішень.

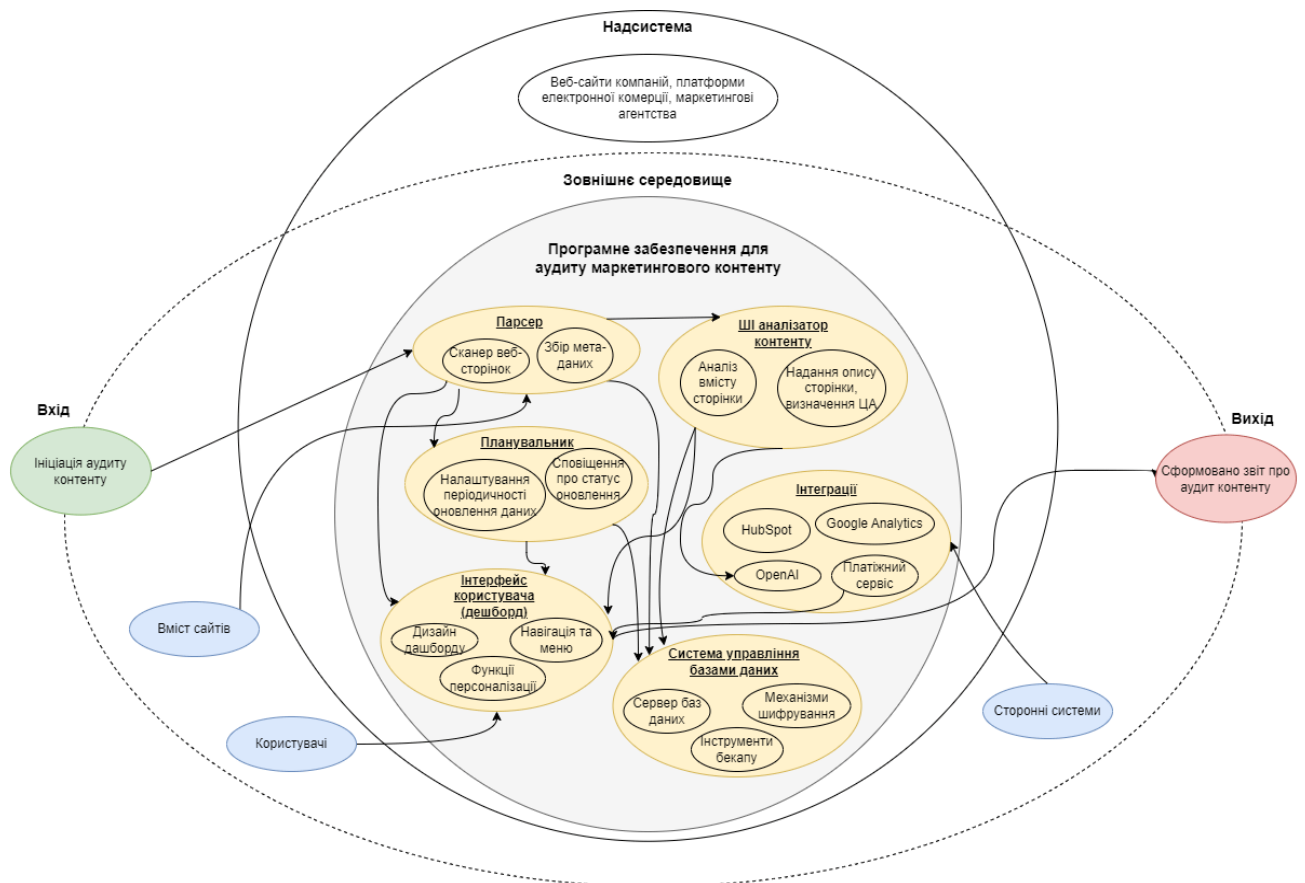


Рис. 2.1. Концептуальна модель проєкту

Отже, розглянемо детальніше основні компоненти, які включає концептуальна модель інформаційної системи аудиту маркетингового контенту:

- **Надсистема.** Програмне забезпечення функціонує у взаємодії з надсистемою — вебсайтами компаній, платформами електронної комерції, маркетинговими агентствами, які виступають основними джерелами контенту для аналізу.
- **Зовнішнє середовище.** Включає зміст сайтів (інформацію з вебресурсів, яку парсер системи збирає для подальшого аналізу), користувачів програмного забезпечення та сторонні системи (такі як Google Analytics, HubSpot і OpenAI, що забезпечують інтеграцію з додатковою функціональністю).
- **Вхідні дані.** Ініціація аудиту контенту — процес, який запускає систему для аналізу вмісту вебсайтів.

- Вихідні дані. Результатом роботи системи є сформований звіт про аудит контенту, який надає детальний аналіз ефективності контенту і рекомендації для поліпшення маркетингової стратегії.
- Основні компоненти програмного забезпечення. Концептуальна модель описує такі ключові складові системи:
 - Парсер вебсторінок — сканує вебсайти для збору контенту (тексти, зображення, мета-дані) і передає ці дані на обробку. Сканер вебсторінок — інструмент для збирання даних з вебсайтів. Збір мета-даних — процес збору ключових даних, необхідних для подальшого аналізу.
 - Планувальник — відповідає за періодичність оновлення даних і сповіщення користувачів про стан оновлення. Планувальник: налаштування періодичності оновлення даних визначає, як часто система повинна оновлювати дані. Сповіщення про статус оновлення — інформує користувачів про стан оновлення даних.
 - Штучний інтелект для аналізу контенту — використовує алгоритми ШІ для глибокого аналізу вмісту сторінок. Аналіз вмісту сторінки використовує штучний інтелект для обробки інформації на ресурсах. Надання опису сторінки та визначення ЦА генерує описи сторінок та визначає цільову аудиторію.
 - Інтеграції — включають взаємодію з Google Analytics, HubSpot, OpenAI та іншими сервісами для отримання маркетингових метрик і додаткових функцій, а також платіжний сервіс для управління підписками на сервіс.
 - Інтерфейс користувача (дешборд) — забезпечує взаємодію з користувачами через інтуїтивно зрозумілий інтерфейс. Функції персоналізації дозволяють налаштовувати дешборд під потреби користувачів.
 - Система управління базами даних — централізовано та безпечно зберігає всю інформацію. Сервер баз даних — центральне місце для

зберігання всіх даних. Інструменти бекапу забезпечують резервне копіювання даних.

Ця модель допомагає структуровано підходити до розробки та впровадження інформаційної системи.

Вона також демонструє логіку взаємодії між компонентами системи, наприклад:

- Парсер передає зібрані дані до бази даних і ШІ аналізатора.
- ШІ аналізатор генерує рекомендації та надсилає їх до інтерфейсу користувача.
- Інтерфейс користувача (дешборд) дозволяє відображати результати аналізу та забезпечувати користувачам доступ до налаштувань і персоналізації звітів.
- Планувальник забезпечує регулярне оновлення даних і надсилає сповіщення про статус процесу тощо.

Концептуальна модель наочно демонструє структуру системи, що дозволяє чітко відобразити її основні функціональні елементи та взаємозв'язки між ними.

Завдяки моделі стає можливим:

- Чітке визначення вимог до системи та її функціональних можливостей.
- Ефективна комунікація між усіма учасниками проєкту.
- Мінімізація ризиків на етапах розробки через контроль відповідності початковим цілям.

Це сприяє ефективному плануванню, реалізації та подальшому вдосконаленню проєкту.

Концептуальна модель системи забезпечує комплексний підхід до створення інформаційного продукту, що відповідає сучасним вимогам ефективності, безпеки та зручності для користувачів.

Таким чином, концептуальна модель є не лише технічним інструментом, але й стратегічним підґрунтям для досягнення успіху проєкту. Її розробка

дозволяє не тільки забезпечити ефективне функціонування майбутнього продукту, але й зробити його адаптивним до змінних потреб ринку та користувачів, що гарантує його конкурентоспроможність і довгострокову актуальність.

2.2 Формалізація математичних моделей для автоматизації процесів аудиту

Для розробки хмарного SaaS рішення для автоматизації контент-аудиту необхідно виконати детальну розробку концептуальних моделей інформаційної системи, формалізувати математичні моделі та поставити задачі в математичному вигляді.

Формалізація математичних моделей є важливим етапом у побудові програмного забезпечення для аудиту маркетингового контенту. Цей процес передбачає визначення алгоритмів, функцій та математичних залежностей, які ляжуть в основу автоматизації збору, обробки та аналізу даних. Основною метою моделювання є забезпечення точності, швидкодії та надійності роботи системи.

Формалізація математичних моделей дозволяє описати складні системи або процеси у вигляді абстрактних математичних структур. Завдяки цьому створюються алгоритми, які забезпечують ефективну реалізацію необхідних функцій, автоматизацію процесів та обґрунтованість рішень, що приймаються на основі аналізу даних.

Математична модель — це формалізоване уявлення про систему, процес або об'єкт у вигляді рівнянь, нерівностей, функцій чи логічних схем. Математична модель є спрощеним відображенням реального світу, яке дозволяє досліджувати поведінку системи за різних умов [1].

Формалізація — це процес перетворення задачі з реального світу в математичний опис, що включає визначення: вхідних даних (параметрів, змінних); взаємозв'язків між елементами системи (функції, залежності);

обмежень та критеріїв оптимальності (задачі лінійного програмування, прогнозування тощо).

Формалізація дозволяє перейти від окремих задач до більш загальних рішень, застосовуючи єдині алгоритми та підходи. За допомогою математичних моделей можливо також визначити оптимальні шляхи досягнення цілей, скорочуючи витрати часу та ресурсів. Крім того, використання формалізованих моделей дозволяє передбачити поведінку системи у майбутньому за умов зміни вхідних параметрів. На основі формалізованих моделей створюються алгоритми, які автоматизують рутинні процеси та забезпечують ефективну роботу систем.

Виділяють наступні етапи формалізації математичних моделей:

1. Аналіз системи. На початковому етапі проводиться детальний аналіз системи або процесу, який підлягає моделюванню. Це включає виявлення всіх основних елементів, змінних і параметрів, що можуть впливати на поведінку системи. Розглядаються внутрішні та зовнішні фактори, визначаються межі системи та її взаємодія з оточенням. Також аналізуються можливі впливи та залежності між компонентами системи. Цей етап є критичним, оскільки від повноти і точності аналізу залежить якість побудованої моделі.
2. Визначення цілей. Після детального аналізу системи необхідно чітко сформулювати мету моделювання. Це може бути оптимізація процесів, прогнозування поведінки системи, виявлення критичних точок або сценаріїв розвитку подій. Визначення цілей допомагає сфокусувати подальшу роботу на ключових аспектах і забезпечити, щоб модель відповідала практичним потребам. Крім того, встановлюються критерії успіху та очікувані результати, які будуть використовуватися для оцінки ефективності моделі.
3. Формування залежностей. На цьому етапі описуються взаємозв'язки між змінними та параметрами системи. Це здійснюється шляхом побудови математичних функцій, рівнянь або логічних умов, які точно

відображають реальні процеси. Можуть використовуватися різні математичні методи, включаючи лінійну та нелінійну алгебру, диференціальні рівняння, статистичні моделі тощо. Важливо забезпечити коректність цих залежностей, оскільки вони є основою для точних розрахунків і прогнозів.

4. Перевірка адекватності. Побудована модель підлягає ретельному тестуванню для перевірки її точності та відповідності реальним процесам. Це включає порівняння результатів моделювання з емпіричними даними, проведення експериментів або симуляцій. У випадку виявлення розбіжностей модель коригується та вдосконалюється. Перевірка адекватності дозволяє впевнитися, що модель здатна надійно відтворювати поведінку системи і може використовуватися для прийняття обґрунтованих рішень.

Отже, оскільки ми вже виявили основні елементи системи і відобразили їх у концептуальній моделі, можна перейти до етапу визначення цілей. Це дозволить нам чітко окреслити, які задачі ми прагнемо вирішити за допомогою моделі, та спрямувати наші зусилля на досягнення конкретних результатів. Встановлення ясних цілей також полегшить подальшу розробку та оптимізацію моделі, забезпечуючи її ефективність та практичну цінність.

Основні задачі можна сформулювати наступним чином:

- Задача парсингу вмісту сайту. Потрібно розробити ефективний механізм, який зможе автоматично визначати та збирати всі мета-атрибути, заголовки та основний контент для кожної сторінки сайту. Це включає аналіз структури HTML, обробку динамічного контенту та забезпечення коректного вилучення інформації незалежно від складності сторінки. Такий інструмент повинен бути здатним працювати з різними форматами даних і адаптуватися до змін у структурі сайту, забезпечуючи актуальність та повноту зібраних даних.
- Задача інтеграції з аналітичними інструментами. Необхідно налаштувати систему так, щоб вона могла автоматично отримувати та обробляти дані з

таких платформ, як Google Analytics та HubSpot. Це дозволить визначити значення результативності для кожної сторінки, включаючи показники трафіку, поведінку користувачів, конверсії та інші метрики. Інтеграція повинна бути надійною та безпечною, забезпечуючи постійний доступ до актуальних даних для подальшого аналізу та прийняття рішень.

- **Задача оновлення даних.** Важливо забезпечити регулярне оновлення зібраних даних відповідно до вимог користувача. Це включає налаштування періодичності оновлення, автоматичне виявлення змін на сайті та оновлення бази даних без втручання користувача. Система повинна бути гнучкою та дозволяти змінювати налаштування оновлення для різних сторінок або секцій сайту, забезпечуючи оптимальне використання ресурсів та актуальність інформації.
- **Задача визначення цільової аудиторії.** Використовуючи можливості штучного інтелекту та машинного навчання, необхідно автоматично визначати цільову аудиторію для кожної сторінки сайту. Це може включати аналіз демографічних даних, інтересів, поведінки користувачів та інших релевантних факторів. Результатом повинен бути короткий та інформативний опис цільової аудиторії, який можна використовувати для персоналізації контенту, маркетингових стратегій та покращення взаємодії з користувачами.
- **Задача прогнозування трафіку та сповіщення про падіння трафіку.** На основі історичних даних та аналізу зовнішніх факторів потрібно розробити модель, яка зможе прогнозувати майбутній трафік на сайт. Система повинна бути здатною виявляти тенденції, сезонні коливання та потенційні ризики зниження трафіку. У випадку виявлення значного падіння трафіку, повинні автоматично надсилатися сповіщення відповідальним особам. Це дозволить швидко реагувати на проблеми та вживати необхідних заходів для відновлення та покращення показників відвідуваності сайту.

2.2.1 Математична модель парсингу вмісту сайту

Одним із критичних етапів процесу аудиту контенту є парсинг вмісту вебсайту, який дозволяє перетворити неструктуровані дані в структуровану форму для подальшого аналізу.

Модель парсингу вмісту сайту включає послідовне виконання двох основних задач:

1. Парсинг сайтмапи для визначення списку сторінок, які необхідно обробити на наступному етапі.
2. Посторінковий парсинг сайту, що передбачає витягування конкретних даних з кожної сторінки.

Детальний розгляд та оптимізація цих задач є важливими для підвищення ефективності процесу збору даних.

Сайтмапа (англ. *sitemap*) — це спеціалізований XML-файл або набір файлів, які містять структурований перелік всіх доступних URL-адрес вебсайту. Вона слугує навігаційним інструментом як для пошукових систем, так і для програмних агентів, що здійснюють парсинг контенту. Сайтмапа спрощує процес виявлення та доступу до всіх сторінок сайту, особливо для великих вебресурсів з багаторівневою структурою.

Математична модель процесу парсингу сайтмапи може бути представлена наступним чином:

$$S = \bigcup_{i=1}^n p(S_i), \text{ де } S_i = P(S_0, i) \quad (2.1)$$

де S — набір всіх сторінок сайту;

n — кількість сайтмап сайту (визначається на етапі парсингу індекс-сайтмапи);

i — порядковий номер сайтмапи;

p — функція витягування та зберігання URL-адрес сторінок з окремої сайтмапи;

S_i — окрема сайтмапа;

P — функція витягування та зберігання URL-адрес сайтмап;

S_0 — індекс-сайтмапа, що містить посилання на всі сайтмапи сайту.

При аналізі даної моделі можна виявити, що після отримання індекс-сайтмапи S_0 та визначення кількості сайтмап n , парсинг кожної сайтмапи S_i може виконуватися незалежно. Це відкриває можливість для паралельного виконання цих задач, що значно прискорює процес збору даних.

Переваги паралельного парсингу сайтмап:

- Підвищення швидкості обробки — одночасна обробка декількох сайтмап зменшує загальний час парсингу.
- Ефективне використання ресурсів — розподіл навантаження на кілька процесорів або серверів.
- Масштабованість — можливість обробки великих вебсайтів з великою кількістю сторінок.

Однак паралельний парсинг можливий лише після повного парсингу індекс-сайтмапи S_0 , оскільки саме вона містить інформацію про структуру та кількість наявних сайтмап.

Посторінковий парсинг передбачає обробку кожної сторінки сайту з метою витягування необхідної інформації, такої як заголовки, мета атрибути та основний контент. Це критичний етап, оскільки саме тут відбувається перетворення неструктурованих даних у структуровану форму, придатну для аналізу.

Модель посторінкового парсингу може бути описана наступним чином:

$$P = \sum_{i=1}^n f(R_i), \text{ де } f(R_i) = \{h(R_i), m(R_i), t(R_i)\} \quad (2.2)$$

де P — результат роботи функції парсингу сайту, дані по кожній сторінці;

i — порядковий номер сторінки сайту;

n — кількість сторінок сайту;

$f(R_i)$ — функція парсингу «сирого» контенту сторінки i (HTML-коду);

$h(R_i)$ — функція витягування та зберігання заголовків сторінки i ;

$m(R_i)$ — функція витягування та зберігання мета атрибутів сторінки i ;

$t(R_i)$ — функція витягування і зберігання основного контенту сторінки i .

Аналіз моделі показує, що парсинг кожної сторінки R_i є незалежною задачею. Це означає, що можна застосувати паралельні обчислення для обробки сторінок, що дозволяє ще більше прискорити процес збору даних.

Це відкриває наступні переваги:

- Скорочення часу обробки завдяки одночасній обробці декількох сторінок.
- Підвищення ефективності через краще використання доступних обчислювальних ресурсів.
- Масштабованість через можливість обробки великої кількості сторінок без зниження продуктивності.

Розглянемо функції парсингу детальніше:

- Функція витягування заголовків $h(R_i)$ відповідає за витягування всіх заголовків сторінки R_i , таких як теги $\langle h1 \rangle$, $\langle h2 \rangle$, ..., $\langle h6 \rangle$. Заголовки зазвичай містять ключову інформацію про зміст сторінки та її структуру.
- Функція витягування мета атрибутів $m(R_i)$ збирає мета атрибути, такі як теги $\langle meta name="description" \rangle$ та $\langle meta name="keywords" \rangle$. Вони надають додаткову інформацію про сторінку, яка може бути корисною для SEO-аналізу та індексації пошуковими системами.
- Функція витягування контенту $t(R_i)$ витягує основний текстовий контент сторінки, видаляючи HTML-теги та інші непотрібні елементи. Отриманий текст може бути використаний для аналізу вмісту, машинного навчання та інших цілей.

Таким чином, розроблена модель парсингу вмісту сайту, яка включає послідовний парсинг сайтмапи та посторінковий парсинг, є ефективним інструментом для збору та обробки вебданих. Можливість паралельного виконання задач на обох етапах процесу дозволяє значно підвищити продуктивність та масштабованість системи. Детальний аналіз та розуміння

кожного компоненту моделі є критичним для її успішної реалізації та застосування у практичних задачах.

2.2.2 Математична модель інтеграції з аналітичними інструментами

У сучасному цифровому середовищі аналіз поведінки користувачів на вебсайтах є критично важливим для оптимізації контенту та підвищення ефективності маркетингових стратегій. Інтеграція даних з Google Analytics та HubSpot дозволяє отримати більш повну картину взаємодії користувачів з сайтом. Для цього необхідно розробити математичну модель, яка об'єднує метрики з обох платформ для кожної сторінки.

Нехай $P = \{p_1, p_2, \dots, p_n\}$ — множина сторінок вебсайту. Для кожної сторінки p_i визначимо набір метрик з Google Analytics та HubSpot.

Метрики з Google Analytics:

- $GAviews(p_i)$ — кількість переглядів сторінки.
- $GAusers(p_i)$ — кількість унікальних користувачів, що відвідали сторінку.
- $GAeng(p_i)$ — показник частки взаємодій для сторінки.
- $GAtime(p_i)$ — середній час перебування на сторінці.

