

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГЬЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему:

«РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ АНАЛІТИКИ
РЕКЛАМНИХ КАМПАНІЙ З ВИКОРИСТАННЯМ ХМАРНИХ
ТЕХНОЛОГІЙ»

Виконав: здобувач 6 курсу групи Іт-62
спеціальності 126 «Інформаційні системи
та технології» Ковалевич М.Ю.

(Прізвище та ініціали)

Керівник: Желєзняк А.М.

(Прізвище та ініціали)

ДУБЛЯНИ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Рівень вищої освіти другий (магістерський)

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

д.т.н., професор, Тригуба А. М.

(вч. звання, прізвище, ініціали)

“ _____ ” _____ 2025 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Ковалевич Михайло Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка інформаційної системи для аналітики рекламних кампаній з використанням хмарних технологій»

керівник роботи к. е н., доцент., Желєзняк А.М.

(наук.ступінь, вч. звання, прізвище, ініціали)

затверджені наказом ЛНУВМ ім.С.З.Гжицького №140/к-с від 28.02.2025

2. Строк подання студентом роботи 08.12.2025 р.

3. Вихідні дані: аналітичні дані роботи та характеристика об'єкту дослідження, опис технології, науково-технічна і довідкова література.

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)
Вступ

1. Теоретичні засади та постановка задачі

2. Вибір та обґрунтування технологій

3. Практична реалізація та результати

4. Охорона праці

5. Визначення ефективності

Висновки

Бібліографічний список

5. Перелік графічного матеріалу Графічний матеріал подається у вигляді презентації

6. Консультанти розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3, 5	Желєзняк А.М., доцент кафедри інформаційних технологій			
4	Городецький І.М., доцент кафедри інженерної механіки			

7. Дата видачі завдання 01.03.2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Відмітка про виконання
1	Отримання завдання. Вивчення літератури по темі роботи. Написання першого розділу	01.03.2025 – 31.05.2025	
2	Проектування та опис технічного завдання, обґрунтування та вибір інструментарію реалізації проекту (написання другого розділу).	01.06.2025 – 31.08.2025	
3	Програмна реалізація поставленого завдання (написання третього розділу)	01.09.2025 – 17.10.2025	
4	Написання розділу з охорони праці	18.10.2025 – 04.11.2025	
5	Оцінка ефективності поставленого завдання (виконання п'ятого розділу)	05.11.2025 – 18.11.2025	
6	Завершення оформлення основної частини, написання висновків та підготовка презентаційного матеріалу	19.11.2025 – 01.12.2025	
7	Завершення роботи в цілому. Підготовка до захисту кваліфікаційної роботи	02.12.2025 – 08.12.2025	

Студент

(підпис)

(прізвище та ініціали)

Керівник роботи

(підпис)

Желєзняк А.М.

(прізвище та ініціали)

УДК 004.04:659.1]:004.7

Розробка інформаційної системи для аналітики рекламних кампаній з використанням хмарних технологій

Ковалевич М.Ю. Кафедра ІТ - Дубляни, ЛНУВМБ ім. С.З. Гжицького, 2025.

Кваліфікаційна робота: 74 текст. 5 част., 16 рис., 5 таб., 58 джерел.

Наведено особливості хмарних технологій, що застосовуються для зберігання, обробки та аналітики даних рекламних кампаній. Проведено огляд сервісів Google BigQuery, Apache Spark на AWS EMR, Google App Engine та систем оркестрації контейнерів Docker + Kubernetes (GKE, EKS, AKS). Сформульовано методи інтеграції та забезпечення масштабованості аналітичної системи з використанням хмарної інфраструктури. Об'єкти дослідження – хмарні сервіси та платформи, що дозволяють забезпечити високу швидкодію та доступність аналітики великих обсягів даних у реальному часі.

Метою кваліфікаційної роботи є розробка інформаційної системи для аналітики рекламних кампаній на основі хмарних технологій із використанням сучасних сервісів зберігання даних, обробки великих обсягів та контейнеризації. Окреслено архітектуру системи та процеси безперервної інтеграції й доставки (CI/CD) із використанням Docker і Kubernetes. Розглянуто можливості розгортання веб-додатків у хмарному середовищі та інтеграції з аналітичними інструментами.

Подано особливості реалізації інформаційної системи, приклади використання хмарних сервісів у реальних сценаріях та оцінку ефективності її функціонування. Розроблено заходи із охорони праці та безпеки у надзвичайних ситуаціях під час експлуатації системи. Визначено критерії продуктивності, масштабованості та економічної доцільності використаних технологій.

Ключові слова: аналітика, хмарні технології, BigQuery, Apache Spark, Kubernetes, рекламні кампанії, інформаційна система..

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ ТА ПОСТАНОВКА ЗАДАЧІ	7
1.1 Аналіз існуючих підходів до аналітики рекламних кампаній	7
1.2 Хмарні технології у зберіганні та обробці даних	13
1.3 Взаємодія хмарних сервісів із базами даних	15
1.4 Формулювання завдання на основі обраної моделі реалізації.....	22
РОЗДІЛ 2. ВИБІРТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ.....	25
2.1 Огляд можливих технологій та сервісів.....	25
2.2 Обґрунтування вибору архітектури та інструментів	28
2.3 Реалізація ключових компонентів системи	32
2.4 Застосування хмарних сервісів у реалізації проекту	36
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ.....	45
3.1 Опис функціональності для кінцевих користувачів	45
3.2 Можливості інтеграції з іншими системами (API, інтерфейси)	48
3.3 Аналіз отриманих результатів і перспективи використання	51
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	53
4.1 Обґрунтування можливих чинників виникнення травмонебезпечних ситуацій	53
4.2 Умови та фактори виникнення травмонебезпечних ситуацій та їхні наслідки.....	55
4.3 Безпека в умовах надзвичайних ситуацій	56
РОЗДІЛ 5 ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ.....	58
ВИСНОВКИ	60
ДОДАТКИ	67

ВСТУП

У сучасних умовах стрімкого розвитку цифрових технологій ефективне управління рекламними кампаніями стало одним із ключових чинників успішності бізнесу. Велика кількість рекламних каналів, різноманітність форматів та необхідність швидкої адаптації до ринкових змін вимагають від компаній впровадження інструментів, здатних забезпечити глибоку, оперативну та наочну аналітику рекламної діяльності. У цьому контексті інформаційні системи для аналітики відіграють важливу роль, надаючи можливість автоматизованого збору, обробки та інтерпретації даних про ефективність рекламних кампаній.

Однією з ключових технологій, яка відкриває нові можливості для таких систем, є хмарні обчислення. Хмарні технології забезпечують масштабованість, доступність, надійність та економічність, що робить їх ідеальним середовищем для розробки сучасних аналітичних рішень. На основі хмарної інфраструктури можливо створити інформаційну систему, яка забезпечує централізований доступ до даних, інтеграцію з різними рекламними платформами, візуалізацію показників і прийняття обґрунтованих управлінських рішень. Використовуючи такий підхід також варто зазначити, що використання SQL баз даних надалі буде мати місце та відігравати важливу роль.

Незважаючи на наявність багатьох готових рішень у сфері аналітики реклами, більшість з них або орієнтовані на специфічні задачі, або є занадто складними для використання в умовах малого чи середнього бізнесу. Також варто зазначити, що значна кількість рекламних площадок мають свої платформи і не виходять за їхні рамки. Як наслідок інтегрована аналітика цих платформ орієнтована на їхню інформаційну модель, що може створювати колізії у порівнянні ефективності між тою чи іншою площадкою. Це створює

потребу у розробці нових, більш адаптивних та індивідуалізованих інформаційних систем.

Магістерську кваліфікаційну роботу присвячено розробці теоретичної моделі інформаційної системи для аналітики рекламних кампаній із використанням хмарних технологій на основі аналізу вже існуючих рекламних площадок та аналітичних рішень. Для досягнення цієї мети передбачено вивчення наявних аналітичних та рекламних платформ, визначення їх сильних і слабких сторін, а також побудова власного підходу до реалізації функціоналу системи з урахуванням сучасних вимог до ефективності, зручності та масштабованості.

Результати кваліфікаційної роботи апробовані на студентській конференції, що підтверджується публікацією тез «Розробка інформаційної системи для аналітики рекламних кампаній із використанням хмарних технологій»[53]

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ЗАСАДИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз існуючих підходів до аналітики рекламних кампаній

Аналітика рекламних кампаній — це процес збору, обробки та інтерпретації даних з метою оцінки ефективності маркетингових заходів і прийняття обґрунтованих рішень. З розвитком цифрових технологій і хмарних сервісів цей процес еволюціонував від простих статистичних методів до складних інтелектуальних систем. В ході виконання кваліфікаційної роботи було розглянуто основні підходи та методи аналітики рекламних компаній, що використовуються на практиці.

Класичні методи базуються на використанні простих метрик і ручного аналізу результатів. Розглянемо більш детально основні інструменти класичних методів аналітики рекламних кампаній, які використовуються не перший рік.

SWOT-аналіз — це метод стратегічного планування, який дозволяє оцінити чотири ключові аспекти об'єкта дослідження. Такі як:

- ✓ S (Strengths) — сильні сторони: внутрішні переваги, які забезпечують успіх (наприклад, унікальні ресурси, досвід, технології).
- ✓ W (Weaknesses) — слабкі сторони: внутрішні недоліки, що можуть заважати розвитку (наприклад, нестача фінансів, слабкий бренд).
- ✓ (Opportunities) — можливості: зовнішні чинники, які можна використати на користь (наприклад, ринкові тренди, нові технології).
- ✓ T (Threats) — загрози: зовнішні ризики, які можуть зашкодити (наприклад, конкуренція, зміни в законодавстві).

SWOT-аналіз допомагає приймати обґрунтовані рішення щодо стратегії розвитку компанії, продукту чи проєкту [14].

У сфері цифрового маркетингу ключові показники ефективності (KPI) дозволяють оцінити успішність рекламних кампаній. До найважливіших із них належать CTR, CPC і ROI:

✓ CTR (Click-Through Rate) — показник клікабельності, що визначає співвідношення кількості кліків до кількості показів реклами. Високий CTR свідчить про зацікавленість аудиторії в оголошенні [15, с.145].

✓ CPC (Cost Per Click) — ціна за клік, яка показує, скільки коштує один перехід користувача за рекламним оголошенням. Цей показник важливий для планування бюджету та визначення вартості залучення потенційного клієнта [15, с. 148].

✓ ROI (Return on Investment) — рентабельність інвестицій, що вимірює прибутковість кампанії щодо витрат. $ROI = (\text{Прибуток} - \text{Витрати}) / \text{Витрати} \times 100\%$. Позитивне значення ROI свідчить про доцільність вкладень у рекламу [16].

Маркетингова воронка — це модель, яка описує шлях потенційного клієнта від знайомства з продуктом до здійснення покупки. Вона поділяється на етапи:

- ✓ Увага (Awareness) – користувач дізнається про продукт або бренд.
- ✓ Інтерес (Interest) – з'являється зацікавленість у пропозиції.
- ✓ Бажання (Desire) – формується потреба або бажання придбати.
- ✓ Дія (Action) – клієнт здійснює покупку або цільову дію.

Ця модель допомагає маркетологам ефективно планувати рекламу, сегментувати аудиторію та підвищувати конверсію [17, с. 126].

Медіапланування — це процес стратегічного підбору та розміщення рекламних повідомлень у засобах масової інформації (інтернет, ТБ, радіо, преса тощо) для досягнення цільової аудиторії у визначений час і з

оптимальним бюджетом. Основна мета — забезпечити максимальну ефективність реклами при мінімальних витратах.

Процес медіапланування включає:

- ✓ визначення цільової аудиторії,
- ✓ вибір медіа каналів,
- ✓ розрахунок охоплення, частоти, GRP або Gross Rating Point (сумарний рейтинг),
- ✓ розподіл бюджету,
- ✓ оцінку ефективності кампанії.

Медіапланування є ключовим етапом у реалізації маркетингових стратегій, оскільки впливає на видимість бренду та результати рекламної кампанії [18, с. 247; 19, с. 89]. До сильних сторін медіапланування можна віднести простота впровадження, не потребують складних інструментів, підходять для малого бізнесу та локальних кампаній. Слабкими сторонами використання підходу є обмежена точність і масштабованість, низька інтеграція з цифровими каналами, але високий рівень суб'єктивізму у висновках.

Як приклад застосування можна розглянути просте підприємство яке аналізує кампанію за кількістю показів, кліків та конверсій, фіксуючи результати в Excel. Такий підхід простий, але не враховує складні взаємозв'язки між каналами та поведінку користувача [1, с. 49].

Тепер розглянемо більш сучасні методи які базуються на класичних та користуються популярністю у сфері хмарових обчислень та соціальних мережах, а саме цифрова аналітика та Big Data, хмарні аналітичні рішення, а також інновації такі як інтелектуальні системи аналітики (AI/ML)

Цифрова аналітика та Big Data базується на зборі великих обсягів даних з веб сайтів, соціальних мереж, рекламних платформ та їх обробці за

допомогою спеціалізованих інструментів. Розглянемо основні характеристики на основі таблиці 1.1.

Таблиця 1.1. — Основні характеристики цифрової аналітики та Big Data

Основні інструменти	Сильні сторони	Слабкі сторони
Інструменти збору інформації з веб сайтів: Google Analytics, Adobe Analytics	Збір даних у реальному часі	Складність налаштування
Інструменти для аналітики у соц мережах: Facebook Insights, LinkedIn Campaign Manager	Деталізоване сегментування аудиторії	Потреба в технічних навичках
Інструменти для аналітики у соц мережах: Facebook Insights, LinkedIn Campaign Manager	Автоматизація звітності та візуалізацій	Залежність від сторонніх сервісів

Як приклади можна розглянути компанію яка використовує Google Analytics 4 для оцінки поведінки користувачів на сайті: тривалість сеансу, глибина перегляду, джерело трафіку. Для кампаній у соцмережах — Facebook Ads Manager, де аналізуються охоплення, частота, CPC та демографія аудиторії [4; 5].

Хмарні аналітичні рішення у свій час стали інноваційними технологіями які вийшли на новий рівень — інтеграція різних джерел

даних, масштабування, висока доступність. Розглянемо основні характеристики на основі таблиці 1.2.

Таблиця 1.2. — Основні характеристики хмарних та аналітичних рішень.

Основні інструменти	Сильні сторони	Слабкі сторони
Google BigQuery, AWS Redshift, Azure Synapse	Масштабованість і швидка обробка великих обсягів даних	Підписка або плата за використання ресурсів
Power BI, Tableau Cloud, Google Looker	Гнучка інтеграція з іншими системами (CRM, ERP)	Необхідність знань у галузі хмарних технологій
Хмарні API (Google Ads, Facebook Marketing API)	Висока доступність для користувачів з різних локацій	Можливі ризики безпеки даних

Агентство використовує Google BigQuery для зберігання історичних даних з кампаній, Power BI для побудови дашбордів ефективності та Zapier для автоматичного оновлення звітів через API [9; 8, с. 214].

Інтелектуальні системи аналітики (AI/ML) представляє найсучасніший підхід передбачає використання штучного інтелекту та машинного навчання для прогнозової аналітики, автоматичного розпізнавання патернів у даних та прийняття рішень. Ці системи відрізняються можливостями від попередніх можливостями прогнозування та рекомендаціями. Розглянемо можливі напрямки та слабкі та сильні сторони на прикладі таблиці 1.3.

Таблиця 1.3. — Основні напрямки та сильні і слабкі сторони інтелектуальні системи аналітики.

Основні напрямки	Сильні сторони	Слабкі сторони
Прогнозування ефективності рекламних кампаній	Висока точність прогнозів	Висока складність впровадження
Рекомендації щодо бюджетного розподілу	Автоматизація оптимізації	Необхідність великих обсягів даних
Персоналізація повідомлень і аудиторій	Підтримка складних моделей поведінки користувачів	Потреба в кваліфікованих аналітиках і дата-сайєнтистах

Рекламна платформа Meta Ads автоматично використовує AI для визначення найефективнішої креативної стратегії для заданої аудиторії. Компанія може застосовувати моделі Prophet або XGBoost для прогнозування доходу від кампаній на основі історичних даних [11; 12].

Сучасна аналітика рекламних кампаній потребує використання гібридного підходу: класичні методи залишаються актуальними для стратегічного аналізу, а цифрові, хмарні та інтелектуальні — для оперативного моніторингу та гнучкого реагування. Використання хмарних рішень забезпечує масштабованість, ефективність і автоматизацію процесів аналітики, що є особливо важливим у середовищі з великою кількістю даних і каналів комунікації.

1.2 Хмарні технології у зберіганні та обробці даних

У контексті аналізу рекламних кампаній обробка великих обсягів даних має вирішальне значення, а хмарні технології надають для цього комплекс необхідних інструментів. По-перше, хмара пропонує економічно ефективну, масштабовану та гнучку інфраструктуру для зберігання даних без необхідності інвестувати в локальні серверні ресурси [20].

Хмарні платформи — як Amazon Web Services, Google Cloud та Microsoft Azure — підтримують зберігання структурованих, напівструктурованих і неструктурованих даних через об'єктні сховища, сховища даних (data warehouses) та озера даних (data lakes) [21; 22]. Наприклад, платформа Google BigQuery — це серверлес-сховище з можливістю аналізу петабайт даних за допомогою SQL, а також вбудованими ML-здатностями [22].

Для обробки в реальному часі та пакетної обробки широко використовуються такі технології, як Apache Hadoop і Spark, що підтримуються хмарною інфраструктурою [23; 24]. Amazon Kinesis дозволяє обробляти та аналізувати стрімінгові дані в реальному часі, що особливо корисно для рекламних систем [25]. В Azure рішення типу Stream Analytics дозволяють писати SQL-подібні запити для обробки даних у реальному часі з підтримкою інтеграції машинного навчання [26].

Традиційна модель ETL поступається сучасному ELT, коли дані спочатку завантажуються у хмару, а трансформації виконуються вже в сховищі. Це дозволяє працювати з різними форматами даних і виконувати обробку тільки там, де це дійсно потрібно [22].

Хмарна архітектура дозволяє використовувати принцип pay-as-you-go, що знижує капітальні витрати і запобігає утриманню зайвої інфраструктури [22; 27]. Моделі Analytics-as-a-Service (AaaS) дають змогу

інтегрувати, трансформувати і візуалізувати дані без суттєвих інвестицій у власні обчислювальні потужності [22].

Розглянемо приклад реалізації інтерфейсу користувача на основі AaaS на прикладі рисунку 1.1. Є багато варіантів реалізації серверної частини, а от інтерфейс користувача у більшості випадків однотипний, відображає один або декілька графіків і данні у вигляді таблиці.

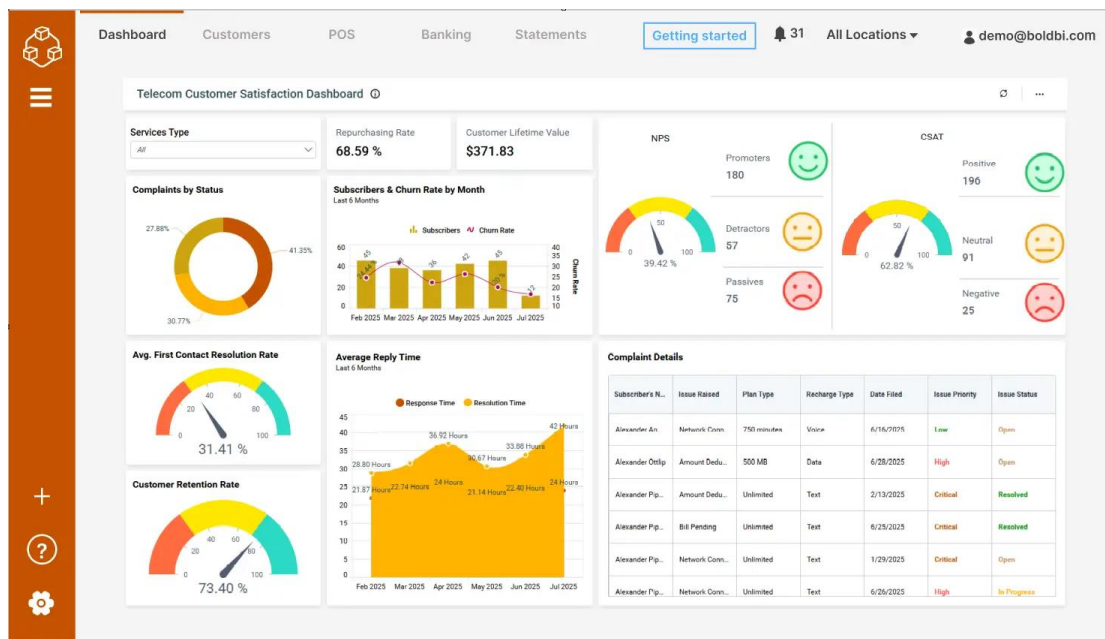


Рисунок 1.1. - Відображає приклад реалізації AaaS [54]

Інтерфейс відображає велику кількість аналітичних даних за малий період часу детально та за значно більший період часу на графіках, але із самого інтерфейсу користувача відсутня чіткість вибірки.

Хмарні провайдери впроваджують сучасні механізми безпеки: шифрування даних, багатофакторну аутентифікацію, аудит [28; 20]. Хмарні сервіси забезпечують доступність з будь-якої точки світу, сприяючи колаборації аналітиків та маркетологів без географічних обмежень [28; 27].

Сучасні хмарні сервіси підтримують інтеграцію AI/ML: BigQuery має вбудовані ML-функціональності, Azure Stream Analytics — інтеграцію з

Azure ML [22; 26]. Інструменти бізнес-аналітики, як-от Tableau або Power BI, дозволяють створювати інтерактивні дашборди та звіти [24; 29].

Інтеграція хмарних та локальних рішень (гібридні та мультихмарні архітектури) допомагає уникнути прив'язки до одного провайдера і оптимізувати продуктивність [30; 31]. Очікується зростання використання серверлес-обчислень і AI/ML-інтеграцій для ще більш гнучкої аналітики [30].

1.3 Взаємодія хмарних сервісів із базами даних

Інтеграція хмарних сервісів із реляційними базами даних є фундаментальним елементом сучасних аналітичних систем, зокрема у сфері маркетингової аналітики, де обробка великих обсягів транзакційних та поведінкових даних відбувається у режимі, близькому до реального часу. Модель Database-as-a-Service (DBaaS) забезпечує можливість розгортання повноцінних SQL-сховищ у хмарній інфраструктурі з переданням адміністративних функцій постачальнику сервісу. У такій архітектурі користувач взаємодіє з базою даних через уніфікований API або стандартні драйвери, тоді як фізичне зберігання, балансування навантаження, шифрування та відновлення після збоїв повністю контролює хмарний провайдер [32; 33, с. 1–2].

Дослідження *Relational Cloud* розглядає архітектуру DBaaS як багаторівневу систему, що складається з рівня клієнтських запитів, логічного рівня оптимізації та маршрутизації, шару управління безпекою та фізичного рівня зберігання. На рівні оптимізації відбувається розподіл запитів між вузлами за принципом шерінгу, що дозволяє рівномірно розподіляти навантаження. Шифрування даних реалізовано таким чином, що сервери зберігання не мають доступу до відкритого вмісту, а виконання SQL-операцій можливе над зашифрованими наборами даних, включаючи

складні агрегати та сортування [33, с. 1–2]. Це особливо актуально у сфері рекламної аналітики, де дані користувачів мають високий рівень конфіденційності.

У гібридних архітектурах, які поєднують публічну та приватну хмару, ефективність роботи із SQL-базами досягається за рахунок застосування Object-Relational Mapping (ORM). Цей підхід дозволяє програмним компонентам працювати з об'єктами, які автоматично відображаються на реляційні таблиці, незалежно від того, де фізично розташовані дані — у локальному центрі обробки чи у віддаленій хмарі. Використання ORM у гібридному середовищі полегшує створення складних аналітичних моделей, зокрема моделей прогнозування ефективності рекламних кампаній, де необхідно оперативно комбінувати історичні дані з новими потоками подій [34, с. 61–64].

Інтеграція SQL- та NoSQL-рішень у межах одного хмарного середовища є окремим викликом. Як показує систематичний аналіз Wisal Khan та ін., головною проблемою є відсутність єдиного стандарту взаємодії між цими двома парадигмами та складність міграції даних між різними DBaaS-платформами. SQL-сховища демонструють кращу узгодженість транзакцій та відповідність ACID-властивостям, тоді як NoSQL підходить для зберігання великого обсягу напівструктурованих даних, наприклад, даних веб-трекінгу або логів поведінки користувачів у рекламних кампаніях [35, с. 2–3]. У практичних реалізаціях для побудови єдиного аналітичного контуру застосовують проміжні інтеграційні сервіси, які транслюють запити SQL у формати, зрозумілі NoSQL-сховищам, і навпаки. Варто розглянути декілька основних відмінностей між NoSQL і SQL базами даних таблиця 1.4. На прикладі якої можна зрозуміти що для доступу до даних побудованих на основі шаблону зручніше використовувати SQL бази,

а для перегляду рекламних об'єктів із медіа даними краще використовувати NOSQL бази.

Таблиця 1.4. — Відмінності між NoSQL і SQL [55].

Параметр	SQL	NOSQL
Дизайн для	Традиційна RDBMS використовує синтаксис і запити SQL для аналізу та отримання даних для подальшого аналізу. Вони використовуються для систем OLAP.	Система баз даних NoSQL складається з різних технологій баз даних. Ці бази даних були розроблені у відповідь на вимоги, пред'явлені для розробки сучасної програми.
Мова запитів	Мова структурованих запитів (SQL)	Немає декларативної мови запитів
Тип	Бази даних SQL є табличними базами даних	Бази даних NoSQL можуть базуватися на документах, парах ключ-значення, графічних базах даних
Схема	Бази даних SQL мають попередньо визначену схему	Бази даних NoSQL використовують динамічну схему для неструктурованих даних.
Найкраще підходить для	Ідеальний вибір для середовища інтенсивного виконання складних запитів.	Це погано підходить для складних запитів.
Найпопулярніші компанії, які використовують	Hootsuite, CircleCI, Датчики	Airbnb, Uber, Kickstarter
Приклади	Oracle, Postgres і MS-SQL	MongoDB, Redis, Neo4j, Cassandra, Hbase.

З технічної точки зору інтеграція хмарних сервісів із SQL-БД для аналітики рекламних кампаній часто реалізується за схемою, де клієнтський застосунок (веб-панель чи мобільний інтерфейс) надсилає запит на API аналітичного модуля, який у свою чергу взаємодіє з DBaaS. Хмарна СУБД обробляє запит, використовуючи попередньо налаштовані індекси, матеріалізовані подання та системи кешування для прискорення відповіді. Результат передається у модуль візуалізації, що генерує графіки, звіти або прогностичні моделі. Для великих обсягів даних часто використовується розподілена обробка з поділом аналітичного навантаження між кількома вузлами, що забезпечує масштабованість системи та мінімізує затримки при генерації результатів.

При розробці таких систем особливу увагу приділяють питанням безпеки та відповідності нормативним вимогам. Навіть за наявності шифрування і багаторівневої автентифікації існує ризик компрометації даних через уразливості у застосунках або неналежне налаштування доступу. Тому на практиці впроваджують додаткові заходи, такі як сегментація мережі, контроль цілісності даних та періодичний аудит безпеки [33, с. 1; 35, с. 2].

Таким чином, інтеграція хмарних сервісів із SQL-базами даних у контексті аналітики рекламних кампаній є комплексним процесом, що вимагає балансування між продуктивністю, масштабованістю, безпекою та сумісністю з іншими технологіями. Використання DBaaS-моделі спрощує адміністрування та підвищує ефективність, ORM забезпечує гнучкий і стандартизований доступ до даних у гібридних середовищах, а уніфіковані API та інтеграційні сервіси дозволяють об'єднати різноманітні сховища в єдиний аналітичний комплекс.

Розглянемо приклад експорту звіту даних рекламних кампаній більш детально. Ця функція присутня у кожній платформі, із підтримкою CSV формату, а також кожен продукт намагається додати щось нове. У більшості випадків це ті ж самі дані які зображені у вікні браузера, але у більшому обсязі. Якщо у браузері присутня певна, обмежена кількість даних, то для експорту використовується значно більша кількість даних.

Розглянемо цей приклад на основі на прикладі інформаційної системи Realize. Мала кількість даних яку показує браузер рисунок 1.2. На перший погляд, для звичайного користувача це виглядає як щось незрозуміле і чому саме так. Відповісти на це питання дуже просто. Навіть якщо браузер зможе завантажити та зобразити тисячі стрічок, то для цього знадобиться значно більше часу ніж для зображення 20 – 50 стрічок у вигляді таблички. Якщо не врахувати що користувачі не люблять чекати, то навіть не кожен пристрій зможе обробити та зобразити табличку на 10 – 20 тисяч стрічок, а доступність на усіх можливих пристроях це важливо. Крім того людина не зможе аналізувати настільки великий об'єм даних, який навіть не вміститься ні в один дисплей Також варто зазначити, що завантаження великої кількості даних для кожного клієнта, яких одночасно бути тисячі, може стати неможливою задачею для бази даних або сервера, що в свою чергу може зупинити роботу усієї системи або окремих її серверів.

Рисунок 1.2. - Відображає приклад користувацького інтерфейсу [56].

Вище ми розглянули як швидко можна зобразити малу кількість даних, але користувачу потрібно мати доступ до всіх його даних за певних проміжків часу. Це створює потребу у наданні можливості перегляду великої кількості даних малими порціями і це питання обхідно вирішити. Розглянемо більше детально частину попереднього рисунку на Рисунок 1.3. Ми бачимо «1 – 50 of 6648» — це означає, що на сторінці зображено 50 рядків з 6648. По бокам від кількості рядків є кнопки, які дозволяють переглядати наступну або попередню сторінку або одразу переміститись на першу або останню сторінку у звіті.