Метрики з HubSpot:

- $HSleads(p_i)$ — кількість згенерованих на сторінці лідів (leads). Це потенційні клієнти, які виявили зацікавленість у продукті або послугі. Ліди можуть бути отримані через різні канали, такі як заповнення форми на вебсайті, підписка на розсилку або участь у заходах. Це найширший етап, де клієнт ще не кваліфікований для подальшого продажу.
- $HSmqls(p_i)$ — кількість згенерованих на сторінці MQL (Marketing Qualified Lead) — це ліди, які демонструють вищий рівень зацікавленості, підтверджений взаємодією з маркетинговими матеріалами. Наприклад, вони завантажили матеріали, регулярно відкривають електронні листи чи

переглядають специфічні сторінки на вебсайті. MQL є готовими до подальшої обробки, але ще не готові до прямого продажу.

- $HSsqls(p_i)$ — кількість згенерованих на сторінці SQL (Sales Qualified Lead) — це MQL, який пройшов додаткову кваліфікацію та оцінений як готовий до взаємодії з відділом продажів. Це клієнти, які висловили конкретний інтерес до придбання продукту чи послуги, наприклад, через запит комерційної пропозиції або консультації [16].
- $HSconversion(p_i)$ — коефіцієнт конверсії для сторінки.

Для інтеграції цих метрик пропонується наступна математична модель:

$$M(p_i) = \sum_{k=1}^K w_k * m_k(p_i) \quad (2.3)$$

де $M(p_i)$ — інтегрований показник ефективності сторінки p_i ;

$m_k(p_i)$ — нормалізована метрика k-го типу для сторінки p_i ;

w_k — ваговий коефіцієнт для метрики m_k .

Розглянемо математичну модель інтеграції з аналітичними інструментами детальніше:

1. Нормалізація метрик. Для забезпечення коректності сумування різнорідних метрик необхідно їх нормалізувати до діапазону $[0,1]$.
2. Визначення вагових коефіцієнтів. Вагові коефіцієнти w_k встановлюються на основі важливості кожної метрики для цілей аналізу. Наприклад, якщо конверсія є пріоритетом, то $w_{conversion}$ (коефіцієнт для метрики конверсії) буде мати більше значення.
3. Інтегрований показник. Підсумовуючи зважені нормалізовані метрики, отримуємо єдиний показник ефективності для кожної сторінки, що дозволяє ранжувати та порівнювати сторінки між собою.

Таким чином, запропонована математична модель забезпечує системний підхід до інтеграції метрик з Google Analytics та HubSpot, дозволяючи отримати узагальнений показник ефективності контенту на рівні окремих сторінок. Це сприяє прийняттю обґрунтованих рішень щодо оптимізації сайту та підвищення

взаємодії з користувачами. Композиція моделі дасть змогу відносно просто розширити набір метрик на етапі реалізації, а отже є масштабованою.

2.2.3 Математична модель оновлення даних сайту

Періодичне оновлення даних вебсторінок є ключовим аспектом забезпечення актуальності та достовірності інформації для кінцевих користувачів. Для формалізації цього процесу пропонується математична модель, яка включає три основні компоненти: абстрактну задачу парсингу сторінки, абстрактну задачу оновлення даних з аналітичних інструментів та задачу злиття існуючих даних з новими, з урахуванням користувацьких налаштувань періодичності.

Математична модель періодичного оновлення даних сторінок сайту відповідно до користувацьких налаштувань періодичності може бути формалізована наступним чином.

Нехай U — множина сайтів, і для кожного сайту $u \in U$ задано інтервал оновлення Δt_u .

Тоді часи оновлення для сайту u визначаються як:

$$T_u = \{t_i^u \mid t_i^u = t_0^u + i\Delta t_u\} \quad (2.4)$$

де T_u — множина часів оновлення сайту u ;

t_i^u — час оновлення сайту u в порядку i ;

t_0^u — час налаштування періодичності оновлення;

i — порядковий номер оновлення;

Δt_u — часовий інтервал оновлення, заданий користувачем.

На кожному часі $t_i^u \in T_u$ виконується:

1. Результати виконання парсингу сайту $P_u(t_i^u)$ — абстрактна задача в цьому контексті, математична модель якої описана в пункті 2.2.1.

2. Результати виконання оновлення даних з аналітичних інструментів $A_u(t_i^u)$ — абстрактна задача в цьому контексті, математична модель якої описана в пункті 2.2.2.
3. Оновлення загального набору даних шляхом злиття з існуючими даними:

$$D_{total}^u(t_i^u) = M(D_{total}^u(t_{i-1}^u), D_{new}^u(t_i^u)) \quad (2.5)$$

де $D_{total}^u(t_i^u)$ — вся сукупність даних по сайту в момент оновлення;

M — операція злиття, що враховує специфіку даних та можливі конфлікти;

$D_{total}^u(t_{i-1}^u)$ — вся сукупність даних по сайту в момент попереднього оновлення;

$D_{new}^u(t_i^u)$ — сукупність даних, які продукуються поточним оновленням.

Таким чином, загальна математична модель періодичного оновлення даних для SaaS продукту з урахуванням масштабованості виглядає наступним чином:

$$D_{total}^u(t) = M(D_{total}^u(t - \Delta t_u), P_u(t) \cup A_u(t)) \quad (2.6)$$

де $D_{total}^u(t)$ — вся сукупність даних по сайту в момент оновлення;

M — операція злиття, що враховує специфіку даних та можливі конфлікти;

$D_{total}^u(t - \Delta t_u)$ — вся сукупність даних по сайту в момент попереднього оновлення;

$P_u(t)$ — результати виконання парсингу сайту;

$A_u(t)$ — результати виконання оновлення даних з аналітичних інструментів.

Запропонована модель відображає взаємодію між основними компонентами процесу оновлення:

- Парсинг сторінки забезпечує витягування актуальної інформації безпосередньо з вебсторінки.
- Оновлення з аналітичних інструментів надає додаткову інформацію та метрики, які можуть бути корисними для глибшого аналізу.
- Злиття даних інтегрує нові дані з існуючими, забезпечуючи цілісність та актуальність інформаційної бази.

Ця модель відображає процес періодичного оновлення даних для кожного користувача відповідно до його індивідуальних налаштувань періодичності. Вона забезпечує масштабованість системи, оскільки дозволяє обробляти множину сайтів U незалежно та паралельно по кожні сторінці, ефективно розподіляючи ресурси в контексті SaaS продукту.

2.2.4 Математична модель визначення цільової аудиторії сторінки

Актуальним завданням у складі проблеми аудиту контенту є автоматизоване визначення цільової аудиторії вебсторінок на основі їх контенту. Це має важливе значення для персоналізації контенту, таргетування реклами та покращення користувацького досвіду. Використання великих лінгвістичних моделей (ВЛМ), таких як GPT-4, відкриває нові можливості для глибинного аналізу текстових даних та семантичного розуміння контенту.

$$TA_i = h(C_i, A_i) \quad (2.7)$$

де TA_i — цільова аудиторія для i -ої сторінки;

C_i — контент сторінки;

A_i — аналіз сторінки за допомогою ШІ.

Ця модель дозволяє визначити цільову аудиторію для конкретного контенту за допомогою ШІ аналізу. Функція h враховує різні аспекти контенту, такі як ключові слова, стиль написання та інші фактори, для визначення найвідповіднішої аудиторії.

Робота функції h базується на наступних кроках:

1. Семантичне представлення: Використання ВЛМ для отримання векторних представлень контенту вебсторінок, що враховують контекстуальні та семантичні зв'язки.
2. Обчислення схожості: Використання косинусної міри (міри подібності між двома векторами) для оцінки схожості між сторінками, що дозволяє виявити семантично близькі документи.
3. Кластеризація: Застосування алгоритмів кластеризації (наприклад, k -середніх) для групування сторінок за схожістю їх векторних представлень.
4. Інтерпретація кластерів: Аналіз отриманих кластерів для визначення характеристик цільових аудиторій, які вони представляють.

Запропонована математична модель дозволяє автоматизувати процес визначення цільової аудиторії вебсторінок на основі їх контенту. Використання великих лінгвістичних моделей забезпечує глибоке семантичне розуміння тексту, що підвищує точність кластеризації та релевантність визначених аудиторій. Для спрощення імплементації і підтримки продукту рекомендується інтеграція з готовим рішенням, яке реалізує доступ до готової великої лінгвістичної моделі, за умови дотримання конфіденційності даних.

2.2.5 Математична модель прогнозування трафіку

Прогнозування трафіку вебсторінок є важливим завданням у сфері аналізу контенту, яке дозволяє оптимізувати маркетингові стратегії, покращувати користувацький досвід та ефективно розподіляти ресурси. Для цього використовуються різноманітні математичні моделі, які базуються на аналізі історичних даних. У даній роботі пропонується регресійна предиктивна модель, яка використовує історичні дані трафіку для прогнозування майбутніх відвідувань сторінок сайту.

Нехай T_t — кількість відвідувань сторінки сайту у час t . Для прогнозування T_{t+1} використовуються історичні дані, а також інші пояснювальні змінні, які можуть впливати на трафік.

Розглянемо нелінійну регресійну модель, яка описується наступною формулою:

$$T_{t+1} = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t}^2 + \dots + \beta_n x_{n,t}^n + \varepsilon_t \quad (2.8)$$

де T_{t+1} — прогнозована кількість відвідувань сторінки у час $t+1$;

β_0 — вільний член моделі;

$\beta_1, \beta_2, \dots, \beta_n$ — коефіцієнти регресії, що характеризують вплив відповідних незалежних змінних;

$x_{1,t}, x_{2,t}^2, \dots, x_{n,t}^n$ — незалежні змінні, які впливають на трафік у час t ;

ε_t — випадкова похибка моделі, що характеризує невраховані фактори.

До незалежних змінних $x_{i,t}$ можуть бути включені різноманітні фактори, які впливають на трафік сторінки:

1. Історичні дані трафіку: попередні значення $T_t, T_{t-1}, \dots, T_{t-k}$.
2. Мета атрибуту сторінок: кількість слів, медіа-елементів, тематика статті.
3. Часові показники: день тижня, місяць, сезонність.
4. Метрики аналітичних систем.
5. Технічні фактори: швидкість завантаження сторінки, мобільна оптимізація.

Для забезпечення якості сервісу необхідно забезпечити процес валідації моделі і оцінки параметрів.

Оцінка параметрів β_n здійснюється методом найменших квадратів, який мінімізує суму квадратів відхилень між спостережуваними та прогнозованими значеннями трафіку.

Для валідації регресійної моделі використовуються наступні метрики:

- Коефіцієнт детермінації R^2 : показує частку дисперсії залежної змінної, яку пояснює модель.
- Середня квадратична помилка (MSE): середнє значення квадратів відхилень прогнозів від фактичних значень.
- Середня абсолютна помилка (MAE): середнє абсолютне відхилення прогнозів від фактичних значень.

Запропонована нелінійна регресійна модель дозволяє ефективно прогнозувати трафік вебсторінок на основі історичних даних та інших релевантних факторів. Використання цієї моделі в рамках сервісу аналізу контенту сприяє покращенню стратегічного планування та оптимізації контенту для підвищення відвідуваності сайту.

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

3.1 Розробка концептуальної моделі бази даних проєкту

Модель даних (data model) описує сутності, класи або об'єкти даних, що мають відношення до предметної області, атрибути, які використовуються для їх опису, та зв'язки між ними, щоб забезпечити загальний набір семантики для аналізу та реалізації [13]. Цей процес є основою для створення бази даних, яка ефективно підтримуватиме цілісність та оптимальність зберігання й обробки необхідних даних.

Моделювання даних (data modelling) є важливим етапом у життєвому циклі інформаційної системи. Згідно з ВАВОК v3 (розділ 10.15), моделювання даних забезпечує основу для системного аналізу та проєктування рішень. Це одна з ключових практик бізнес-аналізу, яка дозволяє перетворювати вимоги бізнесу у структуровану форму, зрозумілу для технічних фахівців.

Модель даних зазвичай має форму діаграми, яка супроводжується текстовим описом. Вона візуально представляє елементи, важливі для бізнесу (наприклад, люди, місця, речі та бізнес-транзакції), атрибути, пов'язані з цими елементами, та важливі взаємозв'язки між ними.

Модель даних часто використовуються для з'ясування та аналізу вимог і проєктування, а також для підтримки впровадження та безперервного вдосконалення.

Існують три основних рівні моделей даних, кожен з яких відповідає певному етапу розробки:

- Концептуальна модель даних (Conceptual data model) створюється на ранніх етапах розробки, щоб відобразити, як бізнес розуміє свою інформацію. Вона не залежить від будь-якого рішення або технології і може бути використана для представлення того, як бізнес сприймає свою інформацію. Вона може бути використана для створення узгодженого

словника, що описує бізнес-інформацію та взаємозв'язки всередині цієї інформації.

- Логічна модель даних (Logical data model) — абстракція концептуальної моделі даних, яка включає правила нормалізації для формального управління цілісністю даних і зв'язків. Вона уточнює атрибути, типи даних та зв'язки, необхідні для реалізації. Вона пов'язана з проектуванням рішення.
- Фізична модель даних (Physical data model) — використовується експертами предметної області для опису того, як фізично організована база даних. Вона вирішує такі проблеми, як продуктивність, паралельність і безпека [34]. Це найбільш деталізована модель, що визначає конкретну реалізацію даних на фізичному рівні. Вона враховує продуктивність, паралельність і безпеку.

Концептуальна модель бази даних (рис. 3.1) є основою для подальшої розробки інформаційної системи. Це інформаційна модель найвищого рівня абстракції. Вона відображає основні об'єкти, їх атрибути, а також взаємозв'язки між ними [5].

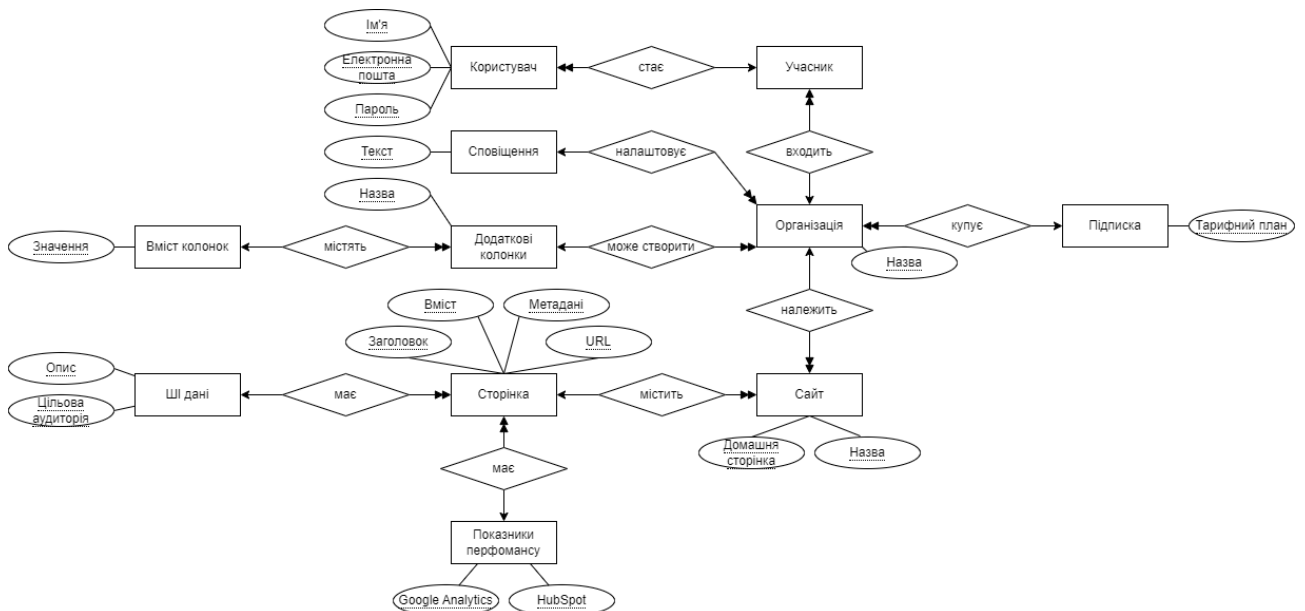


Рис. 3.1. Концептуальна модель бази даних

Концептуальна модель бази даних відіграє ключову роль у забезпеченні узгодженості між бізнес-вимогами та технічним рішенням. Ця модель дозволяє:

- Визначити основні сутності, які будуть використовуватися в системі.
- З'ясувати взаємозв'язки між сутностями.
- Сформуванати базу для подальшого розроблення логічної та фізичної моделей бази даних.

Модель БД для проєкту розробки програмного забезпечення для аудиту та аналізу маркетингового контенту включає такі ключові елементи:

- Користувач (User) — містить атрибути Ім'я, Електронна пошта, Пароль, які необхідні для ідентифікації та автентифікації користувача в системі. Тобто це базовий елемент системи, який містить інформацію про кожного користувача, що взаємодіє з програмним забезпеченням.
- Учасник (Member) — являє собою користувача, який став членом певної організації (наприклад, створив її або приєднався до існуючої).
- Організація (Organization) — сутність, що відображає компанії або групи, які користуються програмним забезпеченням і включає атрибут Назва і має зв'язки з іншими сутностями, такими як Учасник, Сайт та Підписка.
- Сайт (Site) — відображає вебсайт організації та містить атрибути Назва та Домашня сторінка. Сайт може містити безліч сторінок.
- Сторінка (Page) — являє собою вебсторінку, що належить сайту, і включає атрибути Заголовок, URL, Вміст та метадані (Метазаголовок, Метаопис, Метаключові слова).
- Показники перформансу (Performance Data) — включає дані з аналітичних інструментів, таких як Google Analytics та HubSpot, що відображають продуктивність сторінок.
- Додаткові колонки (Custom Columns) та їх вміст (Custom Column Data) — дозволяють зберігати додаткову інформацію, яку може створити користувач для своїх сторінок.
- Сповіщення (Notification) — повідомлення, що надсилаються користувачам про важливі події або зміни в системі.

- Підписка (Subscription) — містить інформацію про підписки організацій, зокрема Тарифний план.

Розроблена концептуальна модель відображає взаємозв'язки між різними сутностями, що є основою для подальшої розробки логічної моделі бази даних.

Таким чином, концептуальна модель є важливим інструментом для визначення основних сутностей, їх атрибутів і взаємозв'язків у системі, що забезпечує чітке розуміння структури даних на початковому етапі розробки.

У межах проекту створення програмного забезпечення для аудиту та аналізу маркетингового контенту концептуальна модель слугує базисом для формування єдиного уявлення про об'єкти системи та їх взаємодію. Вона дозволяє візуалізувати бізнес-вимоги, уточнити структуру інформації та забезпечує основу для подальшої деталізації в логічній моделі.

3.2 Побудова логічної моделі бази даних проекту

Логічна модель бази даних є наступним етапом після розробки концептуальної моделі і включає більш детальний опис структури даних і зв'язків між ними. Логічне (дatalogічне) проектування передбачає створення схеми бази даних на основі конкретної моделі даних, наприклад, реляційної моделі даних. Отже, створення логічної моделі бази даних проекту включає наступні кроки:

1. Вибір моделі даних. Найчастіше вибирається реляційна модель даних у зв'язку з наочністю табличного подання даних та зручності роботи з ними. Для реляційної моделі даних дatalogічна модель — це набір схем відносин, зазвичай із зазначенням первинних ключів, а також «зв'язків» між відносинами, що представляють собою зовнішні ключі.
2. Визначення набору таблиць, виходячи з ER-моделі (Entity–relationship model) та їх документування. Для кожної сутності ER-моделі створюється таблиця. Ім'я сутності — ім'я таблиці. Встановлюються зв'язки між

таблицями за допомогою механізму первинних та зовнішніх ключів. Структури таблиць та встановлені зв'язки між ними документуються.

3. Нормалізація таблиць. Для правильного виконання нормалізації проєктувальник повинен глибоко вивчити семантику та особливості використання даних. На цьому кроці він перевіряє коректність структури таблиць, створених на попередньому кроці через застосування до них процедури нормалізації. Вона полягає у приведенні кожної з таблиць принаймні до 3-ї нормальної форми (3НФ). В результаті нормалізації виходить дуже гнучкий проєкт бази даних, що дозволяє легко вносити до неї необхідні розширення.
4. Перевірка логічної моделі даних щодо можливості виконання всіх транзакцій, передбачених користувачами. Транзакція — це набір дій, що виконуються окремим користувачем або прикладною програмою з метою зміни вмісту бази даних.

Перелік транзакцій визначається діями користувачів у предметній області. Використовуючи ER-модель, словник даних та встановлені зв'язки між первинними та зовнішніми ключами, робиться спроба виконати всі необхідні операції доступу до даних вручну. Якщо якусь операцію виконати вручну не вдається, складена логічна модель даних є неадекватною і містить помилки, які треба усунути. Можливо, вони пов'язані з пропуском у моделі сутності, зв'язку або атрибуту.