0	\$0.000
552	\$0.094

1 - 50 of 6648

Рисунок 1.3. - Пагінація даних [56].

Розглянемо більш детально можливість експорту на прикладі Realize. На рисунку 1.4 можна побачити, що система має обмеження у 100 тисяч рядків та у разі перевищення цього показника пропонує, звузити часові рамки або видалити стовпці які впливають на загальну кількість рядків. Це чітко демонструє, що об'єм даних який можна експортувати є більшим ніж можна побачити на одній сторінці, а точніше у дві тисячі разів.

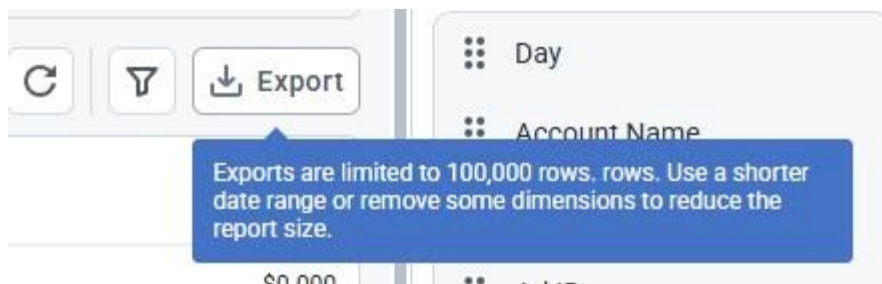


Рисунок 1.4. - Відображає приклад максимального ліміту даних для експорту у CSV файл [56]

Експорт даних із великою кількістю даних — це важливо, для того щоб бізнес аналітики могли аналізувати данні та на основі цього робити висновки у зручних для них програмах. Кім цього це може знадобитись для порівняльного аналізу різних рекламних кампаній або будь, що інше, що не може виконати веб інтерфейсу додатка. Отже ми отримаємо вимогу надати користувачу можливість експортувати велику кількість даних із мінімальних завантаженням систем та з максимально можливою швидкістю. Розглянемо це на прикладі схеми на рисунку 1.5.

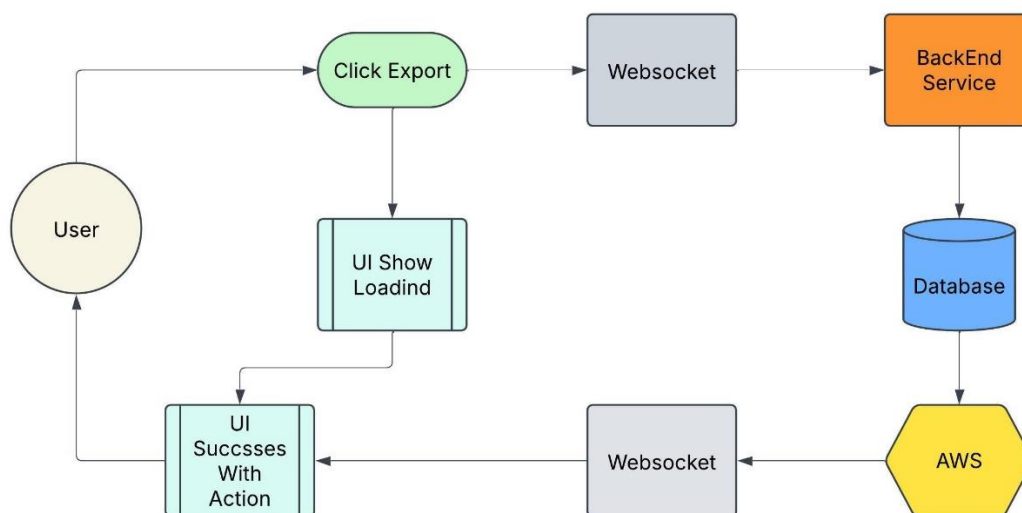


Рисунок 1.5 - Схема експорту звіту.

Може здаватись, що AWS у цій схемі зайвий, але це не так. Одна із важливих можливостей AWS S3 cloud storage це можливість використання архівованих даних у форматі gzip це дозволяє зменшити кількість потрібного трафік у для передачі даних (Додаток В). Сучасна коректна побудова інтерфейсу користувача на основі REST API основана на патерні із застосуванням проміжної API(gateway). Через це виникає велике навантаження на систему у випадку передачі даних, тому делегація цього процесу хмарі, зменшить навантаження на систему та позитивно вплине на швидкість та продуктивність системи.

1.4 Формулювання завдання на основі обраної моделі реалізації

У сучасних умовах цифрової економіки ефективність рекламних кампаній безпосередньо залежить від можливостей підприємства здійснювати оперативний збір, зберігання та аналітику великих обсягів даних, що генеруються різними рекламними каналами. Для вирішення цього завдання необхідним є створення інформаційної системи, здатної інтегрувати дані з різнорідних джерел (соціальні мережі, пошукові системи,

медійна реклама, e-mail маркетинг тощо), забезпечити їх централізоване зберігання у хмарному середовищі та надати інструменти для подальшої обробки і візуалізації.

На основі обраної моделі реалізації, яка передбачає використання хмарних технологій як основи для зберігання та обробки даних, сформульовано завдання магістерської роботи.

Загальне завдання полягає у розробці теоретичної моделі інформаційної системи для аналітики рекламних кампаній, яка використовує можливості хмарних сервісів для забезпечення масштабованості, гнучкості та економічної ефективності обробки даних. Для досягнення цієї мети необхідно розв'язати виконати певні кроки, а саме:

- ✓ Аналіз предметної області – дослідити сучасні підходи до управління рекламними кампаніями, визначити ключові показники ефективності (CTR, CPC, ROI, конверсії) та специфіку їх обліку.

- ✓ Формування вимог до системи – розробити функціональні та нефункціональні вимоги, зокрема до збору даних, інтеграції з рекламними платформами, зберігання, обробки, візуалізації та захисту інформації.

- ✓ Розробка концептуальної моделі системи – описати логічну архітектуру, визначити основні модулі (модуль збору даних, сховище, модуль аналітики, модуль візуалізації результатів).

- ✓ Вибір інструментальних засобів – обґрунтувати доцільність застосування конкретних хмарних рішень (наприклад, Google Cloud, AWS або Azure) для реалізації основних функцій системи.

- ✓ Створення прототипу моделі системи – запропонувати структурно-функціональну схему, яка відображає взаємодію між компонентами та демонструє можливість реалізації поставлених завдань.

✓ Оцінка ефективності моделі – визначити потенційні показники продуктивності, масштабованості та надійності розробленої системи, а також порівняти їх із традиційними підходами.

Таким чином, завдання кваліфікаційної роботи полягає у обґрунтуванні цілісної теоретичної моделі інформаційної системи для аналітики рекламних кампаній з використанням хмарних технологій, яка забезпечить інтеграцію даних, їх ефективне зберігання, аналітичну обробку та підтримку прийняття управлінських рішень у сфері маркетингу.

РОЗДІЛ 2.

ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ

2.1 Огляд можливих технологій та сервісів

Розробка інформаційної системи для аналітики рекламних кампаній передбачає використання сучасних хмарних технологій та сервісів, які забезпечують ефективність у зборі, зберіганні та обробці даних. Для побудови такої системи доцільно розглянути можливості провідних хмарних платформ та інструментів, орієнтованих на аналітику великих даних та маркетингову аналітику.

Розглянемо хмарні платформи загального призначення.

Найбільш поширеними є глобальні провайдери хмарних рішень:

✓ Amazon Web Services (AWS) – надає широкий набір сервісів для роботи з даними: Amazon S3 для зберігання, Amazon Redshift для аналітики, AWS Glue для інтеграції та підготовки даних, Amazon QuickSight для візуалізації результатів [36].

✓ Microsoft Azure – включає Azure Data Lake для масштабованого сховища, Azure Synapse Analytics для комплексної обробки та аналітики, Power BI для побудови інтерактивних звітів та дашбордів [37].

✓ Google Cloud Platform (GCP) – забезпечує сервіси BigQuery для високопродуктивного аналізу даних, Cloud Storage для надійного зберігання, Data Studio для візуалізації та Pub/Sub для потокової обробки інформації [38].

Ці платформи відзначаються масштабованістю, високою доступністю та широким набором інтеграцій, що дозволяє створювати системи аналітики практично будь-якої складності.

Для збору маркетингових даних із різних джерел використовуються різноманітні інструменти збору та інтеграції даних, до яких можна віднести наступні спеціалізовані сервіси:

- ✓ Google Analytics 4 та Google Marketing Platform – для збору даних про поведінку користувачів у цифрових каналах [39];
- ✓ Facebook Ads Manager, Google Ads, LinkedIn Ads – джерела даних про ефективність рекламних кампаній у соціальних мережах та пошукових системах [40];
- ✓ Інтеграційні інструменти на кшталт Zapier, Integromat, Kafka (Make) або Apache NiFi – забезпечують передачу даних між системами без необхідності складного програмування.

Актуальним питанням є вибір технології обробки та аналізу даних. У системах для аналітики рекламних кампаній актуальним є використання як традиційних систем керування базами даних, так і спеціалізованих інструментів:

- ✓ Реляційні СУБД (MySQL, PostgreSQL, MS SQL Server) – для зберігання структурованих даних [42], (Додаток Г);
- ✓ Системи обробки великих даних – Apache Spark, Hadoop, які дозволяють здійснювати розподілену обробку великих масивів інформації [43];
- ✓ Моделі машинного навчання (TensorFlow, Scikit-learn, Azure ML) – для прогнозування ефективності рекламних кампаній та побудови рекомендацій [44].

Візуалізація є ключовим етапом у процесі аналізу рекламних кампаній. Серед найбільш популярних інструментів та засобів аналітики та візуалізації даних є:

- ✓ Tableau – професійний інструмент для бізнес-аналітики з широкими можливостями інтеграції [45];

✓ Google Data Studio / Looker Studio – зручний безкоштовний сервіс для інтерактивних дашбордів [46]. Рис. 2.1;

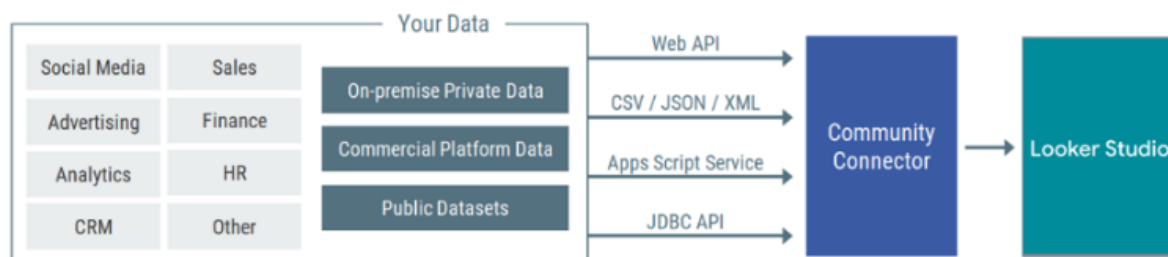


Рисунок 2.1 - Приклад інтеграції Google Looker Studio.

✓ Power BI – продукт Microsoft, який тісно інтегрується з Azure та Office 365, що робить його зручним для корпоративних користувачів [47].

Системи керування базами даних для зберігання та аналітики.

Вибір СУБД є критичним для ефективності інформаційної системи, оскільки саме база даних визначає можливості обробки великих масивів інформації. Для завдань Big Data доцільно використовувати:

✓ Google BigQuery – аналітична СУБД, що працює у хмарі та дозволяє швидко виконувати SQL-запити до терабайтів даних [48];

✓ Amazon Redshift – сховище даних від AWS, орієнтоване на виконання складних аналітичних запитів [36];

✓ Azure Synapse Analytics – рішення від Microsoft, яке поєднує можливості сховища даних та інструментів для аналітики великих обсягів інформації [37];

✓ Snowflake – хмарна платформа для зберігання та аналітики даних, яка підтримує масштабування та високу продуктивність [49].

Для невеликих проєктів або роботи з відносно малими обсягами даних зручно використовувати MySQL чи PostgreSQL, оскільки вони прості в адмініструванні, мають розвинену екосистему інструментів та не потребують значних витрат на інфраструктуру [42].

Обираючи технології для створення інформаційної системи за темою кваліфікаційної роботи, слід враховувати такі критерії: масштабованість (можливість швидкого розширення ресурсів), вартість (гнучкі моделі оплати “pay-as-you-go”), безпека (підтримка шифрування, контроль доступу, відповідність міжнародним стандартам).

Важливим критерієм системи є зручність інтеграції з рекламними платформами та зовнішніми сервісами. Проведений аналіз можливих технологій та сервісів демонстрував широкий спектр інструментів, які можуть бути використані для побудови інформаційної системи аналітики рекламних кампаній.

Використання хмарних платформ дасть змогу в перспективі поєднати масштабоване сховище даних, потужні інструменти обробки та гнучкі засоби візуалізації, що забезпечує ефективність і надійність системи.

2.2 Обґрунтування вибору архітектури та інструментів

Проектування інформаційної системи для аналітики рекламних кампаній вимагає вибору архітектурного підходу та відповідних технологій, які забезпечать виконання вимог до масштабованості, надійності, продуктивності та зручності використання.

Початковим етапом виконання даного завдання є обґрунтування архітектурного підходу. На основі аналізу предметної області та можливих моделей реалізації, доцільним є використання хмарно-орієнтованої багаторівневої архітектури, що включає: рівень збору даних, рівень зберігання, рівень обробки і рівень аналітики та візуалізації.

Рівень збору даних – інтеграція з рекламними платформами (Google Ads, Facebook Ads, LinkedIn Ads) та системами веб-аналітики (Google Analytics 4) [39; 40]. Дані надходять через API або інтеграційні сервіси (Zapier, Apache NiFi) [51].

Рівень зберігання – централізоване сховище на основі хмарних баз даних. Для великих масивів інформації доцільно застосувати Google BigQuery або Amazon Redshift [36; 38], а для середніх і малих обсягів – PostgreSQL чи MySQL [42].

Рівень обробки – використання систем обробки даних у реальному часі (Apache Spark для розподілених обчислень та підготовки даних до аналізу) [43].

Рівень аналітики та візуалізації – застосування інструментів бізнес-аналітики (Google Looker Studio, Power BI, Tableau) для представлення результатів у формі інтерактивних звітів і дашбордів [45; 46; 47]. Також варто зазначити, що готові рішення потребують ретельної роботи над інтеграцією та можуть обмежувати гнучкість.

Така архітектура забезпечує розділення функціональності, спрощує адміністрування системи та дозволяє масштабувати окремі її компоненти залежно від навантаження.

Наступним кроком дослідження є обґрунтування вибору інструментів для кожного рівня проектування інформаційної системи.

Для зберігання та аналітики великих даних детальніше розглянуто СУБД BigQuery, Redshift, Snowflake, Vertica, ClickHouse. За результатами проведених досліджень рекомендовано використовувати хмарну систему управління базами даних BigQuery, оскільки цей сервіс забезпечує високу швидкодію SQL-запитів, інтеграцію з іншими сервісами Google Cloud та має гнучку модель оплати «pay-as-you-go» [38; 48].