5. Визначення вимог підтримки цілісності даних та їх документування. Ці вимоги є обмеженнями, які запроваджуються, щоб запобігти розміщенню в базі даних суперечливих даних. На цьому етапі питання цілісності даних висвітлюються безвідносно до конкретних аспектів її реалізації. Повинні бути розглянуті такі типи обмежень:
 - обов'язкові дані. З'ясовується, чи є атрибути, які не можуть мати значення NULL;
 - обмеження для значень атрибутів. Визначаються допустимі значення для атрибутів;

- цілісність сутностей. Вона досягається, якщо первинний ключ сутності не містить значення NULL;
- посилальна цілісність. Вона розуміється так, що значення зовнішнього ключа має обов'язково бути присутнім у первинному ключі одного з рядків таблиці для батьківської сутності;
- обмеження, що накладаються бізнес-правилами.

6. Створення остаточного варіанта логічної моделі даних та обговорення його з користувачами. На цьому кроці готується остаточний варіант ER-моделі, що представляє логічну модель даних [20].

Таким чином, на основі концептуальної моделі була побудована логічна модель (рис. 3.2), яка відображає всі сутності, їх атрибути та зв'язки.

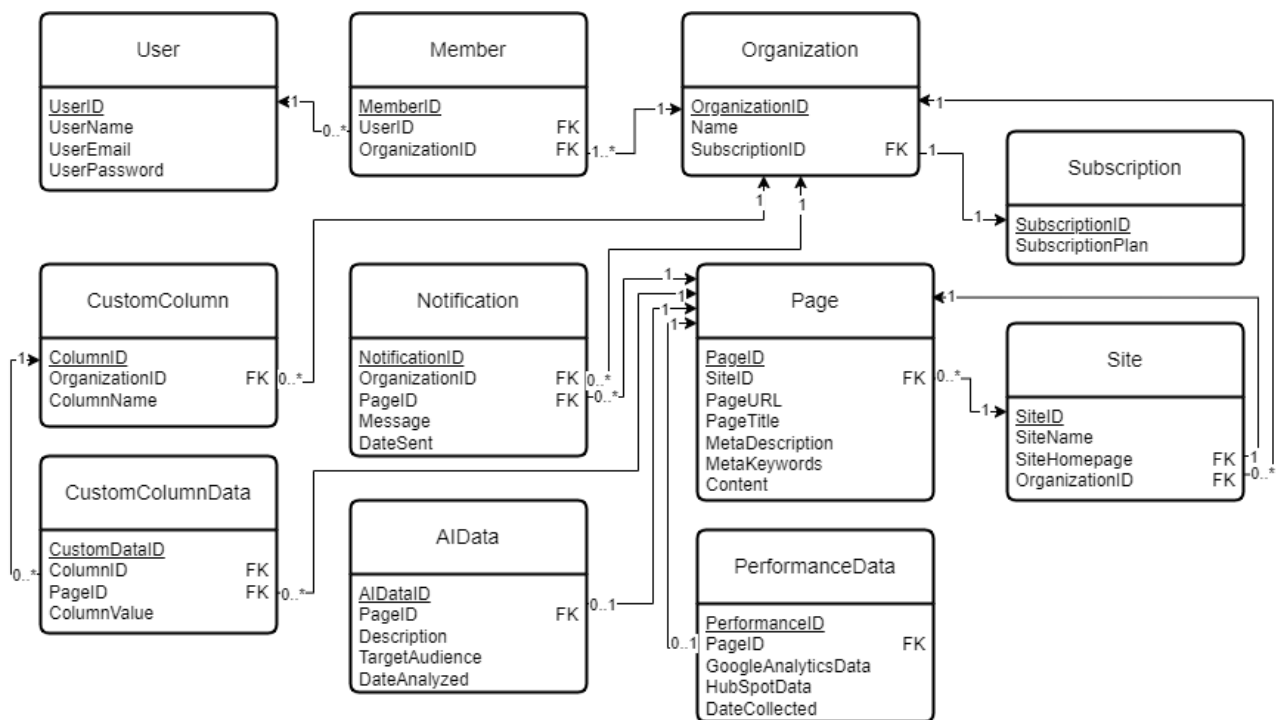


Рис. 3.2. Логічна модель бази даних

Основні елементи логічної моделі БД:

- User (Користувач);
- Member (Учасник);
- Organization (Організація);

- Site (Сайт);
- Page (Сторінка);
- PerformanceData (Показники результативності контенту);
- AIData (Дані, згенеровані штучним інтелектом);
- CustomColumn (Додаткова колонка);
- CustomColumnData (Вміст додаткових колонок);
- Notification (Сповідження);
- Subscription (Підписка).

Логічна модель бази даних забезпечує більш детальний і структурований підхід до зберігання і обробки даних, що є основою для розробки програмного забезпечення для аудиту та аналізу маркетингового контенту. Ця модель також включає всі необхідні атрибути і зв'язки між сутностями, що забезпечує цілісність та взаємозв'язок даних у системі. Вона також є основою для подальшої фізичної реалізації бази даних та забезпечення роботи програмного забезпечення для аудиту та аналізу маркетингового контенту.

3.3 Опис структури програмного забезпечення проєкту

Програмне забезпечення, розроблене в рамках цього проєкту, складається з кількох основних модулів, кожен з яких виконує певну функцію та має свою важливість у загальній функціональній структурі системи (рис. 3.3).



Рис. 3.3. Функціональна структура ПЗ

Нижче наведено детальний опис кожного з модулів та їх роль у загальній архітектурі програмного забезпечення.

- Парсер контенту. Парсер є критичним компонентом, що відповідає за автоматичне збирання даних з різних джерел. Його основні функції включають:
 - Автоматичний парсинг вмісту сайту — це дозволяє зібрати всі необхідні дані без втручання користувача.
 - Підтримка різних форматів та типів контенту — модуль підтримує HTML-сторінки, PDF-файли, зображення тощо, що робить його універсальним інструментом для збору даних.
 - Фільтрація нерелевантних даних — дозволяє уникнути збору зайвої інформації, концентруючись на потрібному контенті.
- Інтеграції з Google Analytics та HubSpot. Цей модуль дозволяє автоматично отримувати та аналізувати дані з популярних аналітичних платформ. Важливі функції включають:
 - Збір даних із аналітичних платформ — дозволяє отримувати метрики, які допомагають оцінювати ефективність контенту.
 - Аналіз зібраних даних — автоматично генерує звіти та рекомендації для оптимізації контенту.
 - Налаштування сповіщень — користувачі можуть отримувати сповіщення про падіння трафіку та інших важливих подій, що дозволяє швидко реагувати на зміни в ефективності контенту.
- Інтеграція зі штучним інтелектом. ШІ аналізатор використовує технології штучного інтелекту для розпізнавання та аналізу контенту. Це включає:
 - Розпізнавання та аналіз контенту — штучний інтелект допомагає автоматично визначати зміст та цільову аудиторію кожного матеріалу.
 - Надання короткого опису кожної сторінки — це допомагає користувачам швидко отримати загальне уявлення про зміст сторінок.

- Редагування та корекція результатів аналізу — користувачі можуть вносити зміни до результатів аналізу, що забезпечує гнучкість та точність даних.
- Дешборд (інтерфейс користувача). Дешборд є основним інтерфейсом взаємодії користувача з програмним забезпеченням. Важливість цього модуля полягає в тому, що саме через нього користувач отримує доступ до всіх функцій та можливостей системи. Дешборд забезпечує:
 - Зручний та інтуїтивно зрозумілий інтерфейс — це дозволяє користувачам легко шукати та працювати зі всіма функціями програми, що знижує поріг входу для нових користувачів і покращує загальний користувацький досвід.
 - Перегляд загальної статистики по контенту — користувачі можуть швидко отримувати інформацію про стан контенту, його популярність, ефективність тощо.
 - Фільтрація та сортування даних — ця функція дозволяє користувачам проводити детальний аналіз зібраних даних, що сприяє прийняттю обґрунтованих рішень.
- Планувальник. Планувальник забезпечує автоматизацію оновлення даних та сповіщення користувачів про статус процесів. Його функції включають:
 - Налаштування періодичності оновлення даних — це дозволяє користувачам налаштувати частоту оновлень відповідно до їхніх потреб.
 - Автоматичне сповіщення — користувачі отримують сповіщення про успішне або невдале оновлення даних, що підвищує прозорість процесу.
- Бази даних. Модуль баз даних відповідає за збереження та організацію всіх зібраних даних. Важливі аспекти включають безпеку, цілісність та доступність даних. Це забезпечує захист даних від несанкціонованого

доступу, втрати або пошкодження, а також їх доступність для користувачів у будь-який час.

- Платіжний модуль для оплати підписок користувачів. Цей модуль відповідає за управління підписками на сервіс. Його основні функції включають автоматизацію процесу оформлення підписок, дозволяючи користувачам легко оформлювати та оплачувати доступ до сервісу. Також модуль збирає звітність по транзакціям, забезпечуючи користувачів можливістю дізнатися інформацію про статус їх платежів.

Усі модулі працюють як єдина система, забезпечуючи ефективність та зручність використання. Інтеграція між компонентами забезпечує швидкий обмін даними, узгодженість результатів та підтримку сучасних технологій автоматизації.

Таким чином, програмне забезпечення пропонує комплексний підхід до автоматизації збору, аналізу та управління контентом, підвищуючи ефективність роботи користувачів і забезпечуючи їх сучасними інструментами для аудиту та оптимізації контенту.

3.4 Розробка алгоритмів та інтерфейсів програмного забезпечення проєкту

Одним з найпоширеніших способів візуалізації алгоритмів є блок-схема. Вона є одним з інструментів в розділі «Візуальні дані та інформація (4.6.6 Visual Data And Information)» в РМВОК 7, та визначається там наступним чином: «Ця діаграма зображує входи, дії процесу та виходи одного або декількох процесів у системі» [17].

Блок-схема (Flowchart) — це тип діаграми, яка представляє робочий потік або процес. Блок-схему також можна визначити як схематичне зображення алгоритму, покрокового підходу до вирішення завдання.

На блок-схемі кроки зображуються у вигляді блоків різного виду, а їх порядок — у вигляді стрілок, що з'єднують блоки між собою. Таке схематичне зображення ілюструє модель розв'язання певної проблеми. Блок-схеми

використовуються для аналізу, проєктування, документування або управління процесом чи програмою в різних галузях [12].

Блок-схема — це спосіб представлення алгоритму в графічній формі, у вигляді геометричних фігур, сполучених між собою лініями (стрілками). Форма блока визначає тип дії, а текст всередині блоку дає детальне пояснення конкретної дії. Стрілки на лініях, що сполучають блоки схеми, вказують послідовність виконання команд, передбачених алгоритмом. Блок-схеми, за рахунок наочності спрощують створення ефективних алгоритмів, розуміння роботи вже створених, а як наслідок і їх оптимізацію. Існуючі стандарти на типи блоків дозволяють легко адаптувати алгоритми, створені у вигляді блок-схем до будь-яких існуючих на сьогоднішній день мов програмування [19].

Як правило, блок-схеми рухаються зверху вниз і зліва направо.

Лінія потоку — стрілка (Flowline — Arrowhead) — показує порядок роботи процесу. Лінія, що виходить від одного символу і вказує на інший. Стрілки додаються, якщо потік не є стандартним зверху вниз, зліва направо.

Термінал (Terminal) — вказує на початок і кінець програми або підпроцесу. Представлений у вигляді овалу або закругленого прямокутника. Зазвичай вони містять слово «Start» або «End», або іншу фразу, що сигналізує про початок або кінець процесу.

Процес (Process) — представляє набір операцій, які змінюють значення, форму або розташування даних. Зображується у вигляді прямокутника.

Рішення (Decision) — показує умовну операцію, яка визначає, яким з двох шляхів піде програма. Операція зазвичай являє собою питання «yes/no» або тест «true/false». Зображується у вигляді ромба.

Введення/виведення (Input/Output) — вказує на процес введення та виведення даних. Зображується у вигляді скошеного прямокутника [19].

Отже, для забезпечення функціонування програмного забезпечення для аудиту та аналізу маркетингового контенту, були розроблені декілька ключових алгоритмів, що описують процеси збору, обробки та аналізу даних. Ці алгоритми реалізовані у вигляді блок-схем, що забезпечують наочне уявлення

про послідовність виконання кроків у кожному з процесів. Нижче наведено детальний опис алгоритмів, які є основою функціонування системи.

Розглянемо перші три алгоритми (рис. 3.4): реєстрації, створення організації, додавання сайту.

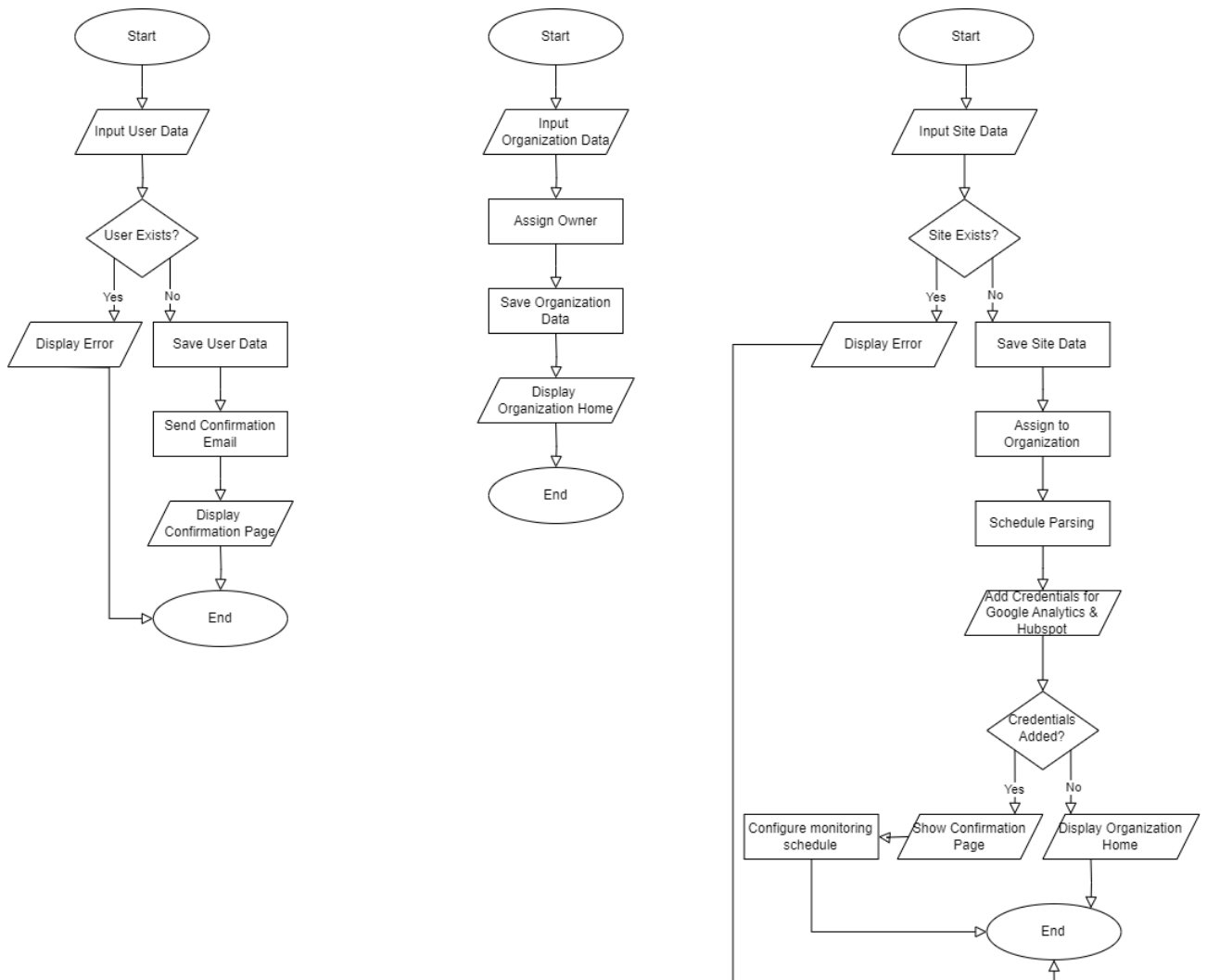


Рис. 3.4. Алгоритми реєстрації, створення організації, додавання сайту

Алгоритм реєстрації користувача забезпечує безпечний і структурований процес реєстрації нових користувачів у системі. Він починається з введення даних користувача і перевірки, чи існує користувач з такою ж електронною поштою. Якщо користувач вже існує, відображається помилка. Якщо ні, дані нового користувача зберігаються в базі даних, після чого надсилається електронний лист з підтвердженням. Після цього користувач бачить сторінку

підтвердження. Цей алгоритм забезпечує безпечний і ефективний процес реєстрації нових користувачів, перевіряючи унікальність електронної пошти та надсилаючи підтвердження для верифікації акаунту.

Алгоритм створення організації, тобто процесу, у якому користувач визначає базову структуру системи, починається з введення даних організації. Система збирає назву, опис та інші базові параметри. Дані нової організації зберігаються, а користувач, який створює організацію, автоматично стає її власником. Далі відбувається збереження даних організації — інформація додається до бази даних. Після цього користувачу показується домашня сторінка організації. Цей алгоритм забезпечує структуру та порядок у системі.

Алгоритм додавання сайту до організації починається з введення даних сайту та перевірки, чи існує сайт з таким URL в системі. Якщо сайт вже існує, відображається помилка. Якщо ні, новий сайт додається до бази даних і прив'язується до організації. Після цього користувач може запланувати парсинг сайту, додати облікові дані для інтеграції з Google Analytics та HubSpot, і, якщо облікові дані додані, конфігурується парсинг, тобто створюється розклад моніторингу для збору даних із сайту. У разі якщо облікові дані для інтеграцій не було введено, користувач перенаправляється на домашню сторінку організації. Цей алгоритм забезпечує інтеграцію нових сайтів до системи, автоматизуючи процес перевірки унікальності, збереження даних, планування парсингу та налаштування інтеграцій з аналітичними інструментами.

Реалізація алгоритмів реєстрації користувача, створення організації та додавання сайту забезпечує основу для побудови ефективної та інтегрованої системи. Вони дозволяють:

- Гарантувати безпеку даних: перевірка унікальності електронної пошти та URL, верифікація користувачів.
- Створювати структуру системи: формування організаційного середовища та інтеграція сайтів.
- Автоматизувати ключові процеси: такі як планування парсингу та налаштування аналітичних інструментів.

Ці алгоритми забезпечують інтуїтивність і простоту для користувачів, а також гнучкість і надійність для подальшого розвитку та інтеграції нових функцій у систему.

Розглянемо решту алгоритмів (рис. 3.5): парсингу, збору аналітики, та моніторингу трафіку.

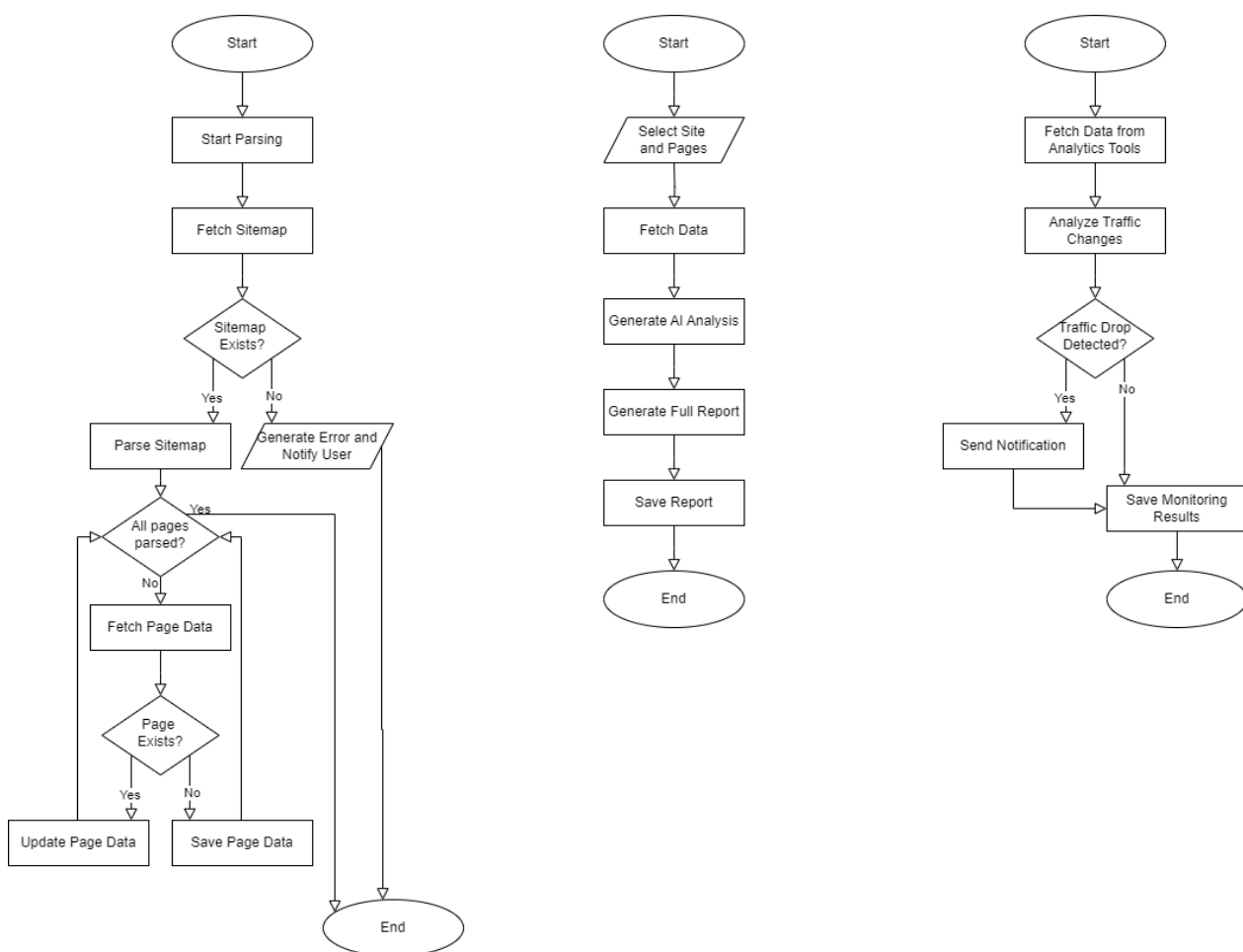


Рис. 3.5. Алгоритми парсингу, збору аналітики та моніторингу трафіку

Алгоритм парсингу сайту автоматизує процес збору даних із сайтів. Він починається з ініціації парсингу користувачем, після чого виконується завантаження sitemap сайту. Якщо sitemap існує, він розбирається і парсяться всі сторінки, зазначені у sitemap. Якщо сторінка вже існує у базі даних, її дані оновлюються, інакше створюється новий запис. Якщо sitemap не існує, генерується помилка і користувач отримує відповідне повідомлення. Цей алгоритм забезпечує автоматичне збирання та оновлення даних про сторінки

сайту, використовуючи інформацію з sitemap, що дозволяє зберігати актуальні дані про вміст сайту.

Алгоритм збору аналітики забезпечує аналіз ефективності контенту. Він починається з вибору сайту та сторінок для аналізу, після чого здійснюється збирання даних. На основі зібраних даних проводиться аналіз з використанням штучного інтелекту (ШІ). Далі створюється повний звіт, який зберігається у базі даних. Цей алгоритм забезпечує автоматичний аналіз контенту сайту за допомогою ШІ та генерацію детальних звітів, що дозволяє отримувати глибоке розуміння ефективності контенту та можливостей його покращення.

Алгоритм моніторингу трафіку відстежує зміни трафіку на сайтах. Він починається з завантаження даних з аналітичних інструментів, тобто система отримує метрики з Google Analytics та інших платформ. Після цього здійснюється аналіз змін трафіку. Якщо виявляється падіння трафіку, користувачу надсилається повідомлення. У будь-якому випадку результати моніторингу зберігаються у базі даних. Цей алгоритм дозволяє автоматично відстежувати зміни трафіку на сайті та оперативно реагувати на суттєві падіння, що допомагає у підтриманні та поліпшенні продуктивності сайту.

Алгоритми парсингу, збору аналітики та моніторингу трафіку формують основу автоматизації процесів збору, аналізу та управління даними. Вони забезпечують:

- Ефективний збір даних: алгоритм парсингу дозволяє автоматично отримувати актуальну інформацію про структуру та вміст сайту.
- Глибокий аналіз контенту: алгоритм збору аналітики інтегрує дані з аналітичних інструментів і використовує штучний інтелект для створення детальних звітів.
- Оперативний моніторинг змін: алгоритм моніторингу трафіку дозволяє швидко реагувати на критичні зміни, сповіщаючи користувачів та зберігаючи результати для подальшого аналізу.

Ці алгоритми сприяють оптимізації процесів управління контентом і забезпечують користувачам точну та своєчасну інформацію для прийняття

рішень. Вони підвищують ефективність роботи системи, роблячи її гнучкою та зручною у використанні.

Кожен із зазначених алгоритмів представлений у вигляді блок-схем. Такі схеми забезпечують зручність аналізу процесів і є основою для їх подальшого вдосконалення. Вони дозволяють виявляти потенційні проблеми на ранніх етапах розробки та оптимізувати роботу системи.

Ці алгоритми формують основу функціонування програмного забезпечення, дозволяючи автоматизувати ключові процеси та покращувати продуктивність і зручність користувачів.

Наступним кроком буде візуалізація користувацького інтерфейсу шляхом створення макету. Макет — це матеріалізований у вигляді набору статичних зображень проєкт. Поведінкові та динамічні стани представляються статичними зображеннями. Сукупність статичних зображень, яка демонструє поведінку системи під час прогону сценарію користувацької взаємодії, називається розкадровкою [4].

Створимо вайрфрейми, які в деталях показують, яка інформація, контент і елементи управління повинні виводитися на кожній сторінці інтерфейсу системи. Також у вайрфреймах вже розставлені акценти щодо різних елементів інтерфейсу: кнопок, зображень, заголовків, текстів, тощо.

Створення вайрфреймів — це один з перших кроків в розробці інтерфейсів для сайтів та мобільних додатків і одна з найважливіших стадій, які впливають на майбутнє вашого продукту. Вайрфрейм, який створюється на ранніх стадіях розробки інтерфейсів, визначає розташування елементів, які користувач бачить на сторінці, а також те, як вони будуть взаємодіяти з ним.

Варто пам'ятати, що вайрфрейми це схеми сторінок, а не кінцевий дизайн системи і всі розміри в ньому відносні, візуалізації елементів є спрощеною, але основні пропорції повинні бути закладені вже на цьому етапі. Чорно-сіро-білий кольори — це типова палітра вайрфрейму (додатково можна використовувати синій, щоб позначити посилання) [25].

Отож, було розроблено вайрфрейми для шести екранів (рис. 3.6) – основних екранів користувацького інтерфейсу, що забезпечує зручність та інтуїтивність використання системи:

- Екран реєстрації;
- Екран створення організації;
- Екран додавання сайту;
- Екран вигляду звіту після парсингу;
- Екран вигляду звіту з аналітичними даними з HubSpot, Google Analytics та даними ШІ аналізу;
- Екран налаштувань.

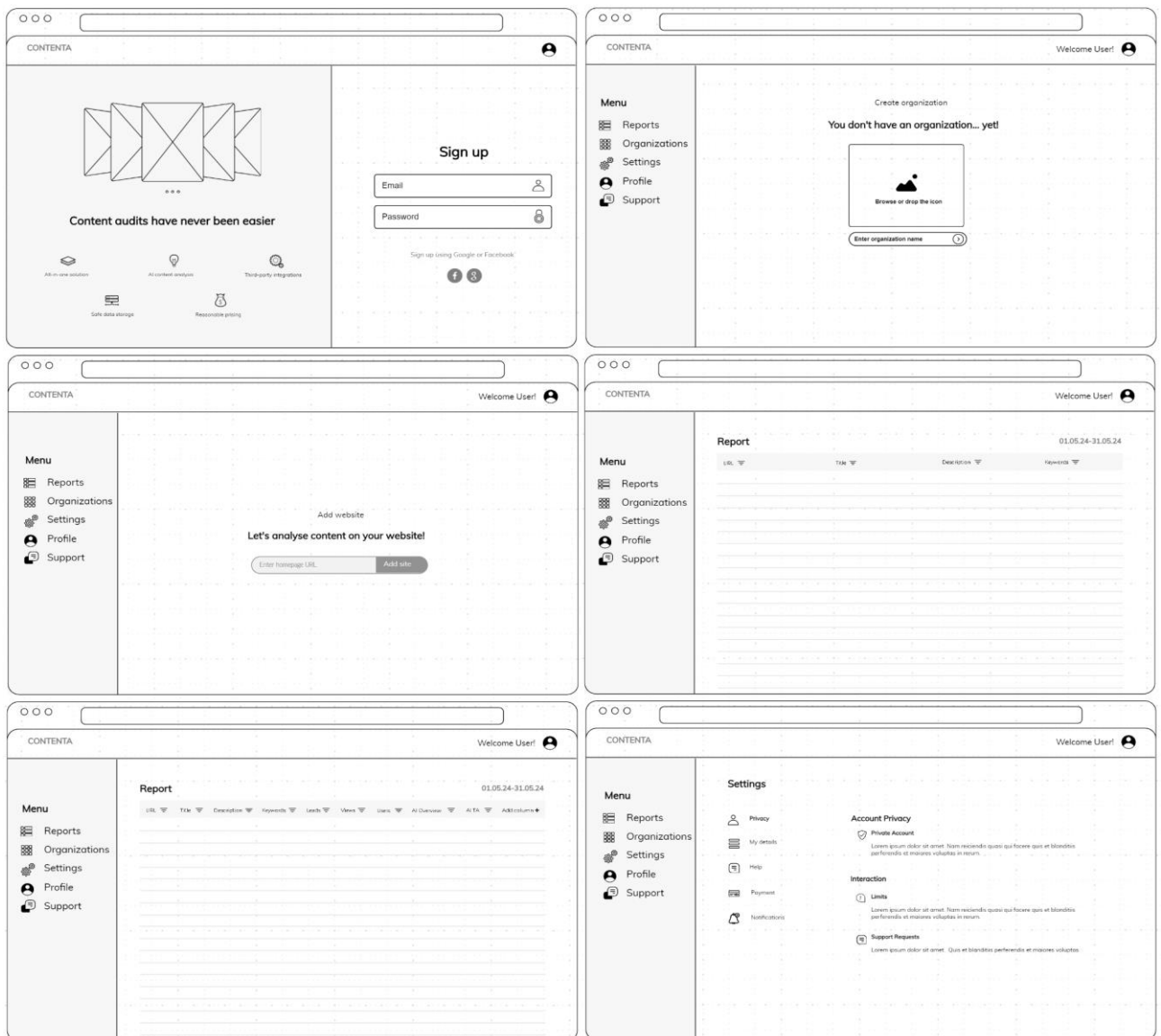


Рис. 3.6. Вайрфрейми 6 екранів ПЗ для аудиту та аналізу контенту

Ці доробки можна використовувати для постановки завдань дизайнеру та розробникам, а також для демонстрації стейкхолдерам і потенційним користувачам.

3.5 Опис технічного стеку програмного забезпечення проєкту

Технічний стек програмного забезпечення (Technology Stack) є основою для розробки та впровадження будь-якої інформаційної системи, включаючи програмне забезпечення для аудиту та аналізу маркетингового контенту. Вибір технологій визначає продуктивність, масштабованість, безпеку та інтеграційні можливості створюваного продукту.

Технологічний стек — це набір інструментів, мов програмування та технологій, які працюють разом для створення цифрових продуктів або рішень, таких як вебсайти, мобільні або вебпрограми. Інакше технологічний стек визначається набір технологій, які організація використовує до створення ПЗ. Це комбінація мов програмування, фреймворків, бібліотек, шаблонів, серверів, рішень UI/UX, програмного забезпечення та інструментів, які використовуються розробниками [23].

У цьому підрозділі розглядаються ключові компоненти технічного стеку, які планується використати в розробці програмного забезпечення проєкту. Описано основні мови програмування, фреймворки, бібліотеки, бази даних, інструменти для автоматизації процесів розробки та технології інтеграції із зовнішніми сервісами. Обґрунтовано вибір кожного елемента стеку з точки зору його відповідності вимогам проєкту, враховуючи такі фактори, як:

- Забезпечення високої продуктивності та мінімального часу відгуку.
- Можливість масштабування програмного забезпечення у разі зростання обсягу даних або користувачів.
- Зручність у розробці та супроводі програмного забезпечення.
- Інтеграційні можливості з аналітичними платформами (Google Analytics, HubSpot тощо).

- Забезпечення захисту даних відповідно до сучасних стандартів безпеки.

Аби сформувати технічний стек, що дозволяє ефективно реалізувати бізнес-завдання проєкту та задовольнити потреби кінцевих користувачів, було проведено ряд консультацій з фахівцями відділу розробки на базі виробничої практики. В результаті були отримані важливі рекомендації, що стосуються вибору мови програмування, архітектури, підходів до розробки, інфраструктури та інструментів для створення інтерфейсу, а саме:

- Розробники пропонують зупинитися на мікросервісній архітектурі як найоптимальнішій для забезпечення масштабованості, гнучкості та надійності ПЗ. Така архітектура дозволить розподілити основні функціональні модулі (наприклад, парсер контенту, ШІ-аналізатор, модуль інтеграції з Google Analytics і HubSpot) на окремі мікросервіси, що зможуть незалежно оновлюватися та підтримуватися. Крім того, мікросервісна архітектура дозволяє підбирати оптимальні технології для вирішення задач окремих мікросервісів, а також ефективно масштабувати процеси розробки.
- Для розробки бекенду фахівці рекомендують звернути увагу на популярність мов програмування на ринку та специфіку функціональних та нефункціональних вимог до кожного з мікросервісів окремо. Так, для сервісів, які займаються обробкою користувацьких запитів, варто використовувати мови програмування, які забезпечують неблокуючий ввід-вивід, наприклад Java, Python, Node.js, Go та ін. Для мікросервісів, які займаються обробкою даних, варто використовувати мови програмування, які забезпечують найвищий рівень продуктивності, наприклад Go, Scala, Erlang, Elixir. Сервіс, який реалізує алгоритми аналізу даних, використовує специфічні бібліотеки та інструменти, зазвичай для таких задач використовується Python чи Erlang.
- Для фронтенду було рекомендовано використовувати React з огляду на його гнучкість та активну підтримку спільнотою, а також можливість швидкого оновлення інтерфейсу, що є важливим для аналітичного ПЗ.

React також дозволяє створювати кастомні компоненти для візуалізації метрик та даних, що полегшує роботу користувачів з дашбордом.

- Щодо інфраструктури, було рекомендовано розміщувати проєкт у хмарному середовищі, наприклад, AWS. Таке рішення надасть готові до використання надійні і масштабовані інструменти вендора для розподілу ресурсів, розгортання сервісів, інтеграції даних, зберігання та обробки великих обсягів даних. Окрім того, хмарна інфраструктура дозволить використовувати сервіси для автоматизації та безперервної інтеграції, а також налаштувати резервне копіювання та безпечне зберігання даних.
- Для зберігання структурованих даних (наприклад, користувацької інформації та результатів аудиту) було рекомендовано використовувати PostgreSQL як основну базу даних. PostgreSQL це система керування базами даних корпоративного класу з відкритим кодом. Він підтримує як SQL, так і JSON для реляційних і нереляційних запитів для розширюваності та відповідності SQL. PostgreSQL підтримує розширені типи даних і функції оптимізації продуктивності, які доступні лише в дорогих комерційних базах даних, наприклад Oracle і SQL Server [31].
- Фахівці відзначили важливість налаштування стабільної інтеграції з API Google Analytics, HubSpot та інших сторонніх сервісів для отримання релевантних даних та забезпечення комплексного аудиту. Було рекомендовано створити окремий мікросервіс для роботи з інтеграціями, що зменшить залежність основної системи від сторонніх сервісів та забезпечить можливість швидкого налаштування нових інтеграцій у майбутньому.
- Для забезпечення безперебійної роботи та зручності управління, команда розробників рекомендувала налаштувати CI/CD процеси (наприклад, за допомогою Jenkins або GitLab CI/CD) для автоматизації збору, тестування та розгортання оновлень ПЗ. CI/CD є поширеною DevOps-практикою. CI (Continuous Integration) — це неперервна інтеграція, а CD (Continuous Delivery) — неперервна доставка. Цей набір методик дозволяє

розробникам частіше і надійніше розгортати зміни в програмному забезпеченні [24]. Крім того, використання контейнеризації (Docker) і Kubernetes для управління мікросервісами допоможе значно спростити налаштування середовищ, полегшити розгортання оновлень та забезпечити високу надійність системи. Контейнеризація — це розгортання програмного забезпечення разом з усіма необхідними компонентами: код, бібліотеки, фреймворки тощо, таким чином, щоб вони були ізольовані у власному контейнері. Якщо взяти традиційні методи розробки ПЗ, то цей процес відбувається в конкретному обчислювальному середовищі. Якщо його перенести в інше місце, наприклад, з ПК у хмару чи з Linux на Windows, може виникнути низка помилок та багів. За допомогою контейнеризації можна об'єднати всі складові програмного забезпечення у так званій контейнер. Він з усім вмістом абстрагується від хостової операційної системи, є автономним і може без проблем працювати на будь-якій платформі [27].

- Фахівці звернули увагу на важливість забезпечення безпеки та захисту даних. Було рекомендовано реалізувати механізми аутентифікації та авторизації (наприклад, через OAuth 2.0 для інтеграцій — це протокол авторизації, який надає одним вебдодаткам можливість отримати доступ до ресурсів іншого додатка від імені користувача, без необхідності розкривати свої облікові дані [30]), а також налаштувати резервне копіювання даних, щоб уникнути втрати важливої інформації у разі системних збоїв. Крім того, система зберігає користувацькі ключі доступу до сторонніх сервісів, які потрібно зашифрувати для мінімізації ризику їх розкриття при можливих витіках даних.
- Для моніторингу роботи ПЗ та своєчасного виявлення потенційних проблем рекомендується налаштувати систему збору телеметрії (наприклад, Prometheus, Grafana, ELK stack), яка дозволить відстежувати продуктивність сервісів і оперативно реагувати на зміни. Це також

забезпечить прозорість процесів і дасть змогу аналізувати ефективність роботи системи.

Підсумовуючи, обраний технічний стек програмного забезпечення забезпечує всебічну підтримку потреб проєкту, починаючи від гнучкої мікросервісної архітектури і закінчуючи інтеграцією з провідними аналітичними платформами. Такий підхід дозволяє створити систему, яка буде масштабованою, продуктивною, безпечною та легкою у підтримці. Крім того, технічні рішення враховують перспективи майбутніх оновлень, забезпечуючи можливість швидкої адаптації до змін у бізнес-процесах або вимогах та очікуваннях користувачів.

Інтеграція сучасних технологій та інструментів автоматизації забезпечить високий рівень ефективності розробки та експлуатації системи, дозволяючи оптимізувати ресурси команди. Налаштування процесів CI/CD та контейнеризація не тільки підвищують надійність, але й сприяють безперервному вдосконаленню системи. Таким чином, технічний стек програмного забезпечення виступає стратегічним фундаментом для успішної реалізації проєкту та досягнення його цілей.

РОЗДІЛ 4. ПЛАНУВАННЯ ЕЛЕМЕНТІВ УПРАВЛІННЯ ПРОЄКТОМ

4.1 Розробка організаційної структури управління проєктом.

Формування команди проєкту

Управління проєктом зі створення програмного забезпечення для аудиту та аналізу маркетингового контенту вимагає чіткої організаційної структури, яка забезпечує ефективну взаємодію всіх учасників команди.

Організаційна структура проєкту (Organization Breakdown Structure, OBS) — ієрархічно організоване зображення організації проєкту, влаштоване таким чином, щоб співвідносити пакети робіт з виконуваними організаційними одиницями [18].

Розподіл відповідальності між усіма елементами підприємства (від рядових співробітників до керівників вищої ланки) є тим критерієм, за яким організаційні структури поділяються на типи: вертикальна (ієрархічна) та горизонтальна (мережева).

За вертикальної структури управління наявний контроль нижчих рівнів вищими. Одним з головних ознак вертикальної структури управління є централізація — тобто зосередженість влади в одних руках (генеральний директор) або в органі управління (рада директорів). Фундаментом або нижнім рівнем вертикальної структури управління є рядові співробітники, яким ніхто не підкоряється, вони не мають широти повноважень і статусу, але мають значний вплив на результати діяльності підприємства. Верхівкою вертикальної структури є дуже малочисельне, але наділене найбільшим статусом і широтою повноважень, керівництво або глава компанії. Саме на них лежить максимальна відповідальність за прийняття стратегічних рішень, контроль над усіма нижчими елементами підприємства. Між фундаментом і вершиною вертикального типу управління знаходяться проміжні ланки управління, що в тій чи іншій мірі володіють як статусом, так і широтою повноважень, безпосередньо впливають на якість і мотивацію інших поверхів структури [32].