Для обробки даних можна використовувати універсальний фреймворк Apache Spark, який може використовуватися для розподіленої обробки даних, що дозволяє масштабувати систему залежно від обсягів інформації та підтримує інтеграцію з BigQuery та іншими сховищами [42].

Для зберігання середніх і малих даних використання MySQL обґрунтоване його надійністю, розвиненими можливостями роботи з аналітичними запитамі та відкритим вихідним кодом [42].

Для швидкого створення інтерактивних дашбордів можна розглянути Google Looker Studio, який інтегрується з BigQuery та іншими сервісами Google [46]. У випадку корпоративних сценаріїв може використовуватись Power BI, що інтегрується з Azure [47]. Для побудови веб інтерфейсу інформаційної системи лише готових рішень буде недостатньо. Для вирішення цієї проблеми можна використати React, з інтеграцією Google Looker Studio що забезпечить повну кастомізацію візуалізації даних.

Для інтеграції даних можна розглянути Apache NiFi, оскільки він підтримує автоматизацію потоків даних та дозволяє інтегрувати різноманітні джерела без додаткового програмування [41].

Варто також зазначити, що хмарні сервіси потребують конфігурування, що правильно виконувати поставлені задачі та відповідати меті створення інформаційної системи. Для досягнення цього як мінімум знадобиться забезпечити безпеку даних. Іншими словами створити свій інтерфейс для реєстрації та авторизації є необхідний крок. Для його виконання знадобиться UI та серверна частина як зможе обробляти та перевіряти доступність даних. Для цього пропоную використати React JS для побудови інтерфейсу користувача, а для серверної частини використати Java та Spring Boot. У поєднанні ці два інструменти дозволять створити гнучке рішення майже будь-яких проблем пов'язаних із веб додатками, а також забезпечить можливість розширювати додаток у майбутньому.

Розглянемо обрані інструменти на рисунку 2.2 – Схема обраних архітектурних інструментів.

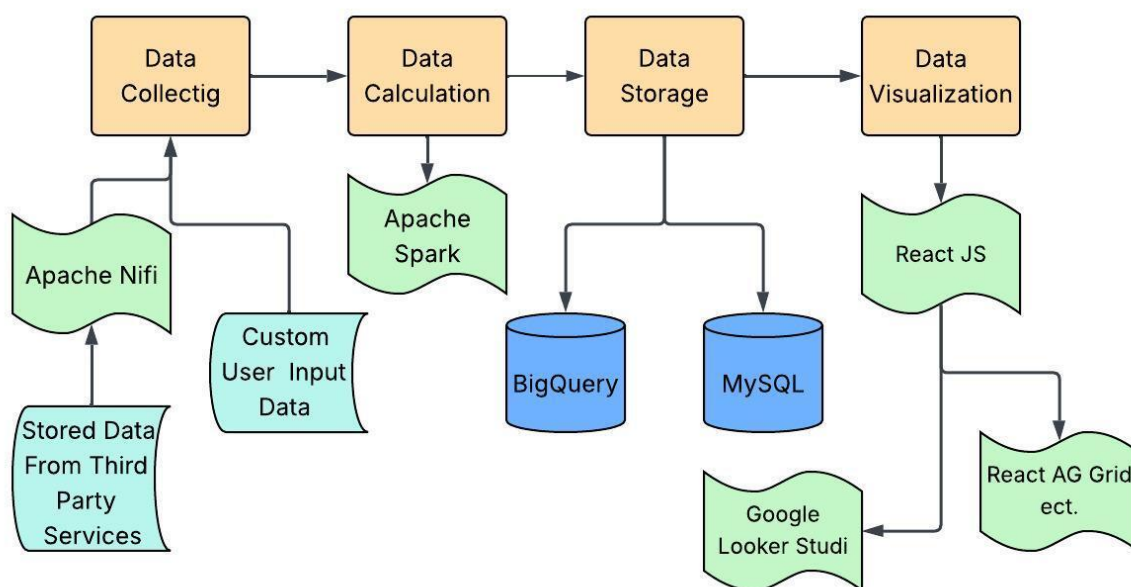


Рисунок 2.2 – Схема обраних архітектурних інструментів.

Запропоноване поєднання архітектури та інструментів дозволяє:

- ✓ забезпечити масштабованість системи відповідно до обсягів рекламних даних (Додаток Д);
- ✓ гарантувати швидкодію при виконанні складних SQL-запитів та аналітичних обчислень (Додаток Б);
- ✓ досягти гнучкості та інтегрованості завдяки підтримці різних рекламних платформ і каналів збору даних;
- ✓ забезпечити зручність прийняття управлінських рішень за допомогою інтерактивних звітів і візуалізацій.

Таким чином, обрана архітектура та набір інструментів повністю відповідають завданню розробки інформаційної системи для аналітики рекламних кампаній, створюючи основу для її ефективного реалізації та подальшого масштабування.

2.3 Реалізація ключових компонентів системи

Реалізація інформаційної системи для аналітики рекламних кампаній базується на багаторівневій архітектурі, що включає рівень збору даних, рівень зберігання та обробки, а також рівень аналітики та візуалізації. В якості технологічної основи було обрано Java для бекенд-частини системи та React для користувацького інтерфейсу. Такий підхід забезпечує надійність серверної логіки, гнучкість інтеграцій та сучасний інтерфейс для користувачів.

Бекенд використовуємо (Java, Spring Boot). Бекенд системи реалізує бізнес-логіку, інтеграцію з джерелами даних та надання API для взаємодії з інтерфейсом користувача. Основні компоненти:

Модуль збору даних – реалізований у вигляді сервісів на Java, що здійснюють виклики до API рекламних платформ (Google Ads, Facebook Ads, LinkedIn Ads). Для цього застосовується Spring WebClient та бібліотеки для роботи з REST API.

Модуль обробки даних – інтегрується з Apache Spark для розподіленої обробки даних, попереднього очищення та агрегації ключових показників (CTR, CPC, ROI).

Модуль зберігання – реалізовано інтеграцію з Google BigQuery для зберігання великих обсягів даних, а також з PostgreSQL для зберігання метаданих та налаштувань користувачів.

REST API – надає доступ до результатів обробки у форматі JSON. JSON (JavaScript Object Notation) – це текстовий формат для обміну структурованими даними, який легко читається як людиною, так і машиною.

Приклад контролера на Spring Boot для отримання даних рекламної кампанії наведено на рисунку 2.3.

```

@RestController
@RequestMapping("/api/campaigns")
public class CampaignController {

    private final CampaignService campaignService;

    public CampaignController(CampaignService campaignService) {
        this.campaignService = campaignService;
    }

    @GetMapping("/{id}/stats")
    public ResponseEntity<CampaignStatsDto> getCampaignStats(@PathVariable Long id) {
        return ResponseEntity.ok(campaignService.getStats(id));
    }
}

```

Рисунок 2.3 — Приклад контролера на Spring Boot

Сховище даних. BigQuery використовується як основне сховище великих масивів даних для аналітичних SQL-запитів. BigQuery дозволить не тільки зберігати велику кількість даних, а й забезпечить швидкий доступ до даних із великою обчислювальною потужністю.

MySQL забезпечить зберігання параметрів користувачів, та додаткових параметрів, таких як інформація про збережені звіти, інформацію про інтеграцію з іншими API тощо. Мова запитів SQL забезпечить можливість створення гнучких запитів із можливістю отримувати дані одразу із багатьох таблиць, при потребі.

Для підключення до BigQuery у Java застосовується офіційний клієнт Google Cloud BigQuery Client Library [38]. В офіційній документації можна знати усю необхідну інформацію із прикладами реалізації, що значно полегшить розробку.

Фронтенд або інтерфейс користувача тут React з використанням сучасних бібліотек для створення та інтеграції. Основні можливості:

- ✓ Інтерактивні дашборди – реалізовані з використанням бібліотеки Recharts для побудови графіків CTR, CPC, ROI та інших показників;
- ✓ Фільтри та пошук – надають можливість користувачу аналізувати дані за періодами, каналами та метриками;
- ✓ Інтеграція з REST API – отримання даних від бекенду через Axios або Fetch API;
- ✓ UI-компоненти – побудовані на базі shadcn/ui для єдиного стилю та зручності користувачів.

Варто зауважити що React JS для побудови користувацького інтерфейсу використовує Java Script XML скорочено JSX. Цей підхід надає можливість написання коду синтаксично схожому на HTML, але всередині файлу JavaScript. HTML. Варто пояснити що таке HTML. Hyper Markup Text Language або HTML — це мова розмітки веб-сторінки яку використовує браузер при візуалізації контенту. Варто пояснити і решту аббревіатур таких як REST та API. REST (Representational State Transfer) — це архітектурний протокол для який визначає правила взаємодії між програмними компонентами, наприклад, між клієнтом і сервером вебзастосунку. API (Application Programming Interface) — це набір правил, стандартів та інструментів який дозволяє елементам системи взаємодіяти між собою. Загалом REST API демонструє розробнику можливості та правила для інтеграції та взаємодії з елементом інформаційної системи.

Приклад React-компонента для відображення метрик кампанії
рисунок.2.4.

```

import React, { useEffect, useState } from "react";
import { Card, CardContent } from "@components/ui/card";
import { ResponsiveContainer, LineChart, Line, XAxis, YAxis, Tooltip } from "recharts";

export default function CampaignMetrics({ campaignId }) {
  const [metrics, setMetrics] = useState([]);

  useEffect(() => {
    fetch(`/api/campaigns/${campaignId}/metrics`)
      .then(res => res.json())
      .then(data => setMetrics(data));
  }, [campaignId]);

  return (
    <Card className="p-4 shadow-lg rounded-2xl">
      <CardContent>
        <h2 className="text-xl font-bold mb-4">Показники кампанії</h2>
        <ResponsiveContainer width="100%" height={300}>
          <LineChart data={metrics}>
            <XAxis dataKey="date" />
            <YAxis />
            <Tooltip />
            <Line type="monotone" dataKey="ctr" stroke="#8884d8" name="CTR" />
            <Line type="monotone" dataKey="cpc" stroke="#82ca9d" name="CPC" />
          </LineChart>
        </ResponsiveContainer>
      </CardContent>
    </Card>
  );
}

```

Рисунок 2.4 — Приклад UI jsx компоненти.

Для побудови динамічних дашбордів користувачі взаємодіють з React-інтерфейсом, який звертається до бекенду.

На рівні бекенду оброблені дані агрегуються у BigQuery і повертаються у вигляді зручних структур (JSON).

Таким чином, реалізація ключових компонентів системи поєднує надійність серверної частини на Java, можливості аналітики у BigQuery,

гнучкість інтеграції з рекламними платформами та сучасний інтерфейс на React. Це забезпечує масштабованість і зручність використання системи у реальних умовах.

2.4 Застосування хмарних сервісів у реалізації проекту

Хмарні сервіси сьогодні активно використовуються для розробки, розгортання та масштабування інформаційних систем, зокрема аналітичних платформ. Вони забезпечують високу доступність, гнучкість і економію ресурсів. Розглянемо конкретні приклади застосування ключових хмарних сервісів у реалізації інформаційної системи рисунок 2.5.

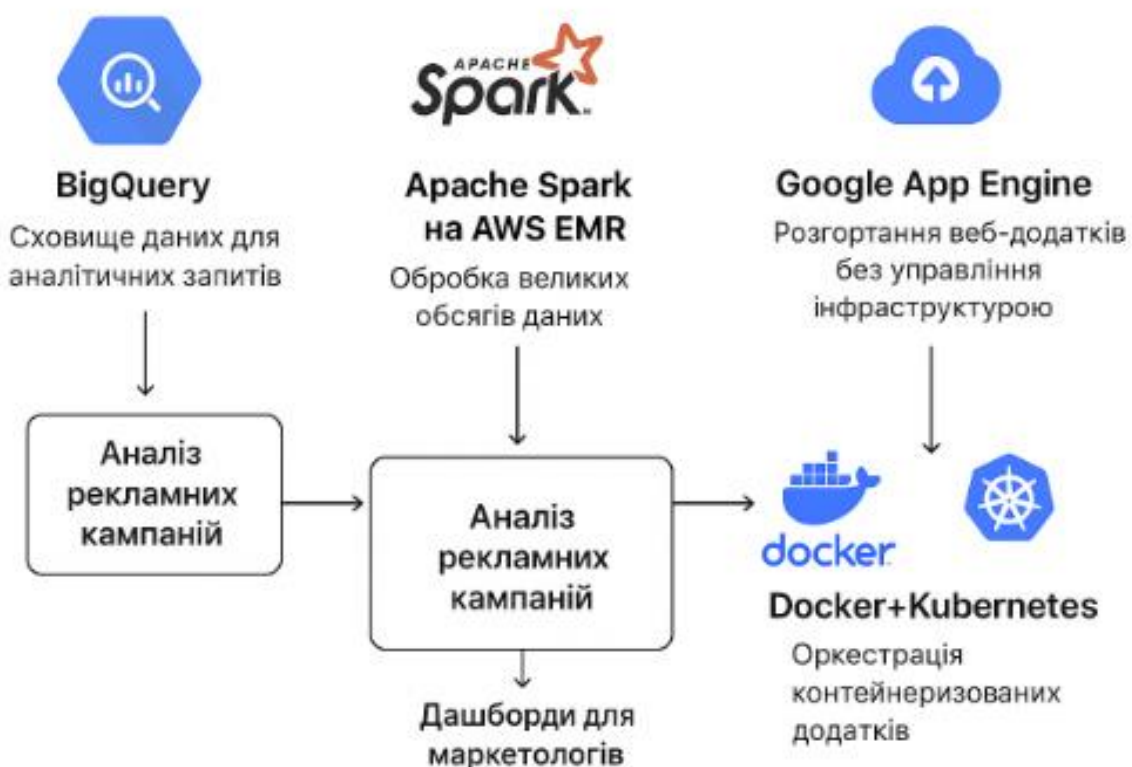


Рисунок 2.5 — Приклад застосування хмарних сервісів у реалізації.

Розглянемо хмарні сервіси які будуть задіяні та взаємодіяти у даній інформаційній системі такі як, Google BigQuery, Apache Spark, Google App Engine.

BigQuery – це серверлес-сховище даних від Google Cloud, призначене для виконання аналітичних запитів на великих обсягах даних у форматі SQL. Основні можливості BigQuery включають:

- ✓ Масштабованість та швидкість. BigQuery дозволяє обробляти петабайти даних без потреби в управлінні інфраструктурою.
- ✓ Інтеграція з іншими сервісами. СУБД легко інтегрується з Cloud Storage, Dataflow, Pub/Sub та іншими сервісами Google Cloud.
- ✓ Аналітика в реальному часі, яка дозволяє підтримувати стрімінгову інгестації даних дозволяє аналізувати події майже миттєво.
- ✓ SQL-аналітика, завдяки якій користувачі можуть виконувати складні запити SQL для агрегування, фільтрації та з'єднання даних [38].

У практичній реалізації система може завантажувати дані рекламних кампаній у BigQuery через ETL-процеси або Apache Spark, виконувати аналітичні запити та формувати дашборди для маркетологів. Наприклад, можна створювати щоденні звіти про CTR, CPC та ROI рекламних кампаній, виконуючи паралельну обробку мільйонів записів за секунди.