Горизонтальна структура управління передбачає команди, які самоорганізуються; відсутність поділу на рівні; децентралізацію управління підрозділами компанії. Така структура дозволяє прискорити прийняття рішень, зменшити роль і кількість менеджерів середньої ланки, значно скоротивши бюрократію. Рядові співробітники отримують більше можливостей для ініціативи, вплив їх на процеси зростає, а керівництво ставить тільки стратегічні цілі. Головна проблема такої структури — в пошуку самостійного, ініціативного і мотивованого персоналу, адже мало хто здатний ефективно працювати в команді «без нагляду» [32].

Організаційна структура управління підприємством часто містить в собі одночасно елементи вертикальної та горизонтальної структур управління, проте загалом домінує перша. Найбільш поширені такі три типи:

- Лінійна структура управління є класичною вертикальною структурою, в якій головному керівнику підпорядковується і звітує керівник нижчої ланки, а йому — колектив працівників компанії. Співробітники в такій структурі звітують тільки перед своїм безпосереднім керівником, який відповідальний за результати їх роботи перед вищим керівництвом.
- Функціональна структура — фахівці одного рівня об'єднуються в спеціалізовані підрозділи. Тобто спеціалісти з продажів — у відділ продажів, усі фахівці бухгалтерії — у фінансовий відділ тощо. До середньої ланки така структура буде аналогічна лінійній, а от нижче — вже формуватиметься за функціональною ознакою.
- Матрична структура — містить в собі елементи горизонтальної та вертикальної структур управління. Тут вертикальна побудова, створюються відділи. Разом з цим здійснюється горизонтальне управління проектами та програмами [32].

Для управління проектом зі створення програмного забезпечення для аудиту та аналізу маркетингового контенту було обрано матричну організаційну структуру. Вона дозволяє зберігати централізоване керівництво при забезпеченні гнучкості та оперативності у виконанні завдань.

Матрична структура управління проектом має такі ключові переваги:

- Вертикальна організація забезпечує чіткий контроль і підзвітність кожного члена команди. Вищий рівень управління відповідає за стратегічні рішення, а проміжні ланки контролюють якість виконання робіт на рівні команд.
- Горизонтальне управління сприяє підвищенню самостійності команд та залученню міждисциплінарних спеціалістів до реалізації окремих компонентів проекту. Це дозволяє зменшити роль бюрократії, прискорити прийняття рішень і підвищити ініціативність працівників.

Матрична структура є оптимальною для комплексних інноваційних проектів, де потрібна взаємодія між різними функціональними підрозділами. Вона передбачає:

- Проектного менеджера — керує командою, здійснює планування, моніторинг виконання робіт, управління ризиками та ресурсами.
- Функціональних керівників (якщо потрібні, наприклад, технічний директор чи куратори окремих напрямів), які координують спеціалістів у своїх галузях.
- Спеціалістів проектної команди, які взаємодіють горизонтально і вертикально у межах визначених завдань.

При управлінні проектами особлива увага приділяється формуванню команди проекту, тому що ефективність управління проекту багато в чому залежить від ефективності роботи команди.

Для команди проекту основними характеристиками є єдність мети, сумісна діяльність, несуперечність інтересів, автономність діяльності, колективна і взаємна відповідальність за результат, спеціалізація і взаємодоповнюваність ролей, стабільність (стійкість) команди [6].

Основними факторами, що визначають принципи формування команди проекту, є специфіка проекту (визначає формальну структуру команди, рольовий склад, основні обмеження проекту), організаційно-культурне середовище, особливості особистого стилю взаємодії керівника й лідера з

членами команди. Склад команди проєкту залежить від специфіки проєкту і його розміру [6].

Отже, склад команди було сформовано з урахуванням ключових вимог проєкту та ролей, які мають забезпечити якісну розробку програмного забезпечення для аудиту та аналізу контенту:

- Проєктний менеджер — відповідальний за загальне керівництво, контроль виконання завдань, управління ресурсами та ризиками.
- Архітектор — проєктує архітектуру системи, визначає технічні вимоги, керує командою розробників.
- Команда розробників (3 особи) — реалізує функціональність програмного забезпечення, відповідає за написання і тестування коду.
- UX/UI дизайнер — розробляє інтуїтивно зрозумілий інтерфейс для забезпечення зручності використання.
- Тестувальник — проводить тестування, забезпечує якість програмного забезпечення.
- Маркетолог — аналізує ринок, визначає цільову аудиторію, створює стратегію просування.
- Спеціаліст підтримки — надає технічну підтримку користувачам, забезпечує зворотний зв'язок із клієнтами.

Таким чином, можемо візуалізувати організаційну структуру проєкту (рис. 4.1), враховуючи специфіку проєкту та програмного забезпечення, що розроблятиметься. Усього над проєктом працюватиме 9 осіб — це невелика команда, що дозволяє уникнути складної ієрархії і ефективніше взаємодіяти один з одним для виконання задач проєкту.

Оскільки розмір проєкту малий, то менеджер проєкту може виконувати декілька ролей (наприклад, на нього буде покладено підбір і найм команди та ведення бухгалтерії).

Також дві ролі буде у архітектора — окрім побудови архітектури системи він також займатиметься інфраструктурою та керуватиме командою розробки,

адже цей співробітник матиме високий рівень технічної експертизи і глибоке розуміння програмного забезпечення з точки зору його реалізації.

Решта співробітників будуть виконавцями у своєму напрямку, а керівником виступатиме безпосередньо проєктний менеджер.

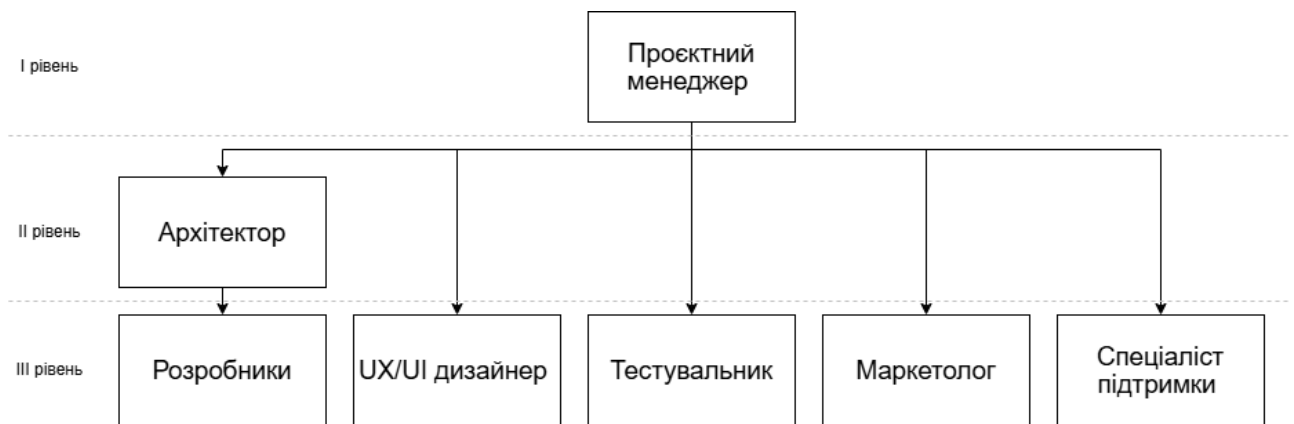


Рис. 4.1. Організаційна структура проєкту

Оскільки проєкт передбачає розробку програмного забезпечення, найбільшою командою виступає команда розробки, до якої входять три розробники та архітектор, який виступає їхнім функціональним керівником.

Кожен з інших напрямків — дизайн, тестування, маркетинг, підтримка — на початковому етапі реалізації та впровадження проєкту буде представлений однією особою, тому немає необхідності виділяти формальні команди і призначати функціональних керівників. На такому етапі і масштабі команди керування цими напрямками здійснюватиме проєктний менеджер.

Вибір членів команди базується на кількох ключових аспектах, які забезпечують якісну реалізацію проєкту. Досвід роботи у відповідній сфері є одним із визначальних чинників, адже кожен спеціаліст повинен володіти знаннями, що дозволять виконати конкретні завдання. Наприклад, архітектор має вміти проєктувати технічну архітектуру, програмісти — розробляти функціональні модулі, а UX/UI дизайнер — створювати інтерфейси, які відповідають потребам користувачів.

Не менш важливим є підтвердження професійних компетенцій через сертифікати, такі як Scrum Master для проєктного менеджера чи ISTQB для

тестувальника. Це не лише демонструє рівень знань спеціаліста, а й підтверджує його систематичний підхід до саморозвитку.

Важливою умовою є наявність навичок роботи у багатофункціональних командах. Успішна реалізація проєкту вимагає тісної співпраці між представниками різних спеціалізацій, де комунікація та взаєморозуміння відіграють ключову роль. Додатково, готовність працювати в умовах адаптивного середовища дозволяє команді швидко реагувати на зміни у вимогах чи умовах розробки. Ця здатність до гнучкості сприяє підтриманню темпів роботи навіть у випадках невизначеності.

Організація роботи команди базується на регулярних комунікаціях, ефективному плануванні та моніторингу результатів. Щотижневі зустрічі (стендапи) дозволяють синхронізувати учасників проєкту, обговорити досягнення, визначити проблеми та скоригувати плани. Кожен учасник звітує про виконану роботу, свої плани на майбутнє та перешкоди, що виникають у процесі роботи.

Для управління задачами використовуються сучасні інструменти, такі як Jira для організації беклогу й спринтів або Trello для візуалізації прогресу. Також забезпечується оперативна комунікація за допомогою Slack. Ці платформи сприяють прозорості процесів та дозволяють уникати непорозумінь у роботі.

Важливою складовою управління є моніторинг ключових показників ефективності. Це дозволяє оцінювати продуктивність команди, якість виконаних завдань, дотримання строків і рівень задоволеності клієнтів. Чітке розподілення обов'язків між членами команди забезпечує відповідальність за конкретні аспекти проєкту, зокрема архітектор проєктує технічну частину, програмісти пишуть код, а тестувальник контролює якість.

Запропонована структура забезпечує чітке розподілення обов'язків, ефективну комунікацію та контроль за досягненням проєктних цілей у встановлені строки та бюджети, забезпечує гнучкість і високу якість

результатів. Вона об'єднує централізований контроль із можливістю кожного члена команди ефективно виконувати свої завдання.

4.2 Розробка ієрархічної структури робіт проєкту та матриці відповідальностей

Методи моделювання є ключовими інструментами для управління та контролю інформаційних технологічних процесів. Одним з найефективніших методів є використання ієрархічної структури розподілу робіт.

За визначенням, наведеним у Зводі знань з управління проєктами (PMBOK 7), ієрархічна структура робіт (Work Breakdown Structure, WBS) — це ієрархічна декомпозиція всього обсягу робіт, який команда проєкту повинна виконати для досягнення цілей проєкту та створення необхідних доробків [17].

Тобто WBS — це ієрархічна та інкрементна декомпозиція проєкту на фази, результати та пакети робіт. Це деревоподібна структура, яка показує розподіл зусиль, необхідних для досягнення мети, наприклад, програми, проєкту або контракту.

Цей підхід дозволяє зосередитися на розробці конкретних частин продукту, що сприяє більш детальному контролю за якістю та термінами виконання робіт.

WBS є важливим засобом для організації робіт, розподілу відповідальності, оцінки вартості та створення системи звітності. Вона дозволяє ефективно підтримувати процедури збору інформації про виконання робіт і відображати результати в управлінській системі для узагальнення графіків робіт, вартості, ресурсів і дат завершення [1].

Розроблення такої структури починається з того, що проєкт в цілому розглядається як кінцевий результат. Спочатку виділяють основні проміжні результати роботи над проєктом, потім визначають, яких дрібніших результатів треба досягти, щоб дійти до основних проміжних результатів. Процедуру

повторюють до тих пір, поки не виділяють найдрібніший неподільний) елемент, який підлягає управлінню і за який відповідатиме тільки один виконавець [6].

Добре розроблена WBS дозволяє легко віднести кожен діяльність проекту до одного і тільки одного кінцевого елемента WBS. На додаток до своєї функції обліку витрат, WBS також допомагає відображати вимоги з одного рівня специфікації системи на інший, наприклад, матриця перехресних посилань відображає функціональні вимоги до проектної документації високого або низького рівня.

WBS може відображатися горизонтально у вигляді схеми або вертикально у вигляді деревовидної структури (як організаційна діаграма).

Розробка WBS зазвичай відбувається на початку проекту і передують детальному плануванню проекту і завдань [14].

Для розробки проекту було створено WBS з використанням програмного забезпечення WBS Chart Pro.

WBS по життєвому циклу (рис. 4.2) складається зі 109 елементів і має 6 рівнів ієрархії. Вона організовує роботи за фазами життєвого циклу проекту: ініціювання проекту, планування, аналіз вимог, проектування, розробка продукту, тестування, реліз, завершення проекту.

Цей підхід дозволяє структурувати проект відповідно до стандартних фаз управління проектами, що сприяє кращому розумінню послідовності робіт і забезпечує ефективне планування ресурсів та часу.

Контроль і управління інформаційними технологічними процесами включає кілька важливих задач:

- Моніторинг прогресу — відстеження виконання завдань та виявлення відхилень від плану.
- Контроль якості — оцінка якості виконаних робіт та вжиття заходів для покращення.
- Контроль витрат — аналіз витрат та корекція бюджету за необхідності.

- Завершення проєкту — задача проєкту замовнику, проведення фінальної перевірки та закриття проєкту [7].

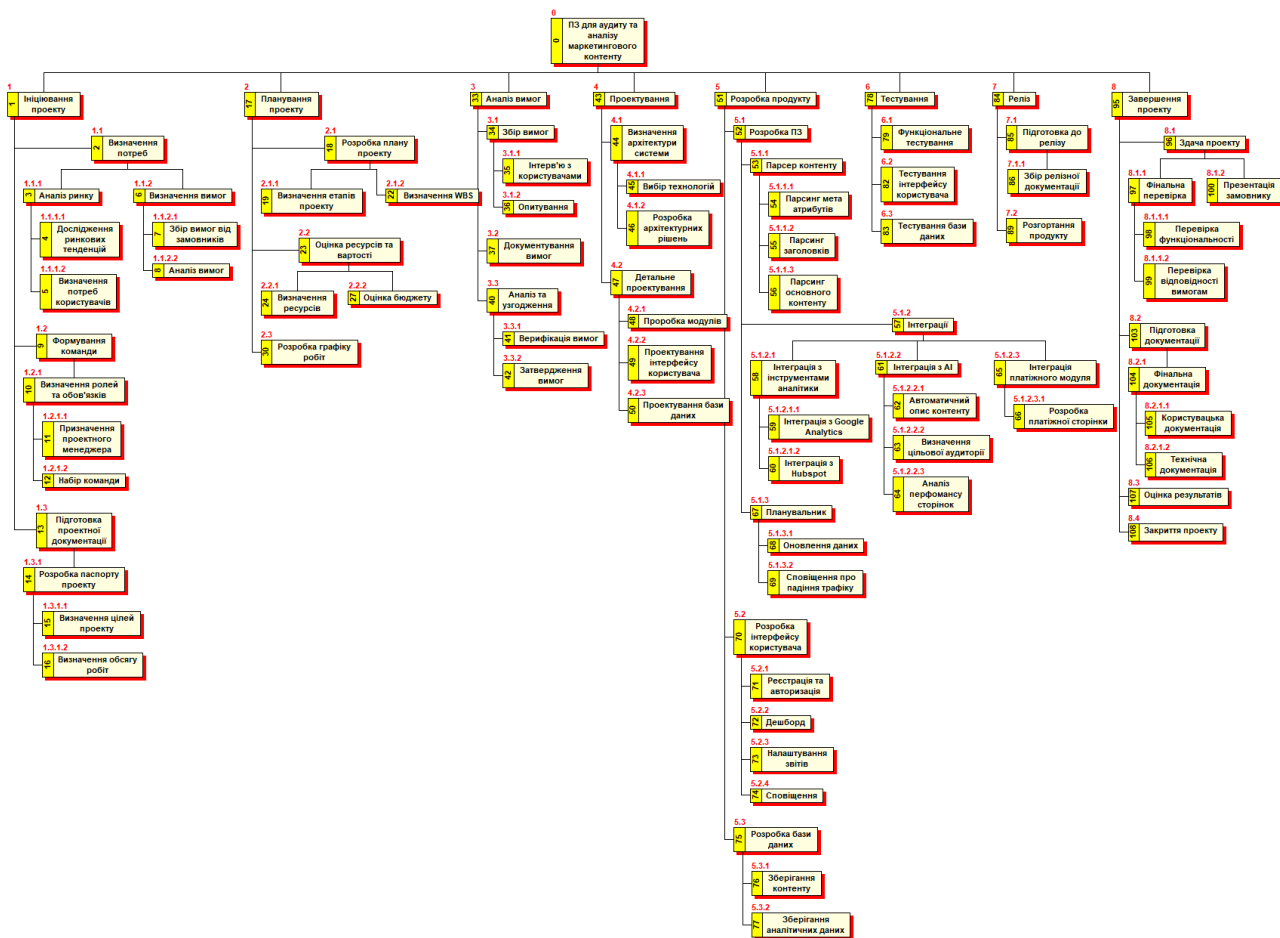


Рис. 4.2. WBS по життєвому циклу

Ці задачі є критично важливими для успішної реалізації проєкту та забезпечення його відповідності встановленим вимогам і стандартам.

Використання WBS дозволяє структурувати роботи проєкту, забезпечуючи ефективний контроль та управління інформаційними технологічними процесами. Це сприяє підвищенню якості виконання робіт, оптимізації витрат ресурсів і часу, а також забезпечує досягнення поставлених цілей проєкту.

Перейдемо до розробки матриці відповідальностей для проєкту розробки та впровадження ПЗ для аудиту та аналізу маркетингового контенту.

Матриця розподілу завдань і відповідальності — це інструмент планування, призначений для встановлення зв'язку роботи, визначеної в

структурі проєкту, з організаційними одиницями, субпідрядниками і окремими співробітниками.

Таким чином, з одного боку, існує організаційна структура проєкту (OBS), а з іншого — робота, що виконується відповідно до ієрархічної структури проєкту (WBS). Мета матриці розподілу відповідальності — об'єднання цих двох структур.

Розроблення такої матриці покаже, чи добре складена структура проєкту, чи досить вона деталізована, чи не надлишкова деталізація. Також планувальникові надається можливість упевнитися, що кожному виконавцеві призначено як мінімум одне завдання.

Складена належним чином матриця завдань і відповідальності є відправною точкою для подальшого планування, управління і контролю [6].

У зарубіжних джерелах цей інструмент має назву RACI matrix і використовується для уточнення ролей та обов'язків команди в межах завдань. Ролі класифікуються як Відповідальний (Responsible), Звітувальник (Accountable), Консультант (Consulted) та Поінформований (Informed). Розглянемо детальніше ці ролі:

- Відповідальний (Responsible, R):
 - Особи, які відповідають за виконання завдань або створення кінцевого результату. Це безпосередньо виконавці, які роблять внесок у завершення завдань.
 - Приклади: менеджери проєктів, аналітики, розробники, графічні дизайнери, копірайтери.
- Звітувальник (Accountable, A):
 - Ті, хто гарантує виконання завдань у встановлені терміни. Часто відповідають також за інформування інших.
 - Приклади: власники продуктів, спонсори, ключові зацікавлені сторони.

- Консультант (Consulted, C):
 - Особи, чия думка або експертна оцінка є важливими. Вони забезпечують зворотний зв'язок, який необхідно враховувати на всіх етапах роботи.
 - Приклади: юристи, експерти з кібербезпеки.
- Поінформований (Informed, I):
 - Особи, які мають бути в курсі проекту, але не беруть участі у прийнятті рішень.
 - Приклади: члени комітетів, зовнішні стейкхолдери.

RACI матриця сприяє організації роботи команд і підвищує прозорість, що дозволяє уникнути плутанини і забезпечити ефективність [36].

Для створення RACI матриці, або матриці відповідальностей, необхідно спершу визначити масштаб та ключові завдання, а також учасників проекту. Після цього можна розподілити ролі (R, A, C, I) для кожного завдання. Готову матрицю потрібно узгодити з командою та зі стейкхолдерами. Оскільки перші етапи вже виконано, перейдемо до розподілення ролей (табл. 4.1).

Таблиця 4.1

Матриця відповідальності (RACI)

Завдання	ПМ	Архітектор	Розробники	Дизайнер	Тестувальник	Маркетолог	Сапорт
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
1.1. Визначення потреб	AR	I				C	
1.1.1. Аналіз ринку	A					R	
1.1.2. Визначення вимог	AR	C				C	
1.2. Формування команди	AR	C					
1.3. Підготовка проектної документації	AR	C				C	
2.1. Розробка плану проекту	AR	I	I	I	I	I	I
2.2. Оцінка ресурсів та вартості	AR						
2.3. Розробка графіку	AR	C		C	C		

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
робіт							
4.1. Архітектурне проектування	A	R	I		I		
4.2.1. Проектування модулів	A	R	C		I		
4.2.2. Проектування інтерфейсу користувача	A	C	I	R	I		
4.2.3. Проектування бази даних	A	R	I		I		
5.1. Розробка ПЗ	A	C	R		I		
5.2. Розробка інтерфейсу користувача	A	I	R	R	I		
5.4. Розробка бази даних	A	C	R		I		
6.1. Функціональне тестування	A	I	C		R		
6.2. Тестування інтерфейсу користувача	A			C	R		
6.4. Тестування бази даних	A	CI	C		R		
7.1. Підготовка до релізу	AR	C	I	I	I	C	I
7.2. Розгортання продукту	A	C	R	I	R	I	I
8.1. Підтримка користувачів після впровадження	A		CI		C	C	R

Таким чином, було розподілено ролі для ключових завдань проекту серед всіх учасників команди.

RACI матриця є важливим інструментом управління проектом, який дозволяє чітко розподілити ролі та зони відповідальності між учасниками проектної команди. На основі побудованої матриці можна зробити наступні висновки:

- Ролі розподілені раціонально, що забезпечує уникнення дублювання функцій і конфлікту відповідальності.

- Проектний менеджер (ПМ) виступає ключовою фігурою, яка відповідає за координацію і затвердження більшості завдань, забезпечуючи стратегічне керівництво проектом.
- Архітектор є основним відповідальним за технічні аспекти проекту, включаючи архітектурне проектування, проектування бази даних і модулів.
- Розробники зосереджені на реалізації функціональності програмного забезпечення, працюючи в тісному контакті з архітектором і тестувальником.
- Дизайнер активно залучений до завдань, пов'язаних із розробкою інтерфейсу користувача, забезпечуючи його інтуїтивність та зручність.
- Тестувальник відповідає за перевірку якості програмного забезпечення на всіх рівнях, забезпечуючи його надійність і функціональність.
- Маркетолог відповідає за початкові етапи аналізу ринку і визначення потреб, а також бере участь у підготовці релізу продукту з точки зору маркетингових комунікацій.
- Спеціаліст підтримки залучений до етапів впровадження та забезпечення підтримки користувачів, що гарантує зворотний зв'язок після запуску.
- Інформування (I) дозволяє всім членам команди отримувати актуальну інформацію про прогрес завдань, забезпечуючи їх залученість і розуміння поточного стану проекту.
- У процесі побудови RACI-матриці завдання структувалися за допомогою кодів задач із Work Breakdown Structure (WBS). Впровадження кодів WBS у матрицю підвищує її узгодженість із загальним планом проекту, дозволяючи легко відстежувати взаємозв'язки між задачами та відповідальними особами.

У роботі над проектом слід регулярно переглядати RACI матрицю для врахування змін у складі команди чи пріоритетах завдань, аналізу ефективності комунікацій і уникнення перевантаження окремих членів команди.

Необхідно також забезпечити доступність матриці для всієї команди як орієнтир для їхньої діяльності.

Завдяки побудові RACI-матриці команда матиме чітке уявлення про ролі, обов'язки та взаємодію, що сприятиме успішному виконанню проєкту.

4.3 Розробка календарного плану. Планування термінів проєкту

Успішне виконання будь-якого проєкту значною мірою залежить від детального планування його етапів, ресурсів та термінів. Календарне планування є одним із ключових елементів управління проєктом, адже дозволяє чітко визначити часові рамки для виконання завдань, встановити взаємозв'язки між ними та забезпечити контроль за досягненням проміжних результатів.

Календарний план — це такий проєктно-технологічний документ, який визначає послідовність, інтенсивність і тривалість виробництва робіт, їх взаємозв'язку, а також потреби (з розподілом у часі) в матеріальних, технічних, трудових, фінансових та інших використовуваних ресурсах [6].

У рамках даного проєкту особливу увагу приділено розробці ефективного календарного плану, що відповідає специфіці Agile-методології.

Гнучка розробка програмного забезпечення (Agile) відноситься до групи методологій розробки програмного забезпечення, заснованих на ітеративній розробці, де вимоги і рішення розвиваються завдяки співпраці між самоорганізаційними крос-функціональними командами.

Agile-методи зазвичай сприяють впорядкованому процесу управління проєктами, який передбачає часті перевірки і адаптацію, командну роботу, самоорганізацію і звітність. Це набір передових методів проєктування, призначених для швидкого релізу високоякісного програмного забезпечення, і бізнес-підхід, який прив'язує розвиток продукту до потреб клієнтів і цілей компанії [11].

Обраний підхід передбачає ітеративний процес виконання робіт через спринти, кожен із яких спрямований на досягнення конкретних цілей. Це

дозволяє регулярно отримувати функціональні результати, швидше виявляти та виправляти помилки, а також оперативно реагувати на зміну пріоритетів, забезпечує прозорість процесу розробки, можливість швидко адаптуватися до змінних умов, а також отримувати регулярний зворотний зв'язок від зацікавлених сторін.

Вибір Agile-методології для управління проектом обґрунтований її гнучкістю, орієнтованістю на кінцевий результат і здатністю забезпечити ефективну комунікацію між усіма учасниками процесу [22].

Agile підходить для цього проекту, оскільки:

- Забезпечує гнучке реагування на зміни у вимогах або зовнішньому середовищі.
- Сприяє залученню всіх зацікавлених сторін у процес управління проектом.
- Дає змогу ефективно використовувати командний час та ресурси через короткі ітерації.

Таким чином, для організації роботи було розроблено календарний план, що складається з 16 спринтів, кожен із яких триватиме 10 робочих днів (2 календарні тижні). Спринти включають виконання конкретних завдань для досягнення поставлених цілей (табл. 4.2). У якості дати початку роботи над проектом обрано 3 лютого 2025 року.

Таблиця 4.2

Перелік спринтів

№ спринту	Назва етапу	Початок спринту	Завершення спринту	Основні виконавці
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
1	Ініціація та планування	03.02.25	14.02.25	ПМ
2	Збір та аналіз вимог	17.02.25	28.02.25	ПМ
3	Проектування системної архітектури	03.03.25	14.03.25	Архітектор, дизайнер, ПМ

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
4	Налаштування середовища розробки	17.03.25	28.03.25	Розробники
5	Розробка дашборда	31.03.25	11.04.25	Розробники
6	Розробка парсера	14.04.25	25.04.25	Розробники
7	Розробка планувальника	28.04.25	09.05.25	Розробники
8	Розробка ШІ аналізатора	12.05.25	23.05.25	Розробники
9	Розробка бази даних	26.05.25	06.06.25	Розробники
10	Інтеграція з Google Analytics і HubSpot	09.06.25	20.06.25	Розробники
11	Розробка платіжного модуля	23.06.25	04.07.25	Розробники
12	Покращення UX/UI	07.07.25	18.07.25	Дизайнер, Розробники
13	Комплексне тестування	21.07.25	01.08.25	Тестувальник, Розробники
14	Підготовка до розгортання	04.08.25	15.08.25	Розробники, ПМ, Тестувальник, Сапорт
15	Маркетинг та запуск	18.08.25	29.08.25	Маркетолог, ПМ, Сапорт, Розробники, Тестувальник
16	Завершення проєкту	01.09.25	12.09.25	ПМ

Таким чином, тривалість проєкту складе 160 робочих днів (або 7,5 календарних місяців), а його завершення припаде на 12 вересня 2025 року.

Використання інструментів, таких як Microsoft Project, дозволило візуалізувати календарний план у вигляді діаграми Ганта, забезпечуючи зручність аналізу прогресу та оцінки витрачених ресурсів.

Діаграма Ганта — горизонтальна лінійна діаграма, на якій роботи проєкту подаються протяжними в часі відрізками, датами, що характеризуються початками і закінченнями, затримками і, можливо, іншими часовими параметрами.

Таку діаграму будують у двох вимірах: по вертикалі йде перелік робіт проєкту, а по горизонталі — шкала часу [6].

Отже, діаграма Ганта (рис. 4.3) відображає календарне планування проєкту з поділом на спринти, відповідно до обраної методології Agile. Графік демонструє часові рамки виконання кожного спринту, їх тривалість, а також взаємозв'язки між задачами (див. Додаток В).

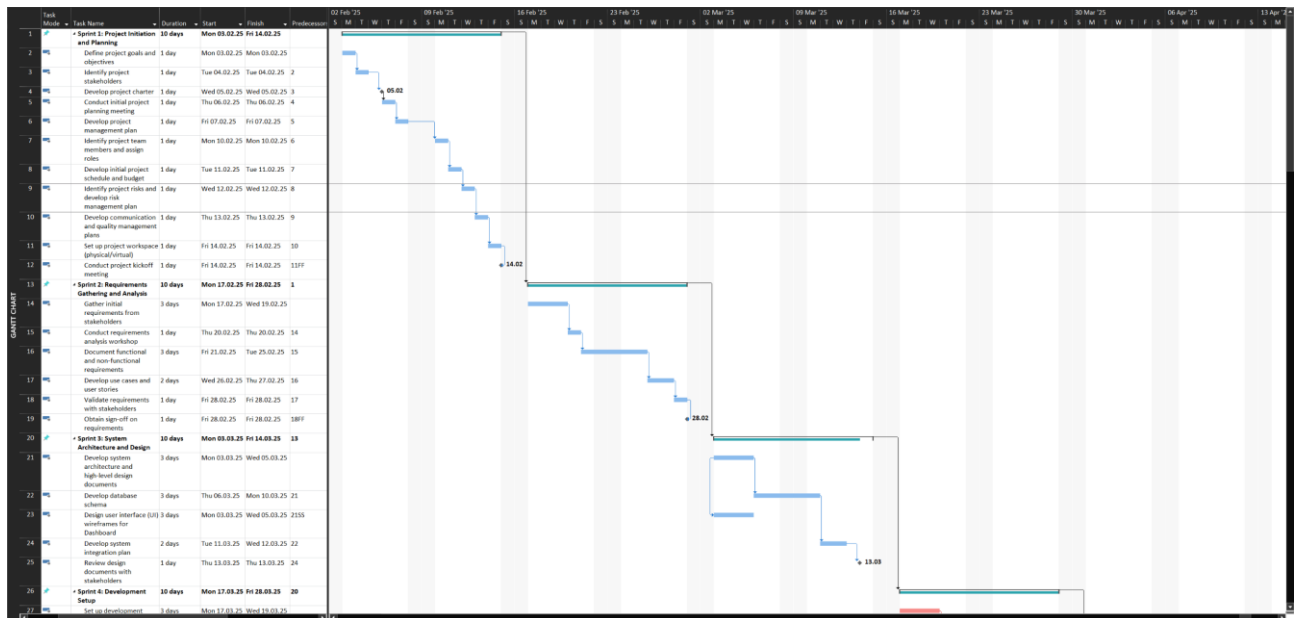


Рис. 4.3. Діаграма Ганта для проєкту в Microsoft Project (фрагмент)

Горизонтальна вісь діаграми позначає часову шкалу, поділену на робочі дні. Кожен рядок таблиці відповідає окремому завданню в межах спринтів. Кольорові смуги, які розташовані вздовж часової осі, візуалізують тривалість виконання спринтів. Стрілки на діаграмі вказують на взаємозв'язки між спринтами. Вони показують, що завершення одного етапу є передумовою для початку наступного.

Цей графік слугує зручним інструментом для контролю виконання проєкту, дозволяючи візуалізувати прогрес у режимі реального часу та

оперативно виявляти можливі затримки. Діаграма також сприяє ефективній комунікації між членами команди та іншими зацікавленими сторонами, забезпечуючи їм легкий доступ до плану проєкту.

У процесі роботи над плануванням термінів проєкту важливо також проаналізувати критичний шлях для визначення етапів, що мають найбільший вплив на строки.

Критичний шлях — це найдовший шлях у всій системі робіт сітьової моделі проєкту; якщо виконання робіт на цьому відрізку затримується, виконання всього проєкту затримується на такий же час [6].

Для більшості спринтів у проєкті було закладено невеликий резерв часу, що дозволяє зменшити ризики затримок у виконанні окремих завдань. Однак така стратегія мала технічний вплив на відображення критичного шляху у стандартних налаштуваннях Microsoft Project. Зокрема, за замовчуванням критичний шлях визначається лише за умов відсутності резерву часу (тобто резерв має дорівнювати 0 днів), через що жоден із спринтів спочатку не був підсвічений як критичний.

Щоб виправити це, було внесено коригування до налаштувань програми, а саме змінено параметр визначення критичного шляху, встановивши граничний резерв часу на рівні 1 дня. Це дозволило підсвітити спринти, що мають найменший запас часу, як частину критичного шляху (рис. 4.4).

Зокрема, до критичного шляху почали входити спринти, пов'язані безпосередньо з розробкою та релізом продукту, оскільки їхні часові резерви є мінімальними (див. Додаток В).

Натомість початкові етапи проєкту, такі як планування та аналіз вимог, не увійшли до критичного шляху через наявність резерву в 2 дні, що надає додаткову гнучкість для виконання цих задач.

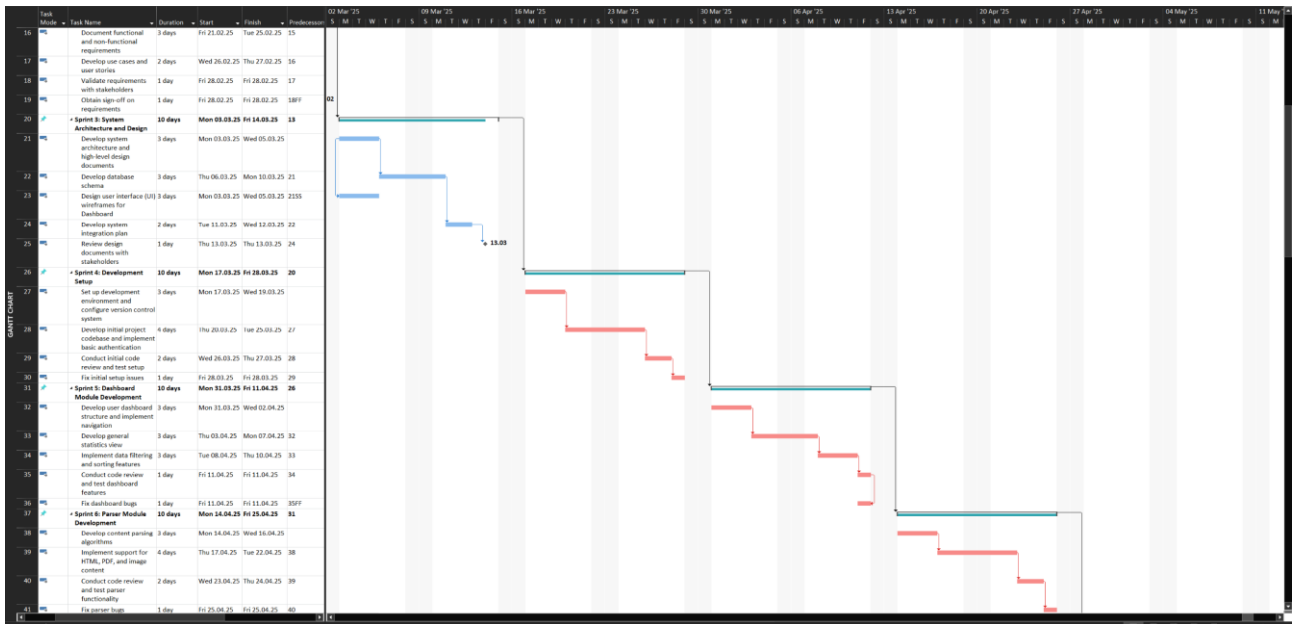


Рис 4.4. Критичний шлях в Microsoft Project (фрагмент)

Це коригування дозволило отримати реалістичніше відображення критичного шляху, що охоплює найбільш важливі та чутливі до часу етапи проекту, надаючи можливість командам зосередити свої зусилля саме на цих завданнях.

4.4 Планування ресурсів та бюджету проекту

Значною мірою вартість проекту визначають в основному вартістю ресурсів, необхідних для здійснення робіт у проекті. Планування ресурсів включає визначення того, які ресурси (людські, обладнання, матеріали, тощо) та у якій кількості мають бути задіяні для виконання проекту. Планування ресурсів має бути тісно пов'язано з оцінюванням вартості.

Після складання розкладу робіт у проекті можна побудувати графіки потреби у ресурсах, необхідних для виконання проекту.

Як було визначено на попередніх етапах проробки проекту, потрібні такі трудові ресурси:

- Проектний менеджер;
- Архітектор;

- Команда розробників (3 особи);
- UX/UI дизайнер;
- Тестувальник;
- Маркетолог;
- Спеціаліст підтримки.

У рамках реалізації даного проєкту передбачається робота команди у віддаленому режимі, що дозволяє значно скоротити витрати на забезпечення фізичної інфраструктури. Віддалений формат роботи має низку переваг, серед яких гнучкість у розподілі робочого часу, економія ресурсів та можливість залучення спеціалістів незалежно від їхнього географічного розташування.

Зокрема, проєкт не потребуватиме витрат на:

- Оренду офісу — відсутність необхідності утримання фізичного робочого простору дозволяє повністю виключити цю статтю витрат із бюджету.
- Комп'ютерне обладнання — співробітники працюватимуть з використанням власних пристроїв.
- Офісне приладдя — витрати на закупівлю меблів тощо не передбачені.

Основні фінансові ресурси будуть спрямовані на створення та підтримку інфраструктури, необхідної для віддаленої роботи команди, а саме:

- Хмарні сервіси — забезпечення зберігання даних, управління завданнями та комунікації між членами команди.
- Сторонні сервіси — спеціалізовані платформи та інструменти для співпраці, тестування та управління проєктом.
- Ліцензійне програмне забезпечення для розробки — це включає IDE (середовище розробки), інструменти для тестування, проєктування архітектури тощо.

Такий підхід забезпечує оптимальне використання бюджетних коштів, дозволяючи зосередити витрати на пріоритетних напрямках, пов'язаних безпосередньо з розробкою продукту. Віддалений формат роботи також сприяє

швидкій адаптації до сучасних умов, коли віртуальна співпраця стає стандартом для багатьох галузей.

Для ефективного планування ресурсів у проєкті використано інструменти Microsoft Project, що дозволили створити деталізований список усіх необхідних ресурсів (рис. 4.5). У цьому списку враховані людські та матеріально-технічні ресурси, необхідні для виконання завдань. Кожному ресурсу призначено ключові параметри, зокрема:

- Роль або назва ресурсу;
- Тип (матеріальний, трудовий);
- Група ресурсів (продуктові або бізнес);
- Кількість, доступна для проєкту;
- Погодинні ставки — стандартні та за наднормові.

	i	Resource Name	Type	Material	Initials	Group	Max.	Std. Rate	Ovt. Rate	Cost/Use	Accrue	Base
1		Architect	Work		A	Product	100%	730,00 €/hr	1 460,00 €/hr	0,00 €	Prorated	Standard
2		Programmer	Work		P	Product	300%	365,00 €/hr	730,00 €/hr	0,00 €	Prorated	Standard
3		UX/UI Designer	Work		U	Product	100%	300,00 €/hr	600,00 €/hr	0,00 €	Prorated	Standard
4		Project Manager	Work		P	Business	100%	300,00 €/hr	600,00 €/hr	0,00 €	Prorated	Standard
5		Tester	Work		T	Product	100%	200,00 €/hr	400,00 €/hr	0,00 €	Prorated	Standard
6		Marketer	Work		M	Business	100%	365,00 €/hr	730,00 €/hr	0,00 €	Prorated	Standard
7		Support Specialist	Work		S	Business	100%	100,00 €/hr	200,00 €/hr	0,00 €	Prorated	Standard
8		Infrastructure	Cost		I	Product					Prorated	
9		Marketing	Cost		M	Business					Prorated	
10		Contingency	Cost		C	Business					Prorated	

Рис. 4.5. Перелік ресурсів для виконання завдань у проєкті в MS Project

Розподіл ресурсів між задачами у проєкті було виконано з урахуванням часових рамок кожного завдання, їхніх взаємозв'язків та обмежень по доступності. На діаграмі Ганта, створеній у Microsoft Project, можна побачити, як ресурси призначені до виконання конкретних задач (рис. 4.6), що дозволяє оцінити їхню завантаженість та ефективність використання (див. Додаток В).

Цей підхід забезпечує прозорість у плануванні ресурсів і допомагає запобігти їх перевантаженню або простою.