Apache Spark – це розподілена обчислювальна платформа для великих даних, що дозволяє виконувати як пакетну, так і стрімінгову обробку. Використання Spark у хмарі на AWS EMR (Elastic Map Reduce) забезпечує:

- ✓ Швидке додавання або видалення вузлів відповідно до навантаження, оптимізуючи витрати, що призводить до автоматичного масштабування кластерів.
- ✓ Spark підтримує обробку структурованих та неструктурованих великих даних, включаючи журнали подій, кліки на сайті та транзакції. Також дозволяє реалізовувати паралельні обчислення шляхом їх розподілу на десятки і сотні вузлів, що значно пришвидшує обробку великих обсягів даних.

- ✓ Інтеграцію з S3 та Redshift, завдяки якій дані можуть зберігатися в S3, а результати обробки – експортуватися у Redshift або BigQuery для подальшої аналітики.

У системі аналітики рекламних кампаній Spark на EMR може виконувати підрахунок метрик у реальному часі та пакетну обробку історичних даних для прогнозування ефективності реклами, наприклад, формувати сегментацію аудиторії та моделі CTR.

Google App Engine (GAE) – це серверлес-платформа для розгортання веб-додатків у хмарі без управління інфраструктурою. Основними перевагами платформи є:

- ✓ Можливість автоматичного масштабування додатків відповідно до навантаження без ручного налаштування серверів.
- ✓ Підтримка різних мов програмування Python, Java, Node.js, Go та інші.
- ✓ Інтеграція з Google Cloud шляхом легкого підключення до BigQuery, Cloud Storage та Pub/Sub [52].

У прикладі аналітичної системи веб-інтерфейс для маркетологів може розгортатися на GAE, обробляти запити до BigQuery та Spark і відображати аналітичні дашборди з даними в реальному часі.

Docker + Kubernetes у хмарі (GKE). Контейнеризація за допомогою Docker та оркестрація через Kubernetes дозволяють забезпечити портативність, масштабованість та стабільність розгортання додатків:

- ✓ Docker ізолює додатки та їхні залежності, забезпечуючи однакове середовище на локальній машині та у хмарі.
- ✓ Kubernetes управляє контейнерами, автоматично масштабуючи додатки, відновлюючи збої та забезпечуючи балансування навантаження.

Розглянемо детальніше що таке GKE? GKE (Google Kubernetes Engine) – керований Kubernetes у Google Cloud, інтегрований з усіма сервісами GCP, де GCP — це Google Cloud Platforms.

Спираючись на попередньо проаналізовані інструменти та сервіси можна зробити висновок, що для побудови інформаційної системи можна використати лише ті хмарні рішення які надає Google. Розглянемо таку модель інформаційної системи на рисунку 2.6.

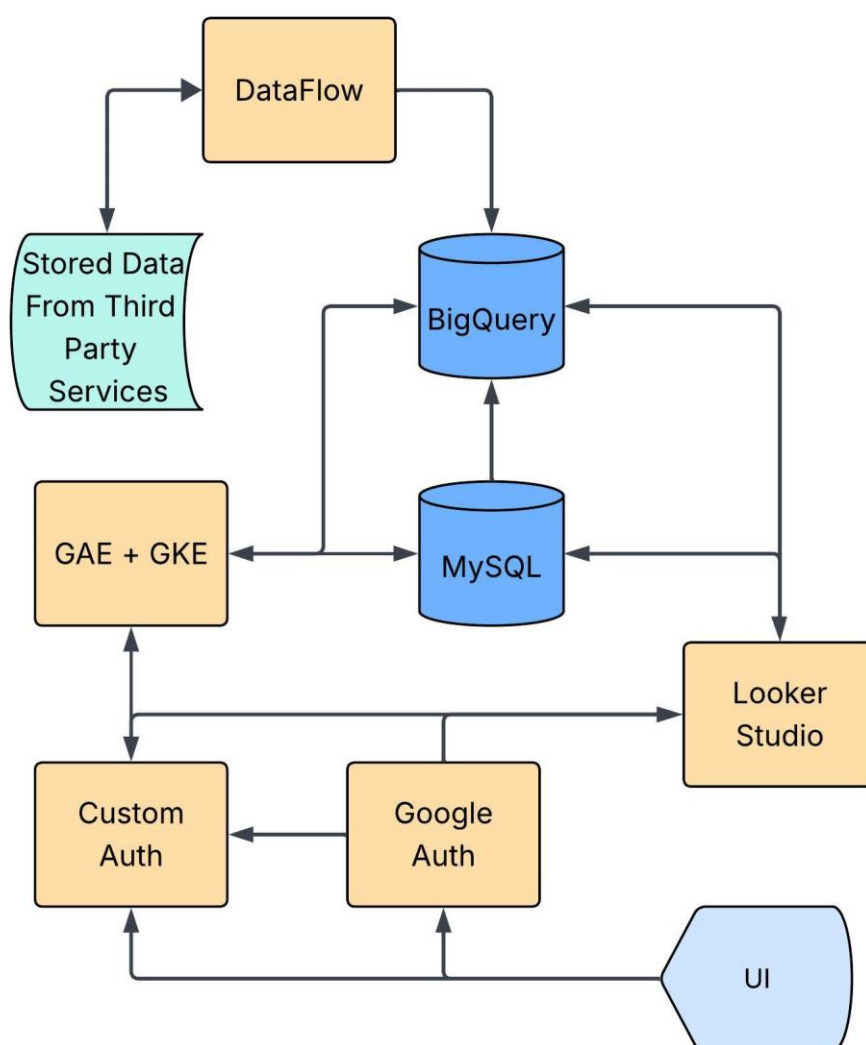


Рисунок 2.6 – Спрощений приклад моделі інформаційної системи із використання лише Google Cloud.

Така інформаційна інформаційна система має як свої переваги так і недоліки. Але спочатку розглянемо що має і може. Звісно існує інтерфейс користувача у якому, можна зареєструватись та авторизуватись. На схемі присутні два аунтифікаційні сервіси, але насправді потрібен тільки один. Google Auth може використовуватись у варіанті якщо це внутрішній продукт без публічного доступу. У випадку якщо система має публічний доступ необхідно забезпечити гнучкість та надійність доступу до даних, для цього можна створити власний сервіс який буде перевіряти права доступу користувача (Додаток А). Як додатковий функціонал, що може спростити авторизацію або реєстрацію, можна використати Google Auth та використовувати отриманий ключ у системі для розпізнання користувача та підтвердження його прав доступу.

Перевагами такої інформаційної системи є те що більша частина сервісів вже існує. Їх потрібно налаштувати, та реалізувати взаємодію на основі прав доступу користувачів. Окрему увагу варто приділити схемам баз даних та реалізувати трансформацію даних із сторонніх сервісів.

Недоліками такої системи є те що у неї є певні обмеження і правила які нам диктує власник сервісу та хмари. Це може завади у реалізації нових можливостей або ускладнити реалізацію.

У системі аналітики можна розгортати мікросервіси, які збирають дані, виконують обчислення Spark або аналітичні запити до BigQuery, і надають API для фронтенду. Kubernetes забезпечує високу доступність та автоматичне масштабування цих сервісів під час пікових навантажень.

Окремої уваги потребують CI-CD. Запровадження процесів безперервної інтеграції (Continuous Integration, CI) та безперервного розгортання (Continuous Deployment, CD) є ключовим етапом підвищення якості програмного забезпечення та забезпечення його безперебійної роботи. Для даної системи було обрано сучасні інструменти

контейнеризації та оркестрації – Docker та Kubernetes, що дозволяють забезпечити гнучкість, масштабованість та надійність у процесах розробки та експлуатації.

Після внесення змін у вихідний код відбувається запуск CI-процесу, який включає такі підзадачі:

- ✓ Збірка Docker-образів передбачає, що усі модулі системи упаковано у контейнери Docker, що забезпечує незалежність від середовища та мінімізує ризик конфліктів залежностей.

- ✓ Статичний аналіз коду та автоматичне тестування як крок визначає, що перед створенням образів виконується статичний аналіз (SonarQube), модульні та інтеграційні тести, що дозволяє виявити помилки ще на ранніх етапах.

- ✓ Збереження образів у реєстрі: перевірені та зібрані Docker-образи завантажуються у приватний реєстр (Docker Hub, GitLab Registry чи Amazon ECR), звідки вони доступні для етапу розгортання.

- ✓ Етап безперервного розгортання (CD)

- ✓ Після успішного проходження етапу CI відбувається автоматичне розгортання додатка у Kubernetes-кластері. Основні процеси включають:

- ✓ Тестове розгортання у staging-середовищі. Kubernetes дозволяє створювати ізольовані середовища для тестування без впливу на робоче середовище.

Kubernetes може застосовувати стратегію поступового розгортання (rolling updates), що забезпечить безперервність роботи системи при масштабному оновленні. Крім цього Kubernetes відповідає за керування контейнерами, балансування навантаження, автоматичне масштабування та самовідновлення сервісів. У разі виявлення критичної помилки при розгортанні Kubernetes використає можливість швидкого відкату.

Швидкого відкат — це одна із можливостей Kubernetes, що гарантує стабільність у разі виявлення помилок.

Інтеграція з Prometheus та Grafana дозволить відстежувати продуктивність сервісів, та наявність помилок у роботі. У разі виявлення критичної помилки можна буде відкотити версію додатка, використовуючи Kubernetes до .коректної версії.

Docker гарантує портативність, уніфікацію середовища виконання та швидке створення ізольованих контейнерів із додатком. Крім цього розробник зможе розгорнути Docker Image на локальній машині, що дозволить протестувати код не оновлюючи додаток. При використанні у поєднанні із мікро-сервісною архітектурою Docker та Kubernetes зможуть суттєво пришвидшити та полегшити процес підтримки та розвідку інформаційної системи. Розробник може у віддаленому тестовому Kubernetes контейнері створити середовище з лише дотичними сервісами до сервісу у який потрібно додати зміни

Поєднання процесу CI/CD з Docker та Kubernetes забезпечує стабільне, передбачуване та масштабоване розгортання нових версій програмного забезпечення. Це дозволяє значно скоротити час виходу оновлень, знизити ризики збоїв і підтримувати безперервну роботу системи навіть при високих навантаженнях. Крім того це дасть змогу розділити інтеграційні тести відповідно до сервісів до яких вони дотичні. Це скорить час тестування змін перед об'єднання із головним кодом додатку, мінімізує ризики виникнення помилок та забезпечить можливість перевіри весь додаток після змін не розгортаючи його на головних серверах продукту, цей процес називається Staging або постановочне середовище. Суть його полягає у тому що весь додаток розгортається у тестовому середовищі із доступом до реальних даних.

На рисунку 2.7 наведено приклад внесення змін в проєкт на основі CI/CD процесів.

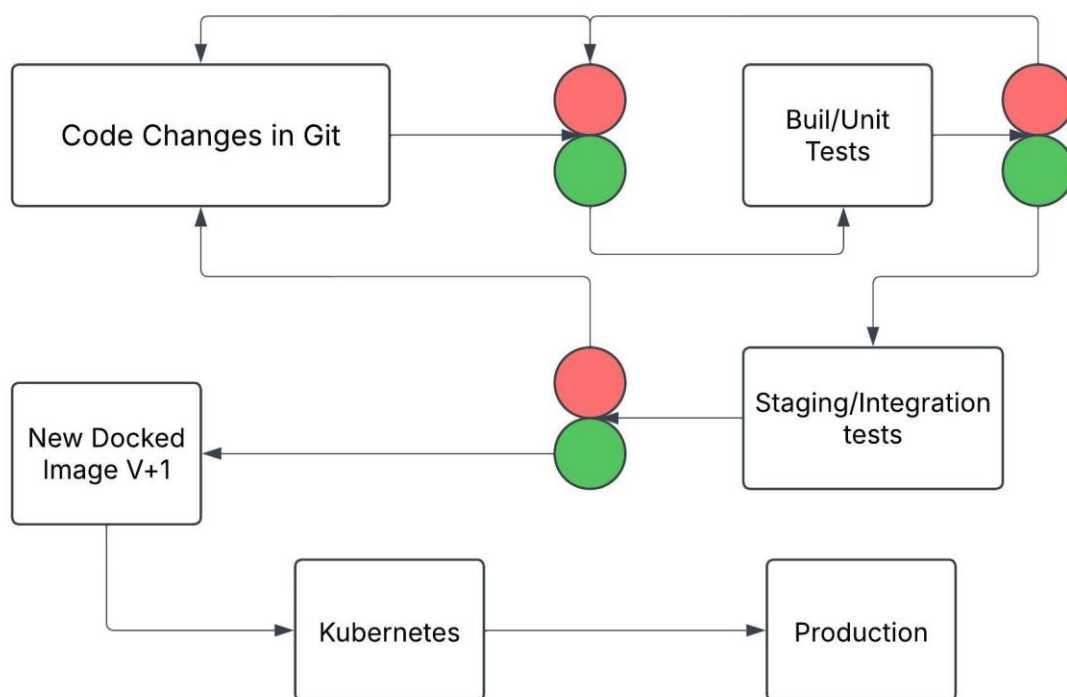


Рисунок 2.7 — Приклад розгортання із застосування Docker + Kubernetes

Хмарні сервіси можуть значно спростити процес створення інформаційної системи. Вони можуть надати додаткові можливості та мають гнучкість у масштабуванні і не потребують додаткового обслуговування. Розглянемо це на простих прикладах у порівнянні із власними серверами.

При роботі із власним сервером обов'язково треба пам'ятати про резервне живлення, резервний доступ до мережі, об'єм пам'яті та технічне обслуговування, що для малих проєктів або стартапах може бути занадто дорого. Як приклад можна розглянути додаток для збереження карток лояльності. Створити окрему самостійну API із власним сервером буде коштувати недешево, а от хмарні сервіси пропонують набагато менші ціни,

тому для додатків із малою залежністю від бази даних це вигідно. Крім того за вас вже продумали як і безперервність доступу так і резервне живлення.

Не варто забувати про розробку та підтримку системи. Для її оновлення як мінімум необхідно перезавантажити сервер, а якщо він існує в одиночному екземплярі — система не буде працювати до завершення процесу перезавантаження. З точки зору користувача якщо програма недоступна у той час коли вона йому знадобилась означає, що це ненадійна програма. Тобто щоб такого не сталося була можливість оновлювати систему потрібно мінімум два сервери які зможуть повертати одні і ті ж самі дані за одним і тим самим запитом. Це ускладнює архітектуру системи, вже є два сервери із додатком і третій із базою даних. Сучасні хмарні сервіси мають готові вирішення цих питань потрібно лише проаналізувати документацію і зрозуміти який сервіс відповідає усім вимогам системи.

РОЗДІЛ 3.

ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ

3.1 Опис функціональності для кінцевих користувачів

В ході виконання кваліфікаційної роботи було здійснено практичну реалізацію поставленого завдання на прикладі проекту інформаційна система Global Ads Analytics(GAA). Дана інформаційна система являє собою аналітичну платформу та передбачає використання даних для розуміння та оптимізації рекламних кампаній на різних ринках та платформах (Додаток Д). Вона охоплює широкий спектр діяльності, від відстеження ключових показників ефективності (KPI) до аналізу поведінки користувачів та ефективності кампаній, зрештою спрямованої на підвищення ефективності реклами та рентабельності інвестицій.

Цільова аудиторія GAA — це компанії з широкою сферою діяльності у різних країнах, які користуються різноманітними рекламними платформами. Такими як Google Ads, Facebook Ads, тощо.

Нижче розглянемо основні функції системи.

1. Авторизація та управління обліковими записами.

Користувачі мають можливість проходити реєстрацію та авторизацію у системі. Доступ передбачає кілька рівнів ролей:

- ✓ Адміністратор — управляє користувачами, налаштовує інтеграції, контролює доступ до даних.
- ✓ Аналітик — має доступ до даних рекламних кампаній, може формувати звіти, будувати візуалізації.
- ✓ Маркетолог — працює з готовими звітами та прогнозами, може відстежувати ключові показники ефективності (KPI).

2. Інтеграція даних

Кінцевий користувач може завантажувати дані у систему з різних джерел:

- ✓ Рекламних платформ (Google Ads, Facebook Ads тощо);
- ✓ Файли у певному форматі на основі шаблону отриманого із інформаційної системи.

3. Аналітика та обробка даних

- ✓ Автоматичне очищення та нормалізацію даних;
- ✓ При потребі розрахунок ключових маркетингових показників таких як: CTR, CPC, ROAS, CPL, conversions, тощо.

4. Візуалізація та звітність

Для зручності користувачів реалізовано інтерактивні дашборди, де можна переглядати:

- ✓ Графіки динаміки показників;
- ✓ Порівняння ефективності різних рекламних каналів.

Розглянемо більш детально вже існуючі варіанти головної сторінки у таких компаній як Alphabet(Google), Yahoo та Taboola.

На рисунку 3.1 представлено головна сторінка Google ADS.

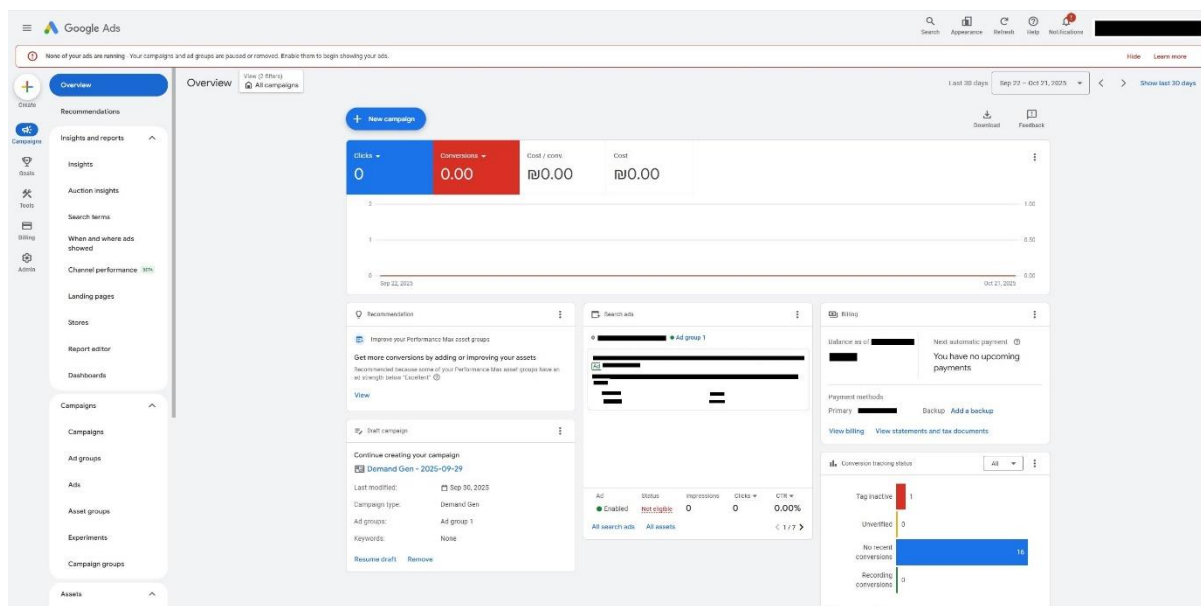


Рисунок 3.1 - Головна сторінка Google ADS [57].

Перша сторінка містить базові дані про баланс аккаунту, базову звітність по всіх кампаніях акаунта, фільтр по даті та декілька меню з великою кількістю підпунктів. А також можливістю завантажити звітність, але це не уточнено. Крім цього присутня значна кількість посилань у самому звіті.

У Yahoo DSP, а точніше Demand Site Platform, Рисунок 3.2 на головній сторінці можна побачити звіт обраним промітком дат і його змінити. Також одразу присутня звітність по активності кампаній та її витратам, за певний проміжок часу. Присутня можливість додати фільтр та стисле меню з лівою сторони. Присутня можливість експорту даних, але яких саме на перший погляд незрозуміло.

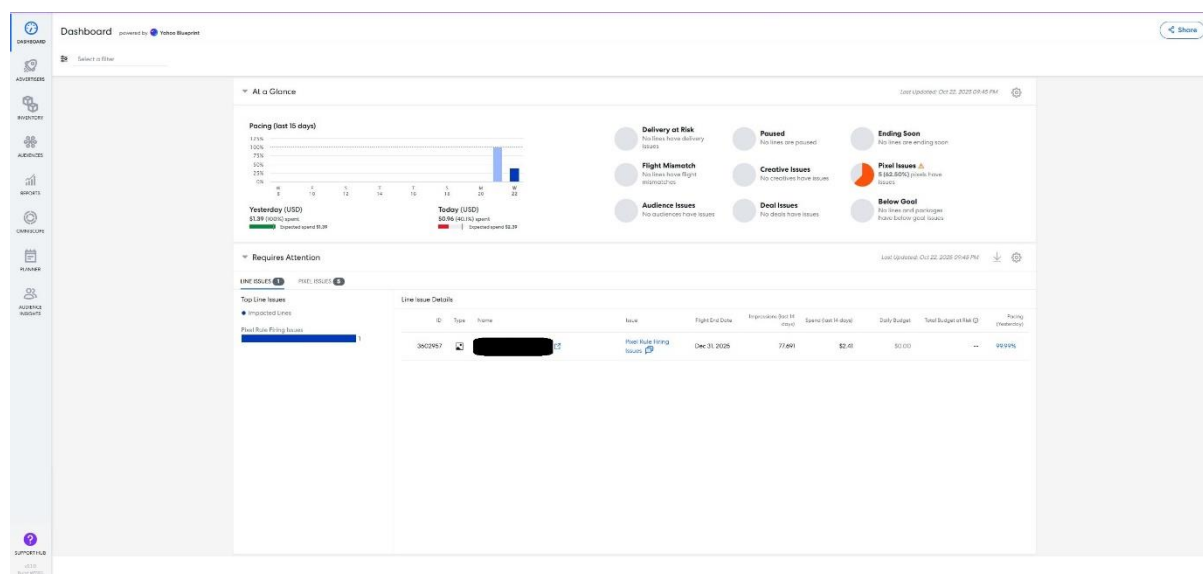


Рисунок 3.2 - Головна сторінка Yahoo DSP [58].

Перейдемо до Taboola Realize, представленого на рисунку 3.3. На головній сторінці можна одразу побачити звітність по усіх рекламних кампаніях за останні 7 днів у двох форматах. Перший — це графік із метриками, другий — це таблиця із більш детальною інформацією. Також у інтерфейсі присутні можливості фільтрації, експорту, зміни колонок на графіку та можливістю перегляду звіту по певній кампанії, а головне III

помічник. Як і у попередньо переглянутих прикладах є меню з лівої сторони тільки із головними пунктами і меню вибору статичних звітів.

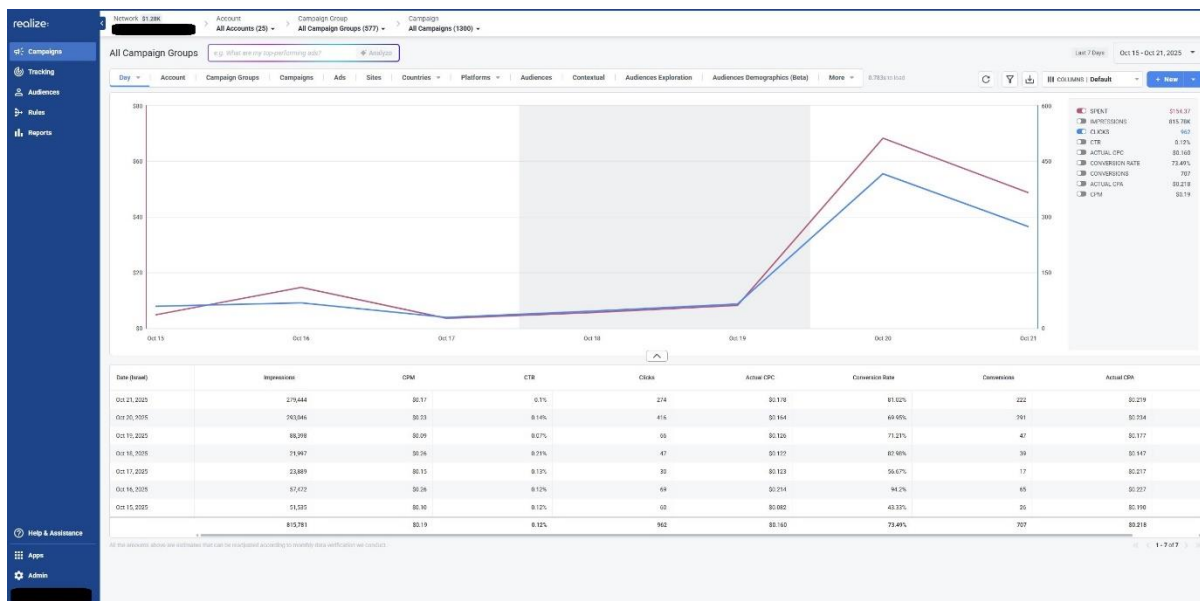


Рисунок 3.3 - Головна сторінка Taboola Realize [56].

Оскільки головна мета роботи орієнтована на звітність рекламних кампаній то Taboola Realize найбільш підходящий як приклад. Він має базові фільтри та статичну звітність із певними колонками. У варіанті універсальної звітності потрібен «конструктор» який дозволить відобразити звітність по будь-якій комбінації колонок. Усі вище переглянуті платформи мають таку можливість тому у нашій моделі вона теж знадобиться, а статичні звіти можна створювати на основі конструктора.

3.2 Можливості інтеграції з іншими системами (API, інтерфейси)

Сучасні інформаційні системи рідко існують у відриві від ширшої цифрової екосистеми, оскільки бізнес-процеси передбачають постійний обмін даними між різними платформами та сервісами. Розроблена система побудована таким чином, щоб забезпечити максимальну відкритість та

гнучкість у взаємодії з іншими рішеннями, як локальними, так і хмарними. Це досягається завдяки використанню відкритих інтерфейсів, стандартних протоколів та підтримки широкого спектра API.

Одним із ключових напрямів інтеграції є взаємодія з рекламними платформами, такими як Google Ads, Facebook Ads чи TikTok Ads. За допомогою API цієї системи можливо автоматично отримувати дані про активність кампаній, витрати та ефективність реклами, що значно зменшує потребу у ручному завантаженні звітів. Водночас передбачено можливість двосторонньої синхронізації із CRM-системами та ERP-рішеннями, що дозволяє, наприклад, автоматично коригувати бюджетні обмеження, відстежувати виконання маркетингових планів або пов'язувати результати рекламної діяльності з реальними продажами.

Важливим аспектом є інтеграція з хмарними сервісами обробки даних. Так, Google BigQuery використовується для виконання складних аналітичних запитів у реальному часі, тоді як інфраструктура AWS (зокрема S3 та EMR на базі Apache Spark) забезпечує можливість масштабованої обробки великих обсягів інформації. Використання контейнеризації через Docker та системи оркестрації Kubernetes (як у Google Kubernetes Engine, так і в Amazon EKS чи Azure AKS) робить систему придатною до розгортання у розподілених середовищах, де кожен сервіс може легко замінюватися чи масштабуватися без впливу на загальну стабільність роботи.

Зручність кінцевих користувачів значною мірою визначається не лише наявністю даних, а й способом їх подання. Тому система підтримує інтеграцію з інструментами бізнес-аналітики, зокрема Google Data Studio, Looker Studio, Power BI та Tableau. Це дозволяє спеціалістам швидко створювати індивідуальні дашборди та аналітичні панелі без глибокого занурення у внутрішню структуру бази даних. Крім того, користувачі

можуть експортувати дані у стандартних форматах CSV, XLSX чи JSON, що відкриває можливість їх подальшої обробки у будь-якому іншому програмному середовищі.

Ще однією перевагою є підтримка автоматизованих повідомлень і подій. За допомогою вебхуків система може в реальному часі інформувати зовнішні застосунки про ключові зміни, наприклад надсилати сповіщення у корпоративні месенджери на кшталт Slack або Telegram. Це дозволяє оперативно реагувати на перевищення KPI чи виявлені аномалії у даних, не чекаючи на заплановані звіти.

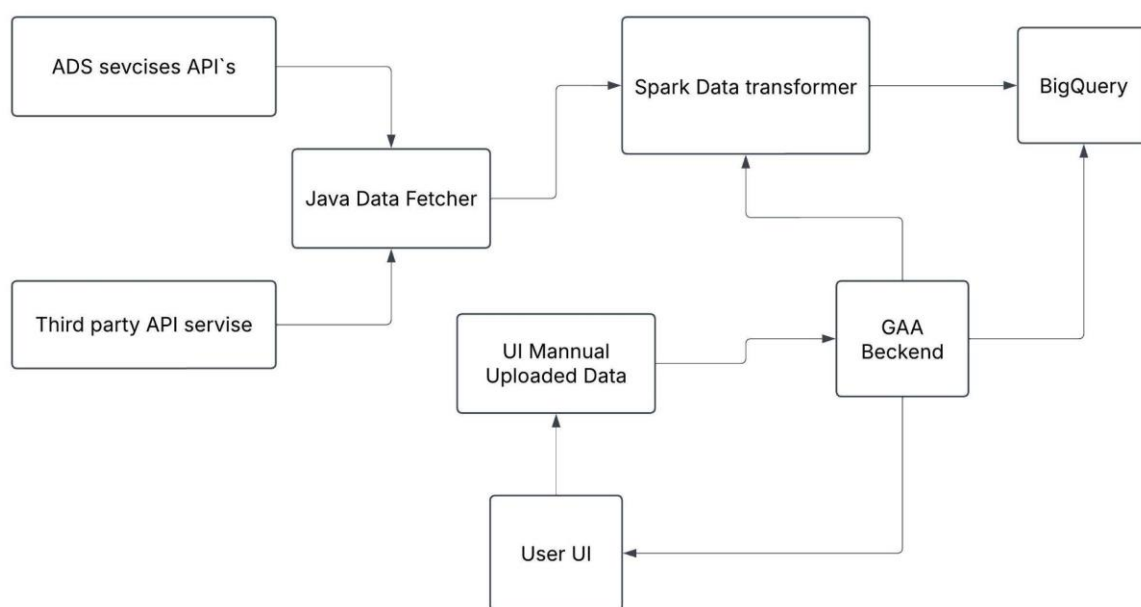


Рисунок . 3.4 — Загальна діаграма додатка.