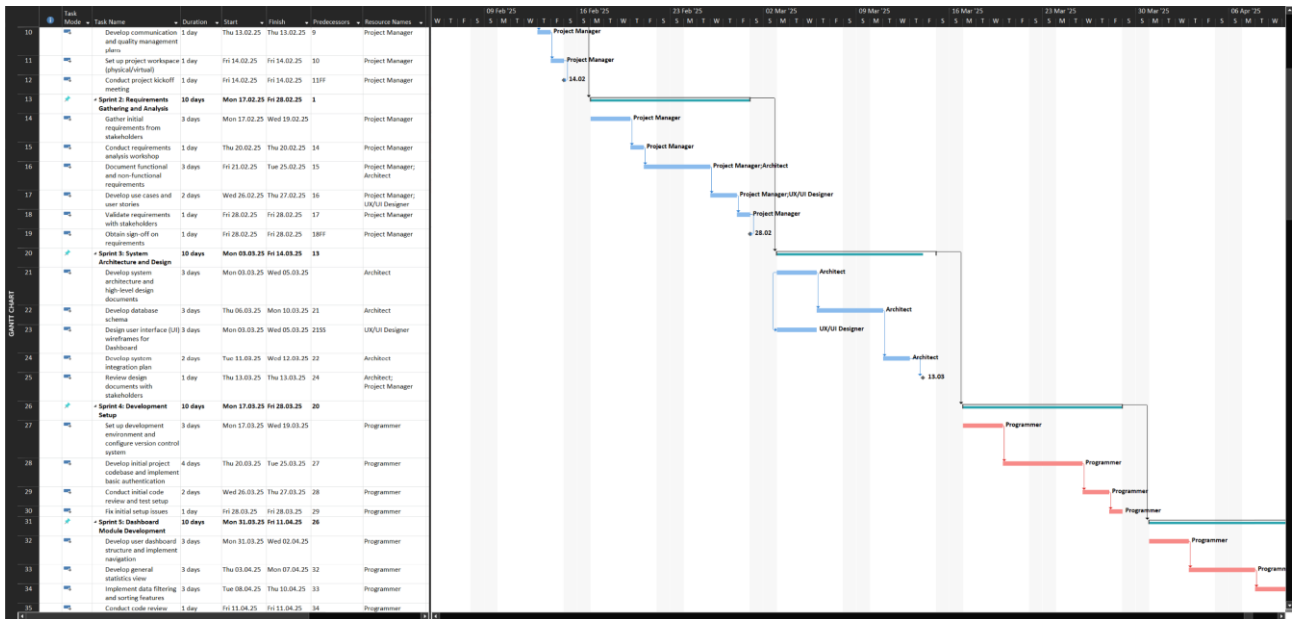


Рис. 4.6. Розподіл ресурсів по задачах проєкту в MS Project (фрагмент)

Після визначення потреби у ресурсах та аналізу їхнього розподілу між задачами проєкту було складено детальний бюджет, який охоплює всі ключові статті витрат (табл.4.3).

Таблиця 4.3

Бюджет та основні статті витрат проєкту

Статті витрат	Вартість, грн
<i>1</i>	<i>2</i>
Заробітна плата	1 077 000
Архітектор	262 800
Програмісти (3 особи)	262 800
Дизайнер UX/UI	108 000
Проектний менеджер	144 000
Тестувальник	72 000
Маркетолог	131 400
Спеціаліст підтримки	96 000
Інфраструктура	702 000
Хмарні сервіси	561 000

<i>1</i>	<i>2</i>
Сторонні сервіси	141 000
Маркетинг	294 000
Реклама та PR	144 000
Створення контенту та рекламних матеріалів	150 000
Неочікувані витрати	100 000
Резерв на непередбачені витрати	100 000
ЗАГАЛЬНИЙ БЮДЖЕТ ПРОЄКТУ	2 173 000

Бюджет проєкту — це план, який виражається в кількісних показниках і відображає витрати, необхідні для досягнення поставленої мети [21].

Бюджет включає витрати на оплату праці команди, інфраструктурні витрати, ліцензування програмного забезпечення та використання хмарних сервісів. Завдяки роботі у віддаленому режимі вдалося оптимізувати витрати, виключивши статті, пов'язані з орендою офісу, придбанням обладнання та забезпеченням офісного простору.

Якщо розглядати бюджет проєкту у розрізі етапів проєкту, то проєкт із створення програмного забезпечення для аудиту та аналізу маркетингового контенту поділяється на наступні етапи, кожен з яких характеризується власними задачами, строками виконання та бюджетом:

1. Етап дослідження

- Задачі: збір вимог, аналіз ринку та конкурентів.
- Спринт: 1-2
- Тривалість: 20 робочих днів.
- Бюджет: 50 000 грн. Основні витрати включають оплату роботи та інструментів дослідження.

2. Етап проектування

- Задачі: розробка архітектури та підготовка до розробки.
- Спринт: 3
- Тривалість: 10 робочих днів.
- Бюджет: 200 000 грн. Витрати пов'язані з роботою архітектора та використанням спеціалізованого ПЗ.

3. Етап розробки

- Задачі: написання коду.
- Спринт: 4-11
- Тривалість: 80 робочих днів.
- Бюджет: 1 100 000 грн. Основні витрати йдуть на оплату праці програмістів та інфраструктуру.

4. Етап тестування

- Задачі: проведення тестів та виправлення помилок.
- Спринт: 12-13
- Тривалість: 20 робочих днів.
- Бюджет: 150 000 грн. Витрати включають зарплату тестувальника та інструменти тестування.

5. Етап пілотування

- Задачі: запуск прототипу з обраними користувачами для отримання зворотного зв'язку.
- Спринт: 14
- Тривалість: 10 робочих днів.
- Бюджет: 100 000 грн. Основні витрати пов'язані з підтримкою пілотного запуску.

6. Етап впровадження

- Задачі: розгортання продукту та навчання користувачів, рекламні кампанії.
- Спринт: 15-16
- Тривалість: 20 робочих днів.

- Бюджет: 500 000 грн. Включає витрати на зарплати, маркетинг та підтримку.

7. Етап масштабування, маркетингу та продажів

- Задачі: розробка маркетингової стратегії, запуск продукту на ринок, оптимізація інфраструктури.
- Не входить в досліджений у роботі скоуп проекту і є наступною фазою його існування.
- Основні витрати будуть пов'язані з маркетингом і збільшенням технічної інфраструктури.

У цій розбивці закладено прогнозовані витрати у 2 млн грн. Враховуючи забюджетований резерв коштів на непередбачувані витрати у розмірі 100 000 грн можна зробити висновок, що даний розрахунок відповідає паспорту проекту.

Візуалізуємо також розподіл вартості проекту за ключовими етапами його реалізації. Візуалізація дозволяє наочно оцінити обсяги фінансових ресурсів, необхідних для кожного етапу, а також визначити найбільш витратні частини проекту (рис. 4.7).

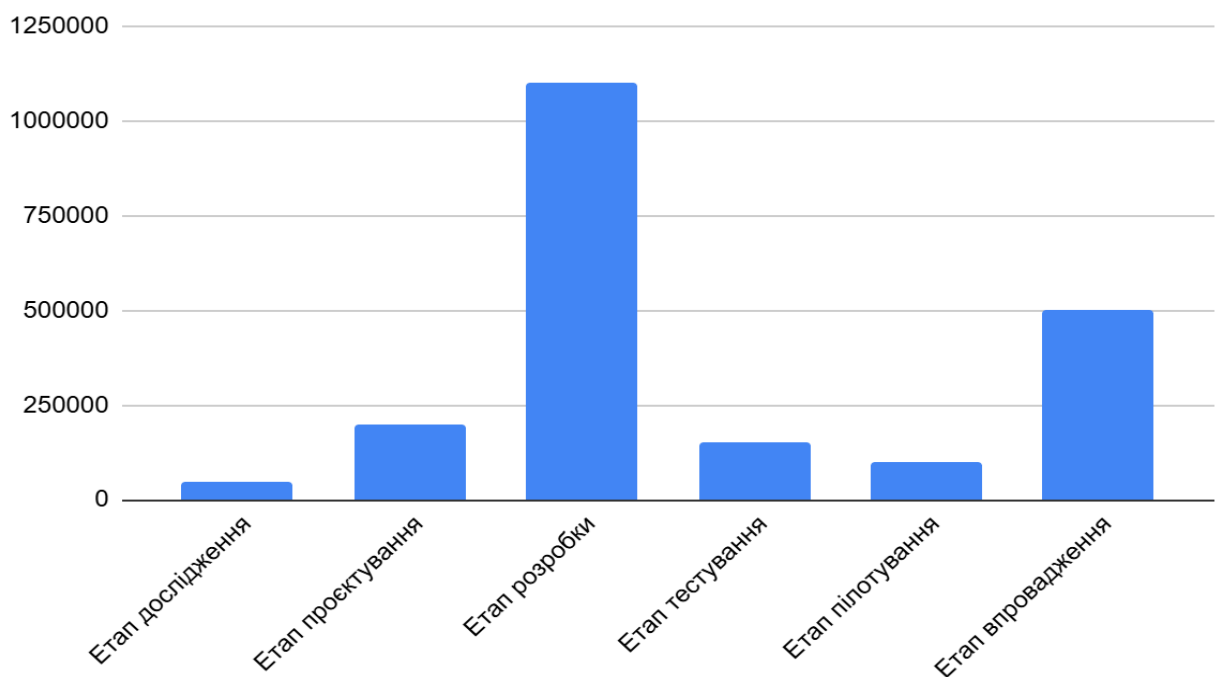


Рис. 4.7. Розподіл вартості за ключовими етапами

Графік ілюструє, що найбільша частка бюджету виділена на етапи розробки та впровадження, що відповідає їхній критичній ролі для успішного завершення проєкту. Інші етапи отримують пропорційно менший обсяг фінансування, але відіграють важливу роль у забезпеченні цілісності та якості кінцевого продукту.

Цей графік є ефективним інструментом для аналізу та оптимізації використання ресурсів, а також слугує основою для подальшого моніторингу фінансових показників проєкту.

Отже, у даному розділі було детально розглянуто планування ресурсів та бюджету проєкту із врахуванням його етапів та спринтів. Проведений аналіз дозволив визначити оптимальний розподіл фінансових та людських ресурсів для кожного етапу, зокрема дослідження, проєктування, розробки, тестування, впровадження та масштабування. Сформований бюджет забезпечує досягнення цілей проєкту в межах виділених ресурсів, а також дозволяє ефективно контролювати витрати на кожному етапі. Візуалізація бюджету сприяє кращому розумінню та управлінню фінансовими показниками, що є ключовим для успішної реалізації проєкту.

4.5 Ключові метрики ефективності у разі впровадження ПЗ

На етапі дослідження проєкту з розробки програмного забезпечення для аудиту та аналізу маркетингового контенту важливо визначити, які ключові показники ефективності (KPI) будуть використовуватися для оцінки успішності впровадження розробленого рішення.

KPI (Key Performance Indicators), або ключові показники ефективності, — це всі фінансові та нефінансові показники, які допомагають оцінити ефективність поточної діяльності. Їхнє значення, виміряне за певний період часу та виражене в цифрах, є зворотним зв'язком про те, чи була досягнута поставлена мета [15].

Основні функції KPI:

- Вимірювання продуктивності: KPI надають кількісні дані, що дозволяють оцінити ефективність роботи.
- Контроль і моніторинг: За допомогою KPI можна відстежувати прогрес і виявляти відхилення від плану.
- Мотивація: KPI можуть слугувати основою для мотиваційних систем, заохочуючи співробітників за досягнення цілей.
- Управлінські рішення: KPI допомагають керівникам приймати обґрунтовані рішення, базуючись на об'єктивних даних.
- Комунікація: KPI забезпечують ясність і прозорість у спілкуванні між керівництвом і співробітниками, допомагаючи зрозуміти пріоритети та очікування [29].

Існує кілька видів KPI:

- кількісні — вимірюють результати у величинах та числах (кількість продукції, частка ринку, обсяг продажів);
- якісні — оцінюють якість виконаної роботи (рівень задоволення клієнтів якістю обслуговування або продукту);
- стратегічні — спрямовані на оцінку досягнення стратегічних цілей організації (збільшення прибутку, покращення репутації);
- індивідуальні — використовуються для оцінки внеску окремих працівників чи команд за певний період до загальних цілей компанії [26].

Оскільки проєкт ще знаходиться на етапі планування, основою аналізу є прогнознi оцінки та припущення щодо результатів впровадження.

Для оцінки ефективності впровадження програмного забезпечення передбачається використання таких основних показників:

1. Рентабельність інвестицій (ROI).

Рентабельність інвестицій дозволяє визначити економічну доцільність розробки та впровадження програмного забезпечення. Цей показник відображає

співвідношення між вигодами, які компанія отримує завдяки впровадженню системи, та витратами на її розробку й інтеграцію.

ROI демонструє, наскільки проєкт є прибутковим у фінансовому плані. Високий ROI підтверджує, що інвестиції в розробку ПЗ виправдовують себе і сприяють економії або отриманню додаткових доходів.

Методика розрахунку ROI для користувачів ПЗ:

$$ROI = \frac{Ben - Exp}{Exp} * 100\% \quad (4.1)$$

де ROI — коефіцієнт рентабельності інвестицій;

Ben — очікувані вигоди від використання ПЗ;

Exp — загальні витрати на ПЗ.

Приклад вигод: економія на ручному аудиті, оптимізація маркетингових витрат, зростання доходів через ефективніший аналіз контенту.

2. Час на проведення аудиту

Цей показник оцінює, наскільки автоматизація процесів аналізу маркетингового контенту скорочує часові витрати порівняно з ручними методами.

Він дає змогу оцінити швидкість роботи системи та економію часу, що особливо важливо для масштабних проєктів або компаній із великими обсягами даних.

Скорочення часу аналізу дозволяє маркетинговим командам оперативніше приймати рішення та зменшувати час реакції на зміни ринку.

Методика вимірювання: порівняння часу, витраченого на аналіз певного обсягу контенту до і після впровадження ПЗ.

3. Точність аналізу

Метрика точності аналізу дозволяє оцінити, наскільки автоматизовані процеси знижують кількість помилок, спричинених людським фактором.

Висока точність аналізу забезпечує якісніші результати аудиту, зменшує ризики помилкових рішень та підвищує довіру до результатів роботи системи.

Уникнення помилок сприяє ефективнішому використанню ресурсів компанії.

Методика вимірювання: аналіз кількості помилок у даних до і після впровадження програмного забезпечення.

Відсоток зменшення помилок розраховується за формулою:

$$Acc = \frac{E_0 - E_1}{E_0} * 100\% \quad (4.2)$$

де Acc — коефіцієнт зниження помилок в аудиті (коефіцієнт точності);

E_0 — кількість виявлених помилок в аудиті до впровадження ПЗ;

E_1 — кількість виявлених помилок в аудиті після впровадження ПЗ;.

4. Продуктивність системи

Цей показник відображає кількість аналізованих об'єктів (наприклад, вебсторінок, зображень, текстів) за одиницю часу.

Підвищення продуктивності дає змогу обробляти більші обсяги контенту в обмежений час, що є критичним для великих організацій.

Висока продуктивність системи знижує потребу у додаткових ресурсах.

Методика вимірювання: розраховується як кількість оброблених об'єктів за годину/день/інший період часу.

5. Рівень задоволеності користувачів

Цей показник відображає, наскільки кінцеві користувачі задоволені функціональністю, інтерфейсом та результатами роботи програмного забезпечення.

Задоволені користувачі частіше використовують систему, що сприяє підвищенню її окупності та ефективності.

Низький рівень задоволеності може свідчити про потребу в доопрацюванні системи.

Методика вимірювання: проведення опитувань серед користувачів із використанням шкали оцінки (наприклад, від 1 до 5). Аналіз відгуків користувачів щодо функціональності, простоти використання та відповідності очікуванням.

Таким чином, на основі прогнозів можна припустити, що впровадження програмного забезпечення сприятиме значному підвищенню ефективності процесів аудиту та аналізу маркетингового контенту. Очікується, що економічна ефективність зросте за рахунок оптимізації витрат, оскільки автоматизація процесів дозволить зменшити потребу у трудових ресурсах і мінімізувати витрати, пов'язані з ручним аналізом.

Крім того, час, необхідний для проведення аналізу, значно скоротиться. Згідно з прогнозами, автоматизація дозволить зменшити часові витрати на 70% у порівнянні з традиційними ручними методами. Це забезпечить швидший доступ до результатів аналізу, що особливо важливо для прийняття оперативних рішень у динамічному середовищі.

Іншою важливою перевагою є очікуване зниження кількості помилок у даних. Завдяки автоматизації аналізу та виключенню людського фактору передбачається зменшення кількості помилок на 90%-100%. Це підвищить якість аналітичних звітів і знизить ризики, пов'язані з прийняттям рішень на основі некоректної інформації.

Продуктивність системи також має суттєво зрости. Автоматизовані процеси дозволять обробляти значно більший обсяг даних у коротші строки, що стане ключовою перевагою для компаній, які працюють з великими обсягами контенту.

Окрім цього, необхідно прагнути до постійного підвищення рівня задоволеності користувачів програмного забезпечення. Завдяки інтуїтивно зрозумілому інтерфейсу та широкому функціоналу кінцеві користувачі зможуть ефективно використовувати систему, отримуючи якісні результати з мінімальними зусиллями. Така зручність у використанні сприятиме активному впровадженню програмного забезпечення в робочі процеси.

На етапі впровадження програмного забезпечення критично важливо здійснити фактичну оцінку ключових показників ефективності (KPI), що дозволить точно виміряти успішність системи в реальних умовах експлуатації.

Ця оцінка має базуватися на даних, отриманих у процесі роботи системи, і повинна враховувати як технічні, так і функціональні аспекти її роботи.

Для досягнення цього необхідно розробити надійні механізми збору та обробки даних для кожного КРІ. Ці механізми мають забезпечувати автоматичний або напівавтоматичний збір даних, пов'язаних з показниками продуктивності, точності аналізу, часу обробки даних та рівня задоволеності користувачів. Наприклад, для збору даних про продуктивність система повинна автоматично фіксувати кількість оброблених об'єктів за певний період часу. Дані повинні бути структурованими, доступними для аналізу та відповідати вимогам до конфіденційності.

Щоб оцінити задоволеність користувачів, необхідно регулярно проводити опитування, спрямовані на отримання зворотного зв'язку від кінцевих користувачів програмного забезпечення. Це можуть бути як формалізовані анкети з чіткими питаннями про зручність використання, функціональність та відповідність очікуванням, так і відкриті форми для пропозицій та коментарів. Отримані відповіді допоможуть не лише оцінити поточний рівень задоволеності, але й виявити напрями для вдосконалення системи.

Таким чином, комплексний підхід до оцінки КРІ забезпечить точне розуміння ефективності програмного забезпечення після його впровадження і дозволить коригувати стратегію розвитку відповідно до потреб користувачів та бізнесу.

Аналіз ключових метрик ефективності впровадження програмного забезпечення демонструє, що автоматизація процесів аудиту та аналізу маркетингового контенту забезпечує значне підвищення продуктивності, точності та швидкості виконання завдань. Це сприяє зниженню трудовитрат і помилок, що, в свою чергу, дозволяє оптимізувати бізнес-процеси компанії та досягати стратегічних цілей з мінімальними витратами. Таким чином, запропоноване ПЗ створює вагомий внесок у покращення управлінських та операційних показників організації.

ВИСНОВКИ

У даній кваліфікаційній роботі магістра було здійснено дослідження моделей управління проектом створення та впровадження програмного забезпечення для аудиту та аналізу маркетингового контенту.

Актуальність теми дослідження обумовлена тим, що у сучасних умовах розвитку цифрових технологій та глобалізації інформаційних процесів маркетинговий контент стає стратегічним інструментом у досягненні бізнес-цілей. Ефективний аналіз маркетингового контенту не лише визначає конкурентоспроможність компанії, але й створює основу для прийняття стратегічно важливих рішень. Проте існуючі інструменти для аудиту та аналізу контенту мають суттєві обмеження, які ускладнюють роботу з великими обсягами даних та знижують оперативність і точність аналітики. Таким чином, розвиток цифрового маркетингу створює нові можливості для підприємств, але разом з тим висуває високі вимоги до ефективного аналізу маркетингового контенту.

Сучасні інструменти, такі як HubSpot, Google Search Console та Google Analytics, не забезпечують комплексного підходу до вирішення завдань аудиту та аналізу контенту. Їхня обмежена функціональність та складність інтеграції значно збільшують витрати часу та ресурсів. У результаті виникає потреба у створенні програмного забезпечення, яке дозволить автоматизувати ці процеси, забезпечуючи точність і швидкість аналізу.