Веб-інтерфейс системи (рис.3.4) спроектовано з урахуванням можливостей керування інтеграціями безпосередньо через зручний графічний інтерфейс. Користувач має змогу налаштовувати підключення до зовнішніх сервісів, регулювати права доступу та контролювати процеси обміну даними без залучення додаткового програмного забезпечення чи технічних спеціалістів. Адаптивність інтерфейсу дозволяє здійснювати такі

операції як з комп'ютера, так і з мобільних пристроїв, що розширює можливості застосування системи у динамічному робочому середовищі.

У сукупності ці можливості інтеграції роблять систему не ізольованим інструментом, а частиною великої корпоративної екосистеми. Вона може бути одночасно аналітичним ядром, сховищем даних і платформою для прийняття управлінських рішень, яка з'єднує різні джерела інформації в єдине цілісне середовище.

3.3 Аналіз отриманих результатів і перспективи використання

Розроблена інформаційна система показала свою ефективність як інструмент для збору, зберігання та аналітики даних рекламних кампаній із використанням хмарних технологій. У процесі тестування було підтверджено, що система здатна інтегруватися з різними зовнішніми джерелами даних, обробляти великі обсяги інформації та формувати звіти у зручному для користувачів вигляді. Це значно підвищує продуктивність роботи маркетологів та аналітиків, оскільки усуває необхідність ручного перенесення інформації з різних платформ та зменшує ризик помилок, пов'язаних із дублюванням чи неповнотою даних.

Завдяки застосуванню сервісів BigQuery та Apache Spark система продемонструвала здатність до масштабування, що є критично важливим у випадках зростання обсягів рекламних даних. Використання контейнеризації та хмарних середовищ (Docker, Kubernetes) забезпечує гнучкість у розгортанні та підтримці рішення, а інтеграція з CRM дозволяє відслідковувати ефективність рекламних кампаній не лише у межах медійних показників, а й з точки зору реальних бізнес-результатів.

Аналіз отриманих результатів показує, що система успішно виконує поставлену мету – створює єдине середовище для аналітики рекламних кампаній, яке поєднує функції збору даних, їх очищення, обробки,

візуалізації та формування рекомендацій. Особливо корисним для кінцевих користувачів є інструментарій інтерактивної аналітики, можливість налаштовувати власні дашборди, а також отримувати сповіщення у реальному часі про ключові зміни у показниках.

Перспективи подальшого розвитку системи пов'язані з розширенням можливостей інтеграції та впровадженням інтелектуальних методів аналізу. Зокрема, у майбутньому доцільним є застосування алгоритмів машинного навчання для побудови персоналізованих рекомендацій щодо розподілу рекламного бюджету, прогнозування поведінки користувачів та виявлення прихованих закономірностей у даних. Це дозволить не лише реагувати на зміни у кампаніях, а й проактивно планувати їх стратегію.

Також перспективним напрямом є розширення мобільного функціоналу та інтеграція з системами управління бізнес-процесами, що зробить систему корисною не лише для маркетологів та аналітиків, а й для керівників компаній, які потребують швидкого доступу до ключових показників ефективності. У майбутньому система може стати основою для побудови комплексної екосистеми бізнес-аналітики, яка об'єднуватиме дані не лише з рекламних, але й з фінансових, виробничих та логістичних підсистем підприємства.

Таким чином, отримані результати підтверджують практичну цінність розробленого рішення, а подальший розвиток його функціоналу відкриває перспективи широкого використання у сфері цифрового маркетингу та управління бізнес-процесами.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ

4.1 Обґрунтування можливих чинників виникнення травмонебезпечних ситуацій

Офіс ІТ-компанії займає площу близько 200 м² та поділений на робочі зони і конференц-зали. Він обладнаний великою кількістю комп'ютерної техніки, серверів та іншого обладнання, а також системами вентиляції, кондиціонування й резервними джерелами живлення для безперервної роботи. Робочий простір розрахований на значну кількість співробітників, які працюють у форматі open space, забезпеченому всім необхідним оснащенням.

Для зручності працівників в офісі облаштована кухня з обладнанням для зберігання та підігріву їжі, а також приготування гарячих напоїв. Серед основних ризиків — електробезпека. Ймовірними є перевантаження електромережі, що особливо актуально в умовах нестабільної енергосистеми країни та у пікові періоди (в холодну чи спекотну погоду). Загрозу становлять пошкоджені або оголені дроти, відсутність заземлення обладнання та інші несправності.

Додатковим чинником є шкідливий вплив тривалого перебування за комп'ютером. Це може призводити до погіршення зору, болю у спині, а також розвитку синдрому зап'ясткового каналу при неправильному положенні рук.

Окремо слід виділити пожежну небезпеку, що може бути спричинена некоректним використанням кухонної техніки, несправністю електроприладів, акумуляторів або проводки. Підвищений ризик також

пов'язаний із роботою зовнішніх дизельних генераторів, які використовуються для резервного живлення.

Не менш важливою є хімічна небезпека, що може виникати під час задимлення приміщень у разі пожежі. Додатковим фактором є поширення неприємних запахів із кухні чи під час плавлення пластикових деталей обладнання при неправильній експлуатації.

Крім вище перевищених небезпек Слід окрему увагу приділити повітрю. Надмірне накопичення вуглекислого газу CO₂ у повітрі офісного приміщення площею 200 квадратних метрів є важливою проблемою охорони праці, що прямо впливає на здоров'я, самопочуття та продуктивність ІТ-співробітників. Хоча CO₂ сам по собі не є токсичним у невеликих концентраціях, його підвищений вміст слугує індикатором неякісного повітрообміну або несправності вентиляційної системи. Таблиця 4.1 показує вміст вуглекислого газу у повітрі та наслідки до яких це може призвести.

Таблиця 4.1 – можливі наслідки при певній концентрації CO₂ у повітрі.

CO ₂ ppm	Рівень	Наслідки
400 – 800	Оптимальний.	Висока якість повітря.
800 – 1000	Допустимий.	Можливе незначне зниження концентрації.
1000 - 1500	Низький.	З'являються втома, сонливість, зниження уваги, уповільнення реакції, головний біль. Зростає кількість помилок.
> 1500	Критичний.	Виражений головний біль, апатія, нудота, відчуття задухи. Збільшується ризик хронічної втоми та довгострокових порушень

4.2 Умови та фактори виникнення травмонебезпечних ситуацій та їхні наслідки

З метою підвищення рівня електробезпеки в офісі здійснюються регулярні технічні перевірки стану електромережі, а також проводиться інструктаж персоналу щодо правил безпечного користування електрообладнанням. Для запобігання аварійним ситуаціям встановлені реле контролю напруги та пристрої захисного відключення.

Для зменшення негативного впливу тривалої роботи за комп'ютером організовано періодичні перерви. У цьому допомагають спеціальні програмні засоби, які за таймером нагадують працівникам про необхідність зробити паузу в роботі.

Важливим елементом безпеки є систематичні тренінги з пожежної безпеки, під час яких працівників ознайомлюють з алгоритмами дій у надзвичайних ситуаціях. Вони включають практичні навички використання вогнегасників, правила поведінки з електроприладами, ознайомлення з евакуаційними виходами, сходовими клітками та контактами відповідальних осіб.

Для підтримання належної якості повітря та усунення небажаних запахів функціонує система вентиляції. Додатково встановлені датчики та газоаналізатори, зокрема для виявлення чадного газу, що дозволяє своєчасно попереджати про небезпеку, яка може бути непомітною для людини.

Урахування та контроль зазначених умов і чинників дозволяють значною мірою знизити ймовірність виникнення травмонебезпечних ситуацій, а також мінімізувати можливі ризики й негативні наслідки для діяльності компанії.

4.3 Безпека в умовах надзвичайних ситуацій

З огляду на можливість виникнення пожежі чи відключення електропостачання, в офісі розроблено план евакуації та встановлено системи оповіщення про аварійні ситуації. Працівники проходять тренування, що включають моделювання пожежних сценаріїв, а також навчаються користуванню вогнегасниками різних типів, які розташовані у приміщеннях.

Для підтримання роботи критично важливих серверів застосовуються блоки безперебійного живлення (on-line UPS), а зовнішні промислові генератори з функцією автозапуску та автоматичним керуванням забезпечують резервне живлення. Додатково передбачено альтернативні канали зв'язку та інтернету для гарантування стабільної комунікації.

Рівень безпеки постійно контролюється через систему моніторингу, аналіз інцидентів і коригування заходів відповідно до змін робочого середовища.

В умовах воєнних дій чи збройних конфліктів питання безпеки набуває особливої актуальності. ІТ-компанії повинні впроваджувати комплекс заходів, спрямованих на захист персоналу, інформаційних систем та інфраструктури. Це включає:

- ✓ Фізичну безпеку співробітників (зокрема організацію евакуації та рекомендації щодо дій у кризових ситуаціях);
- ✓ Цифровий захист даних і використання безпечних каналів комунікації;
- ✓ Охорону серверних приміщень і дата-центрів від несанкціонованого доступу.

Для працівників, які можуть перебувати в зоні ризику, мають бути розроблені чіткі інструкції та надані засоби індивідуального захисту. Компанія може організовувати навчання та тренінги, щоб відпрацювати

алгоритми реагування в умовах бойових дій чи інших непередбачуваних загроз.

У разі повітряних тривог працівникам, що перебувають в офісі, має бути забезпечений доступ до сертифікованих укриттів. Також необхідно передбачити можливість організації віддаленої роботи, що дозволить мінімізувати ризики для персоналу та водночас підтримати безперервність бізнес-процесів.

РОЗДІЛ 5

ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ

Ефективність розробленої інформаційної системи (GAA) оцінюється з позицій її відповідності поставленим завданням, зручності використання для кінцевих користувачів та здатності забезпечувати надійну й масштабовану аналітику рекламних кампаній. Для цього застосовувалися як кількісні, так і якісні показники.

Ефективність системи визначається рівнем автоматизації процесів збору та обробки даних. У ході апробації було підтверджено, що завдяки інтеграції з рекламними платформами та CRM-системами скорочується час на підготовку звітності. Якщо у традиційних умовах формування звіту займало від кількох годин до одного робочого дня, то після впровадження системи цей час зменшився до декількох хвилин. Це свідчить про суттєве підвищення продуктивності роботи аналітиків і маркетологів.

Оцінювалася точність і повнота даних. Автоматичне зіставлення інформації з різних джерел дозволило уникнути дублювань та втрат даних, що підвищує достовірність аналітики. Таким чином, користувачі отримують більш якісну інформаційну базу для прийняття управлінських рішень.

Важливим критерієм стала зручність для кінцевого користувача. Завдяки реалізації інтуїтивного інтерфейсу та інтерактивних дашбордів кінцеві користувачі можуть швидко орієнтуватися у ключових показниках ефективності рекламних кампаній. Опитування тестової групи користувачів показало, що більшість відзначає зменшення складності у щоденній роботі з даними та полегшення процесу прийняття рішень.

Також ефективність системи проявляється у її здатності до масштабування. Використання BigQuery та Apache Spark забезпечує

обробку великих масивів даних без зниження швидкодії, що дозволяє зберігати стабільність роботи при зростанні обсягів інформації. Це підтверджує доцільність використання системи у середніх та великих компаніях, де обсяги рекламних даних постійно збільшуються.

Нарешті, ефективність оцінюється за економічним критерієм. Завдяки автоматизації аналітичних процесів компанія зменшує витрати на ручну обробку даних, скорочує час підготовки звітів та мінімізує ймовірність помилок, які у традиційних умовах могли призводити до фінансових втрат. Таким чином, система створює додану вартість, забезпечуючи економію ресурсів і підвищення результативності рекламних кампаній.

Узагальнюючи, можна стверджувати, що ефективність розробленої системи проявляється у трьох вимірах: підвищення швидкості й точності обробки даних, зручності використання для кінцевих користувачів та позитивному економічному ефекті. Це робить систему перспективним інструментом для подальшого впровадження у сфері цифрового маркетингу.

ВИСНОВКИ

У ході виконання роботи було створено інформаційну систему для аналітики рекламних кампаній із використанням хмарних технологій, що продемонструвала високу ефективність та відповідність поставленим завданням. Система успішно поєднує функції збору, зберігання, обробки та візуалізації даних, забезпечуючи зручність використання для кінцевих користувачів і можливість інтеграції з різними зовнішніми платформами.

Одним із головних досягнень є створення комплексного середовища, здатного автоматично синхронізувати дані з рекламних платформ, CRM та інших інформаційних систем. Це дозволяє отримувати актуальну та повну картину ефективності маркетингових активностей, зменшуючи потребу у ручній роботі та мінімізуючи ризик помилок. Крім того, впроваджена можливість інтерактивної аналітики, яка дає змогу користувачам формувати власні дашборди, здійснювати глибокий аналіз та отримувати сповіщення у реальному часі.

Особливу увагу було приділено масштабованості та надійності системи. Завдяки використанню хмарних технологій BigQuery та Apache Spark забезпечується стабільна обробка великих обсягів даних без зниження швидкодії. Контейнеризація за допомогою Docker та оркестрація на базі Kubernetes гарантують гнучкість у розгортанні та подальшій підтримці, що відкриває перспективи використання системи як у середніх, так і у великих компаніях.

Перспективи розвитку системи пов'язані з подальшим розширенням функціональних можливостей. Зокрема, інтеграція алгоритмів машинного навчання дозволить не лише оцінювати ефективність минулих кампаній, але й прогнозувати результати майбутніх, а також надавати персоналізовані рекомендації щодо оптимізації бюджетів і вибору каналів просування. Ще

одним важливим напрямом є вдосконалення мобільного інтерфейсу, що забезпечить повноцінний доступ до аналітики з будь-якого пристрою.

Таким чином, створене рішення довело свою практичну цінність, забезпечуючи швидку, точну та зручну роботу з маркетинговими даними. Подальше вдосконалення системи дозволить перетворити її на універсальну платформу для бізнес-аналітики, яка зможе об'єднувати дані з різних сфер діяльності підприємства, сприяти прийняттю зважених управлінських рішень та підвищенню ефективності бізнес-процесів у цілому.