У процесі роботи було досягнуто поставлену мету — розроблено концепцію унікального програмного забезпечення для аудиту та аналізу маркетингового контенту, яке забезпечить автоматизацію процесів збору, обробки та аналізу даних з використанням сучасних технологій, включаючи штучний інтелект та інтеграцію з ключовими аналітичними платформами (Google Analytics, HubSpot тощо).

Для цього було виконано всі поставлені завдання, а саме:

- 1) Проведено дослідження предметної області. Вивчено сучасні підходи до аудиту контенту та наявні на ринку рішення для його автоматизації, такі як MonsterInsights, Analytify, Zapier, Supermetrics. Вивчено їхні можливості та обмеження.
- 2) Обґрунтовано доцільність проєкту. Визначені проблеми, які виникають при використанні існуючих рішень, а саме:
 - a. Фрагментація функціональності — існуючі інструменти, що збирають дані, необхідні для аудиту та аналізу контенту, такі як HubSpot, Google Search Console та Google Analytics, не дають цілісної картини ефективності контенту. Для цього необхідно вручну збирати та об'єднувати дані з цих систем.
 - b. Складність та ресурсоемність процесів — використання декількох систем вимагає значних витрат часу та ресурсів, зокрема для інтеграції даних, їхньої обробки та формування звітів.
 - c. Ризик помилок — ручні процеси створюють додаткові ризики втрати даних, дублювання інформації або некоректного аналізу.
 - d. Обмеженість у прогнозуванні — більшість інструментів не пропонують глибинної аналітики та можливості прогнозування змін трафіку, не використовують можливості ШІ для аналізу.

Таким чином, було виявлено потребу в рішенні, що забезпечить комплексний підхід до автоматизації процесів аудиту та аналізу контенту.

- 3) Розроблено концепцію проєкту — сформульоване технічне завдання у вигляді паспорту проєкту та розроблено концептуальну модель інформаційної системи, моделі баз даних, основні алгоритми та вайрфрейми екранів інтерфейсу.
- 4) Формалізовано математичні моделі – створено 5 математичних моделей для аналізу маркетингових даних: парсингу вмісту сайту, інтеграції з аналітичними інструментами, оновлення даних сайту, визначення цільової аудиторії сторінки, прогнозування трафіку.

- 5) Здійснено планування проєкту розробки програмного забезпечення. Було розроблено організаційну структуру команди проєкту з 7 ролей та 9 осіб, сформовано матрицю розподілу відповідальності між ролями, створено ієрархічну структуру робіт (WBS), складено календарний план реалізації проєкту, відповідно до якого його тривалість складе 160 робочих днів (або 7,5 календарних місяців).
- 6) Сформульовано економічне обґрунтування проєкту через проведення аналізу витрат та оцінку економічної ефективності. Так, за бюджету у 2173000 грн, коефіцієнт рентабельності інвестицій (ROI) складатиме 24.13%, що свідчить про рентабельність розробки даного рішення. Описано прогнозований вплив запропонованого рішення на ефективність бізнес-процесів через розрахунок на прикладі бази практики — компанія може економити 9972.5 грн щомісяця.
- 7) Наведено рекомендації щодо впровадження для подальшої реалізації проєкту та інтеграції розробленого ПЗ в роботу організацій. Наведено 5 ключових показників ефективності, за допомогою яких можна оцінювати успішність впровадження даного рішення.

Отримані результати дослідження демонструють, що запропоноване програмне забезпечення є перспективним інструментом для автоматизації процесів аудиту та аналізу маркетингового контенту.

Розроблені моделі та рекомендації можуть бути застосовані не лише у сфері маркетингу, але й у суміжних галузях, таких як електронна комерція, аналітика соціальних мереж, управління проєктами та бізнес-аналітика. Це підкреслює масштабність і міждисциплінарність запропонованого підходу.

Робота також має значний науковий внесок, оскільки включає формалізацію нових математичних та концептуальної моделей, адаптованих до специфіки маркетингового аналізу.

Отже, виконане дослідження відповідає поставленим цілям, а результати цієї роботи можуть стати основою для розробки нових рішень, які будуть сприяти підвищенню ефективності бізнес-процесів у цифрову епоху.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Математичне моделювання в ІТ проєктах: Методичні вказівки для виконання практичних, лабораторних та самостійних робіт з навчальної дисципліни / Морозов В.В., Коломієць Г.С. — К.: КНУ імені Тараса Шевченка, 2023. — 64 с.
2. Методи розробки концепцій ІТ проєктів: Методичні вказівки для виконання практичних, лабораторних та самостійних робіт / Морозов В.В. — К. : КНУ імені Тараса Шевченка, 2022. — 76 с.
3. Методика проведення практичних занять з елементами тренінгу з курсу «Соціальне проєктування та соціальна інженерія»: навчально-методичний посібник. / І. І. Ігнат'єва. Одеса: Фенікс, 2021, 28 с.
4. Сучасні інструментальні засоби розробки користувацького інтерфейсу: Опорний конспект лекцій / Шевчук Р.П. — Тернопіль, 2012. — 103 с.
5. Технології управління даними та знаннями [Текст]: Методичні вказівки для практичних та лабораторних робіт / А.О. Хлевний, О.В. Єгорченков — К. : ВИДАВНИЦТВО, 2020. — 54 с.
6. Управління проєктами: процеси планування проєктних дій [Текст]: підручник / І.В. Чумаченко, В.В. Морозов, Н.В. Доценко, А.М. Чередніченко. — К. : КНУ імені Тараса Шевченка, 2016. — 673 с.
7. Project Management: A Systems Approach to Planning, Scheduling, and Controlling / Kerzner, H. — Wiley, 2017. — 880 p.
8. Аутсорсинг. Що таке аутсорсинг? Визначення та найкращі практики // Режим доступу: <https://peopleforce.io/uk/hr-glossary/outsourcing>
9. Від проблеми до проєкту : Навчальний семінар-тренінг // Режим доступу: https://learn.ztu.edu.ua/pluginfile.php/327142/mod_resource/content/1/2%20Від%20проблеми%20до%20проєкту.pdf
10. Вільчавський І. Бізнес-аналітики і нефункціональні вимоги в agile розробці // Режим доступу: <https://www.ba.in.ua/2023/08/18/biznes-analityky-i-nefunkczionalni-vymogy-v-agile-rozrobci/>

11. Гнучка методологія розробки ПЗ — Agile // Режим доступу:
<https://training.gatestlab.com/blog/technical-articles/flexible-software-development-methodology-agile/>
12. Зосим М. Блок-схема (Flowchart) // Режим доступу:
<https://www.maxzosim.com/blok-skhema/>
13. Зосим М. Моделювання даних (Data Modelling) // Режим доступу:
<https://www.maxzosim.com/data-modelling/>
14. Зосим М. Структура розбиття робіт (Work Breakdown Structure — WBS) // Режим доступу: <https://www.maxzosim.com/struktura-rozbittia-robit/>
15. Ключовий показник ефективності (KPI) // Режим доступу:
<https://peopleforce.io/uk/hr-glossary/key-performance-indicator>
16. Кузнєцова А. Marketing Qualified Lead (MQL) — як визначити, залучити та перетворити на постійного клієнта // Режим доступу: <https://web-promo.ua/ua/blog/marketing-qualified-lead-mql-yak-viznachiti-zaluchiti-ta-peretvoriti-na-postijnogo-kliyenta/>
17. Настанова до зводу знань з управління проєктами (PMBOK v.7) // Режим доступу: <https://pmiukraine.org/pmbok7>
18. Організаційна структура проєкту (Organizational Breakdown Structure, OBS) // Режим доступу: <https://pmb.com.ua/uk/slovar-terminov/organizatsionnaya-struktura-organizational-breakdown-structure-obs-instrument/>
19. Основи алгоритмізації : Блок-схеми алгоритму // Режим доступу:
<https://yevshan.com.ua/info/006/content/content3.html>
20. Основні етапи проєктування баз даних // Режим доступу:
<https://javarush.com/ua/quests/lectures/ua.questhibernate.level17.lecture01>
21. Поняття та порядок складання проєктного бюджету // Режим доступу:
<https://buklib.net/books/25595/>
22. Популярні життєві цикли розробки ПЗ // Режим доступу:
<https://training.gatestlab.com/blog/technical-articles/popular-software-development-life-cycles/>

23. Технологічні стеки // Режим доступу: <https://infdev.com.ua/docs/web-development/technology-stacks/>
24. Філімонов С., Андреев М. Що таке CI/CD, як він працює та коли знадобиться на проєкті. Лайфхаки та bad practices // Режим доступу: <https://highload.today/uk/blogs/shho-take-ci-cd-yak-vin-pratsyuye-ta-koli-znadobitsya-na-proyekti-lajfhaki-ta-bad-practices/>
25. Що таке вайрфрейм (wireframe) та для чого він потрібен у процесі розробки сайту чи додатку // Режим доступу: <https://goldwebsolutions.com/uk/blog/shho-take-vajrfrejwm-wireframe-ta-dlya-chogo-vin-potriben-u-protsesi-rozrobki-sajtu-chi-dodatku/>
26. Що таке ключові показники ефективності КРІ та як впровадити систему в своїй компанії: 7 простих кроків // Режим доступу: <https://krupkin.pro/uk/knowledge/finansova-model-viddiluprodzahiv/kpi-klyuchovi-pokazniki-efektivnosti-ta-yih-vpliv-na-biznes/>
27. Що таке контейнеризація та в чому її переваги // Режим доступу: <https://gigacloud.ua/blog/navchannja/chomu-kontejneri-ce-majbutne-virtualizacii>
28. Що таке чиста поточна вартість (NPV)? // Режим доступу: <https://finances.in.ua/shcho-take-chysta-potochna-vartist-npv/>
29. Що таке КРІ (ключові показники ефективності)? // Режим доступу: <https://hurma.work/blog/shho-take-kpi-klyuchovi-pokazniki-efektivnosti/>
30. Що таке Open Authorization // Режим доступу: <https://foxminded.ua/oauth-shcho-tse/>
31. Що таке PostgreSQL? Вступ, переваги та недоліки // Режим доступу: <https://www.guru99.com/uk/introduction-postgresql.html>
32. Як організувати структуру підприємства: актуальні види і формати // Режим доступу: <https://online.novaposhta.education/blog/yak-organizuvati-strukturu-pidpriyemstva-aktualni-vidi-i-formati>
33. Як розрахувати рентабельність інвестицій у інструменти керування вимогами // Режим доступу: <https://visuresolutions.com/uk/посібник-з->

[відстеження-управління-вимогами/як-розрахувати-управління-вимогами-до-ROI/](#)

34. A Guide to the Business Analysis Body of Knowledge (BABOK v.3) // Режим доступу: <https://www.scribd.com/document/338083331/BABOK-3-ONLINE-A-Guide-to-the-Business-Analysis-Body-of-Knowledge>
35. Ahrefs Plans & Pricing // Режим доступу: <https://ahrefs.com/pricing>
36. Bradon Matthews. What Is a RACI Matrix? Definition, Examples, Uses // Режим доступу: <https://project-management.com/understanding-responsibility-assignment-matrix-raci-matrix/>
37. Content marketing Market Size, Share, Competitive Landscape and Trend Analysis Report by Channel Type, by End Use : Global Opportunity Analysis and Industry Forecast, 2023-2032 // Режим доступу: <https://www.alliedmarketresearch.com/content-marketing-market-A68748>
38. Flori Needle. How To Price A SaaS Product // Режим доступу: <https://blog.hubspot.com/sales/price-saas-product>
39. HubSpot's Free Business and Marketing Resources // Режим доступу: <https://www.hubspot.com/resources>
40. Kayla Carmicheal. How to Run A Content Audit in 2024 // Режим доступу: <https://blog.hubspot.com/marketing/company-content-audit>
41. Lucie Simonova. A practical guide on conducting a successful content audit // Режим доступу: <https://kontent.ai/blog/a-practical-guide-on-conducting-a-successful-content-audit/>
42. Michael Jung. Unleashing the Power of AI: A Definitive Guide to AI-Driven Content Audits // Режим доступу: <https://www.clearvoice.com/resources/ai-driven-content-audits/>
43. Screaming Frog SEO Spider // Режим доступу: <https://www.screamingfrog.co.uk/seo-spider/>
44. SemRush Plans & Pricing // Режим доступу: <https://www.semrush.com/pricing/>

45. Supermetrics. Maximize your marketing returns with better data // Режим доступа: <https://supermetrics.com/>
46. The State of Content Marketing 2023 Global Report // Режим доступа: <https://lp.semrush.com/rs/519-IIY-869/images/StateofContentMarketing2023.pdf>
47. Tomas Laurinavicius. Top 10 Content Audit Tools: Best Software for 2024 // Режим доступа: <https://marketful.com/blog/content-audit-tools/>
48. Tushar Pol. The Ultimate Guide on How to Do a Content Audit (+ Template) // Режим доступа: <https://www.semrush.com/blog/content-audit/>
49. What are the most common challenges in a content audit? // Режим доступа: <https://www.linkedin.com/advice/1/what-most-common-challenges-content-audit-skills-content-creation-18t8e>
50. Zapier for Marketing // Режим доступа: <https://zapier.com/solutions/marketing>

ДОДАТКИ

Додаток А

Функціональні та нефункціональні вимоги до ПЗ

ФУНКЦІОНАЛЬНІ ВИМОГИ

- Платформа:
 - Можливість реєстрації користувачів і створення організації.
 - Налаштування двофакторної аутентифікації.
 - Можливість входу в акаунт (авторизація, аутентифікація), зміни паролю, видалення організації.
 - Можливість додавати користувачів до організації з різними правами доступу: на перегляд або на внесення змін.
 - Можливість управління підпискою: зміна тарифного плану, скасування підписки.
- Автоматичний парсинг контенту:
 - Система повинна мати здатність автоматично сканувати та парсити контент вебсайтів.
 - Здатність ідентифікувати та збирати різні типи контенту, такі як текст, зображення, відео тощо.
 - Можливість збирати мета-дані, такі як заголовки, мета-опис, ключові слова тощо.
- Дешборд (інтерфейс користувача):
 - Можливість відображення зібраного контенту у вигляді таблиці з розширеними фільтрами та сортуванням.
 - Здатність відображати метрики та статистику з Google Analytics та HubSpot.
 - Можливість вибору періоду для відображення агрегованих даних з Google Analytics та HubSpot.
- Редагування та анотування даних:

- Можливість для користувачів додавати власні колонки для анотування контенту.
- Здатність редагувати та оновлювати дані про контент в системі.
- Інтеграція з сторонніми інструментами:
 - Інтеграція з Google Analytics та HubSpot для автоматичного отримання та аналізу додаткової інформації про ефективність контенту.
 - Інтеграція з платіжним провайдером для реалізації функціональності підписки.
 - Інтеграція з AI для розпізнавання тематики, цільової аудиторії кожної сторінки на основі аналізу її змісту та надання короткого опису кожної сторінки з урахуванням відповідних характеристик, таких як мета, ключові слова, структура тощо.
- Система сповіщень:
 - Можливість налаштування сповіщень про важливі події, такі як падіння трафіку на певних сторінках або досягнення певних метрик.
 - Можливість створення сповіщень за визначеними користувачем формулами.
 - Здатність надсилати сповіщення користувачам через електронну пошту або месенджери (наприклад, про завершення парсингу або важливі події).
- Автоматичне оновлення даних:
 - Можливість налаштування періодичності парсингу даних.
 - Можливість налаштування періодичності оновлення даних з Google Analytics та HubSpot.
 - Здатність автоматично оновлювати дані з Google Analytics та HubSpot для забезпечення актуальності аналітичної інформації.

НЕФУНКЦІОНАЛЬНІ ВИМОГИ

- Розробка системи як хмарного рішення з можливістю доступу через веббраузер з будь-якого пристрою.
- Зручний та інтуїтивно зрозумілий інтерфейс.
- Забезпечення безпеки та конфіденційності даних користувачів.
- Збереження та організація всіх даних, зібраних під час аудиту контенту.
- Можливість інтеграції з іншими зовнішніми системами аналітики чи управління контентом за допомогою API.

Паспорт проєкту

Назва проєкту	Проєкт створення та впровадження програмного забезпечення для аудиту та аналізу маркетингового контенту “Contenta”.
Ознаки проєкту	<ol style="list-style-type: none"> 1) Унікальність — проєкт передбачає створення унікального програмного забезпечення з використанням специфічних технологій та інтеграції з існуючими аналітичними інструментами та штучним інтелектом. 2) Просування вперед, цільова діяльність — проєкт спрямовано на підвищення ефективності маркетингових стратегій шляхом автоматизації аудиту та аналізу контенту. 3) Конкретні дії — передбачаються конкретні роботи з проєктування, розробки, впровадження рішення та управління проєктом. 4) Обмеженість у часі — розробка і впровадження відбудуться протягом визначеного терміну (12 місяців). 5) Обмежені ресурси — реалізація проєкту має відбуватись в рамках визначеного бюджету (2 173 000 грн) та з використанням обмежених ресурсів. 6) Командна робота — виконання проєкту вимагає залучення міждисциплінарної команди спеціалістів (проєктний менеджер, розробники, дизайнери, маркетологи і т.д.). 7) Результат проєкту — розроблене та впроваджене програмне забезпечення для аудиту маркетингового контенту, яке може самостійно функціонувати.
Цілі проєкту	<ol style="list-style-type: none"> 1) Автоматизувати процес збору та аналізу даних з вебсайтів для оцінки ефективності контенту. 2) Розробити інтуїтивно зрозумілий інтерфейс для

	<p>ефективного контент-аудиту.</p> <p>3) Інтегрувати програмне забезпечення з ключовими аналітичними платформами (Google Analytics, HubSpot) та штучним інтелектом.</p> <p>4) Надати користувачам можливість кастомізації звітів за допомогою власних колонок та коментарів.</p> <p>5) Забезпечити користувачів інструментом для моніторингу змін трафіку та оптимізації контенту на основі аналітики.</p>
<p>Зацікавлені сторони проекту</p>	<p>1) Команда проекту — інтерес в реалізації унікального продукту, професійному зростанні.</p> <p>2) Контент-маркетологи — полегшення процесу аудиту контенту, підвищення ефективності роботи.</p> <p>3) Власники вебсайтів — отримання детального аналізу ефективності контенту, оптимізація маркетингових стратегій.</p> <p>4) Інвестори — повернення інвестицій через продаж підписок на SaaS-рішення.</p> <p>5) Розробники ПЗ — виклик у реалізації технологічно складного продукту.</p> <p>6) Маркетингові агентства — розширення інструментарію для обслуговування клієнтів.</p>
<p>Задачі проекту</p>	<p>1) Дослідити потреби ринку та визначити ключові функції продукту.</p> <p>2) Розробити технічне завдання для створення програмного забезпечення.</p> <p>3) Вибрати технологічний стек та розробити продукт.</p> <p>4) Інтегрувати програмне забезпечення з аналітичними платформами та штучним інтелектом.</p>

	<p>5) Розробити алгоритми для автоматизованого збору та аналізу даних.</p> <p>6) Протестувати продукт і зібрати зворотній зв'язок від бета-тестерів.</p> <p>7) Запустити маркетингову кампанію для просування продукту.</p> <p>8) Налаштувати процеси підтримки та оновлення програмного забезпечення.</p> <p>9) Розробити план масштабування продукту.</p>
<p>Результати проекту</p>	<p>1) Розроблене та впроваджене програмне забезпечення для аудиту маркетингового контенту.</p> <p>2) Здійснена інтеграція з ключовими аналітичними інструментами та штучним інтелектом.</p> <p>3) Покращено ефективність контент-маркетологів через автоматизацію аудиту.</p> <p>4) Збільшено кількість клієнтів серед контент-маркетологів і власників сайтів.</p> <p>5) Отримано позитивні відгуки від користувачів програми.</p>
<p>Продукт проекту</p>	<p>1) Цільова група — контент-маркетологи, SEO-спеціалісти, власники вебсайтів.</p> <p>2) Потреби — автоматизація аудиту та аналізу контенту, підвищення ефективності маркетингових стратегій.</p> <p>3) Продукт — хмарне SaaS рішення для аудиту та аналізу маркетингового контенту з інтеграцією до аналітичних інструментів та штучного інтелекту.</p> <p>4) Опис цінності — значне покращення процесу аудиту контенту, економія часу та ресурсів для контент-маркетологів, підвищення ROI маркетингових кампаній.</p>

Оцінка вартості проєкту	Загальна сума трудових ресурсів: 1 077 000 грн Інфраструктура: 702 000 грн Маркетинг: 294 000 грн Резерв на непередбачені витрати: 100 000 грн Усього: 2 173 000 грн
--	--

Діаграма Ганта для проекту в Microsoft Project з відображенням критичного шляху та розподілу ресурсів по задачах