Окремо варто зауважити, що у контексті роботи доведено, що ми не маємо змоги повністю відійти від написання програмного коду ми можемо лише мінімізувати його кількість на основі використання готових або хмарних рішень, що у свою чергу може створювати перешкоди у досягненні головної мети.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Попович О.І. Маркетингові дослідження: теорія і практика. — Київ: Центр учбової літератури, 2021. — 180 с.
2. Котлер Ф., Келлер К. Маркетинг менеджмент. — 15-е вид. — СПб.: Питер, 2016. — 816 с.
3. Ryan D. Understanding Digital Marketing. — Kogan Page, 2020. — 368 p.
4. Clifton B. Advanced Web Metrics with Google Analytics. — Wiley, 2021. — 600 p.
5. Tuten T.L., Solomon M.R. Social Media Marketing. — SAGE, 2021. — 384 p.
6. Google Marketing Platform Help. Attribution models in Google Ads. URL: <https://support.google.com/google-ads/answer/6259715>
7. Facebook for Developers. Marketing API Overview. URL: <https://developers.facebook.com/docs/marketing-api/>
8. Provost F., Fawcett T. Data Science for Business. — O'Reilly, 2013. — 414 p.
9. Google Cloud. BigQuery Overview. URL: <https://cloud.google.com/bigquery>
10. Microsoft Azure. Data Factory Documentation. URL: <https://learn.microsoft.com/en-us/azure/data-factory/>
11. Kumar V., Gupta S. AI in Marketing, Sales and Service. — Springer, 2018. — 150 p.
12. Chaffey D., Ellis-Chadwick F. Digital Marketing. — Pearson Education, 2022. — 712 p.
13. IBM Cloud Education. What is AI personalization? URL: <https://www.ibm.com/cloud/learn/personalization>

14. Kotler P., Keller K.L. Marketing Management. — 15th ed. — Pearson Education, 2016. — p. 47–49.
15. Чекмарьов С.Ю. Інтернет-маркетинг. — Київ: КНЕУ, 2019. — 240 с.
16. Ryan D. Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation. — Kogan Page, 2020. — 368 p.
17. Філіпенко, А. С. Маркетинг: навчальний посібник. — Київ: Центр учбової літератури, 2020. — 256 с. — ISBN 978-617-673-895-7.
18. Котлер Ф., Келлер К. Л. Маркетинг менеджмент. — Київ: Дорчестер, 2013. — 816 с. — ISBN: 978-617-7009-06-4.
19. Воронкова А. Є. Медіапланування: навчальний посібник. — Київ: Центр учбової літератури, 2017. — 312 с. — ISBN: 978-617-673-473-8.
20. Understanding Cloud Computing: Benefits, Services, and Security. URL: <https://www.investopedia.com/terms/c/cloud-computing.asp>
21. The Role of Cloud Computing in Data Management & Analytics. URL: <https://www.koenig-solutions.com/blog/data-management-analytics>
22. Big data analytics in Cloud computing: an overview. Journal of Cloud Computing. URL: <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-022-00301-w>
23. Big Data Analytics in Cloud Computing: A Comparative Study of Different Approaches. ResearchGate. URL: https://www.researchgate.net/publication/390729533_Big_Data_Analytics_in_Cloud_Computing_A_Comparative_Study_of_Different_Approaches
24. Cloud Computing Guide for Big Data Storage and Processing. URL: <https://moldstud.com/articles/p-cloud-computing-for-big-data-storage-and-processing>

25. Amazon Kinesis. URL: https://en.wikipedia.org/wiki/Amazon_Kinesis
26. Azure Stream Analytics. URL: https://en.wikipedia.org/wiki/Azure_Stream_Analytics
27. The Role of Cloud Computing in Modern Data Analytics. URL: <https://www.linkedin.com/pulse/role-cloud-computing-modern-data-analytics-mnvff>
28. The Benefits of Using Cloud Computing for Big Data Analytics. Sikidang Blog. URL: <https://blog.sikidang.com/cloud-computing-for-big-data>
29. Big Data Analytics in Cloud Computing. Hero Vired. URL: <https://herovired.com/learning-hub/blogs/big-data-analytics-in-cloud-computing>
30. The Future of Cloud Computing and Big Data Analytics. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/role-of-cloud-computing-in-big-data-analytics>
31. Why it is vital that you understand the infrastructure behind AI. Financial Times. URL: <https://www.ft.com/content/8452bf94-9a41-4040-913f-ef1a462d6ea6>
32. What Is Database as a Service (DBaaS)? Everything You Need to Know to Choose a Provider. URL: <https://onlinedegrees.sandiego.edu/what-is-dbaas/>
33. Carlo Curino, Evan P. C. Jones, Raluca Ada Popa, Nirmesh Malviya, Eugene Wu, Sam Madden, Hari Balakrishnan, Nickolai Zeldovich. *Relational Cloud: A Database-as-a-Service for the Cloud*, 2011, c. 1–2. URL: <https://people.eecs.berkeley.edu/~raluca/cidr11.pdf>
34. Oleg Lukyanchikov, Evgeniy Pluzhnik, Simon Payain, Evgeny Nikulchev. *Using Object-Relational Mapping to Create the Distributed Databases in a Hybrid Cloud Infrastructure*, 2014, c. 61–64. URL: <http://dx.doi.org/10.14569/IJACSA.2014.051208>

35. Wisal Khan, Teerath Kumar, Zhang Cheng, Kislay Raj, Arunabha M. Roy, Bin Luo. *SQL and NoSQL Databases Software architecture Performance Analysis and Assessments — A Systematic Literature Review*, 2022, c. 2–3. URL: <https://arxiv.org/pdf/2209.06977>
36. Amazon Web Services. AWS Analytics Services Overview. URL: <https://aws.amazon.com/analytics/>
37. Microsoft Azure. Azure Synapse Analytics Documentation. URL: <https://learn.microsoft.com/en-us/azure/synapse-analytics/>
38. Google Cloud. BigQuery and Data Analytics. URL: <https://cloud.google.com/bigquery>
39. Google Analytics. GA4 Documentation. URL: <https://support.google.com/analytics/answer/10089681>
40. Meta for Business. Ads Manager. URL: <https://www.facebook.com/business/tools/ads-manager>
41. Zapier. Automation Platform. URL: <https://zapier.com/>
42. MySQL. MySQL Reference Manual. Oracle, 2023. URL: <https://dev.mysql.com/doc/>
43. Apache Software Foundation. Apache Spark Overview. URL: <https://spark.apache.org/>
44. TensorFlow. Machine Learning Platform. URL: <https://www.tensorflow.org/>
45. Tableau. Tableau Product Overview. URL: <https://www.tableau.com/>
46. Google. Looker Studio (ex-Data Studio). URL: <https://lookerstudio.google.com/>
47. Microsoft. Power BI Overview. URL: <https://powerbi.microsoft.com/>
48. Google Cloud. BigQuery Documentation. URL: <https://cloud.google.com/bigquery/docs>

49. Snowflake Inc. Snowflake Cloud Data Platform. URL: <https://www.snowflake.com/>
50. Vertica. Vertica Analytics Platform. URL: <https://www.vertica.com/>
51. ClickHouse. ClickHouse Documentation. URL: <https://clickhouse.com/docs>
52. App Engine Application Platform | Google Cloud. URL: <https://cloud.google.com/appengine>
53. Ковалевич М. Розробка інформаційної системи для аналітики рекламних кампаній із використанням хмарних технологій. Тези доповідей студентської конференції факультету механіки, енергетики та інформаційних технологій «Дні студентської науки» 14–15 травня 2025 р. м. Львів / Львів. нац. ун-т вет. мед. та біотехн. ім. С. З. Гжицького. Львів, 2025. С. 65.
54. Customer-service-dashboard.webp. URL: <https://cdn.boldbi.com/wp/blogs/analytics-as-a-service/customer-service-dashboard.webp>
55. SQL проти NoSQL – різниця між ними. URL: <https://www.guru99.com/uk/sql-vs-nosql.html>
56. Tabolla Ads Realize Performance. URL: <https://ads.realizeperformance.com>
57. Google Ads. URL: <https://ads.google.com/>
58. Yahoo! Demand Side Platform. URL: <https://dsp.yahooinc.com/>

ДОДАТКИ

Додаток А

Приклад конфігурації Spring Security у контексті REST API

1. Конфігурація безпеки.

```
@Configuration
@EnableWebSecurity
public class SecurityConfig {

    private final JwtAuthenticationFilter jwtAuthenticationFilter;
    private final UserDetailsService userDetailsService;
    private final PasswordEncoder passwordEncoder;

    public SecurityConfig(JwtAuthenticationFilter jwtAuthenticationFilter,
                        UserDetailsService userDetailsService,
                        PasswordEncoder passwordEncoder) {
        this.jwtAuthenticationFilter = jwtAuthenticationFilter;
        this.userDetailsService = userDetailsService;
        this.passwordEncoder = passwordEncoder;
    }

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
            .csrf(AbstractHttpConfigurer::disable)
            .authorizeHttpRequests(auth -> auth
                .requestMatchers("/api/auth/**").permitAll()
                .anyRequest().authenticated()
            )
            .sessionManagement(session -> session.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
            .authenticationProvider(authenticationProvider())
            .addFilterBefore(jwtAuthenticationFilter, UsernamePasswordAuthenticationFilter.class);

        return http.build();
    }

    @Bean
    public AuthenticationProvider authenticationProvider() {
        DaoAuthenticationProvider provider = new DaoAuthenticationProvider();
        provider.setUserDetailsService(userDetailsService);
        provider.setPasswordEncoder(passwordEncoder);
        return provider;
    }

    @Bean
    public AuthenticationManager authenticationManager(AuthenticationConfiguration config) throws Exception {
        return config.getAuthenticationManager();
    }

    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
}
```

2 Код у application.properties або application.yml.

```
app.jwtSecret=your_super_secret_jwt_key_here
app.jwtExpirationInMs=3600000 # 1 hour
```

3. Конфігурація JWT фільтра.

```

@Component
public class JwtAuthenticationFilter extends OncePerRequestFilter {

    private final JwtTokenProvider jwtTokenProvider;
    private final UserDetailsService userDetailsService;

    public JwtAuthenticationFilter(JwtTokenProvider jwtTokenProvider, UserDetailsService userDetailsService) {
        this.jwtTokenProvider = jwtTokenProvider;
        this.userDetailsService = userDetailsService;
    }

    @Override
    protected void doFilterInternal(HttpServletRequest request,
                                    HttpServletResponse response,
                                    FilterChain filterChain) throws ServletException, IOException {

        String token = getJwtFromRequest(request);

        if (StringUtils.hasText(token) && jwtTokenProvider.validateToken(token)) {
            String username = jwtTokenProvider.getUsernameFromToken(token);
            UserDetails userDetails = userDetailsService.loadUserByUsername(username);
            UsernamePasswordAuthenticationToken authentication = new UsernamePasswordAuthenticationToken(
                userDetails, null, userDetails.getAuthorities());
            authentication.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
            SecurityContextHolder.getContext().setAuthentication(authentication);
        }
        filterChain.doFilter(request, response);
    }

    private String getJwtFromRequest(HttpServletRequest request) {
        String bearerToken = request.getHeader("Authorization");
        if (StringUtils.hasText(bearerToken) && bearerToken.startsWith("Bearer ")) {
            return bearerToken.substring(7);
        }
        return null;
    }
}

```

4 UserDetailsService

```

@Service
public class UserDetailsServiceImpl implements UserDetailsService {

    @Autowired
    private UserRepository userRepository; // Your data source repository

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        User user = userRepository.findByUsername(username)
            .orElseThrow(() -> new UsernameNotFoundException("User Not Found with username: " + username));

        // Returns a Spring Security UserDetails object
        return new org.springframework.security.core.userdetails.User(
            user.getUsername(),
            user.getPassword(), // Hashed password from DB
            user.getAuthorities() // User roles/permissions
        );
    }
}

```

5 JWT provider

```
@Component
public class JwtTokenProvider {

    @Value("${app.jwtSecret}")
    private String jwtSecret;

    @Value("${app.jwtExpirationInMs}")
    private int jwtExpirationInMs;

    public String generateToken(Authentication authentication) {
        UserDetails userDetails = (UserDetails) authentication.getPrincipal();
        Date now = new Date();
        Date expiryDate = new Date(now.getTime() + jwtExpirationInMs);

        return Jwts.builder()
            .setSubject(userDetails.getUsername())
            .setIssuedAt(new Date())
            .setExpiration(expiryDate)
            .signWith(SignatureAlgorithm.HS512, jwtSecret)
            .compact();
    }

    public String getUsernameFromToken(String token) {
        Claims claims = Jwts.parser()
            .setSigningKey(jwtSecret)
            .parseClaimsJws(token)
            .getBody();
        return claims.getSubject();
    }

    public boolean validateToken(String authToken) {
        try {
            Jwts.parser().setSigningKey(jwtSecret).parseClaimsJws(authToken);
            return true;
        } catch (JwtException ex) {
            // Log exception
        }
        return false;
    }
}
```

Додаток Б

Приклад складного SQL запита.

```

WITH aggregated_metrics AS (
    SELECT
        campaign_id,
        SUM(clicks) AS clicks,
        SUM(impressions) AS impressions,
        SUM(spent) AS spent,
        AVG(roas) AS average_roas
    FROM
        mertics.campaign_metrics
    WHERE
        date BETWEEN :start_date AND :end_date
    GROUP BY
        campaign_id
)
SELECT
    c.name AS campaign_name,
    co.name AS owner_name,
    am.clicks,
    am.impressions,
    am.spent,
    am.average_roas
FROM
    main.campaign c
JOIN
    main.campaign_owners co ON c.owner_id = co.id
JOIN
    aggregated_metrics am ON c.id = am.campaign_id
JOIN
    main.campaign_targeting ct ON c.id = ct.campaign_id
WHERE
    co.id = :owner_id
GROUP BY
    c.name,
    co.name,
    am.clicks,
    am.impressions,
    am.spent,
    am.average_roas
ORDER BY
    co.name,
    c.name;

```


Додаток В

Приклад інтеграції AWS S3 на основі Java

1. Базове підключення до S3 (S3Client).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;

public class S3Connection {

    public static S3Client createS3Client(String regionName) {
        Region region = Region.of(regionName);

        S3Client s3Client = S3Client.builder()
            .region(region)
            .build();

        return s3Client;
    }

    public static void main(String[] args) {
        String bucketRegion = "us-east-1";
        S3Client s3 = createS3Client(bucketRegion);

        s3.listBuckets().buckets().forEach(bucket -> {
            System.out.println("Bucket Name: " + bucket.name());
        });

        s3.close();
    }
}
```

2. Явне надання облікових даних.

```
import software.amazon.awssdk.auth.credentials.AwsBasicCredentials;
import software.amazon.awssdk.auth.credentials.StaticCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;

S3Client s3Client = S3Client.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(StaticCredentialsProvider.create(
        AwsBasicCredentials.create("YOUR_ACCESS_KEY", "YOUR_SECRET_KEY")))
    .build();
```

3. Налаштування HTTP-проксі.

```
import java.net.URI;
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;

S3Client s3Client = S3Client.builder()
    .region(Region.US_EAST_1)
    .overrideConfiguration(ClientOverrideConfiguration.builder()
        .proxyConfiguration(p -> p.endpoint(URI.create("http://your.proxy.com:8080")))
        .build())
    .build();
```

4. Кастомна кінцева точка.

```
import java.net.URI;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;

S3Client s3Client = S3Client.builder()
    .region(Region.US_EAST_1)
    .endpointOverride(URI.create("http://localhost:4566"))
    .build();
```

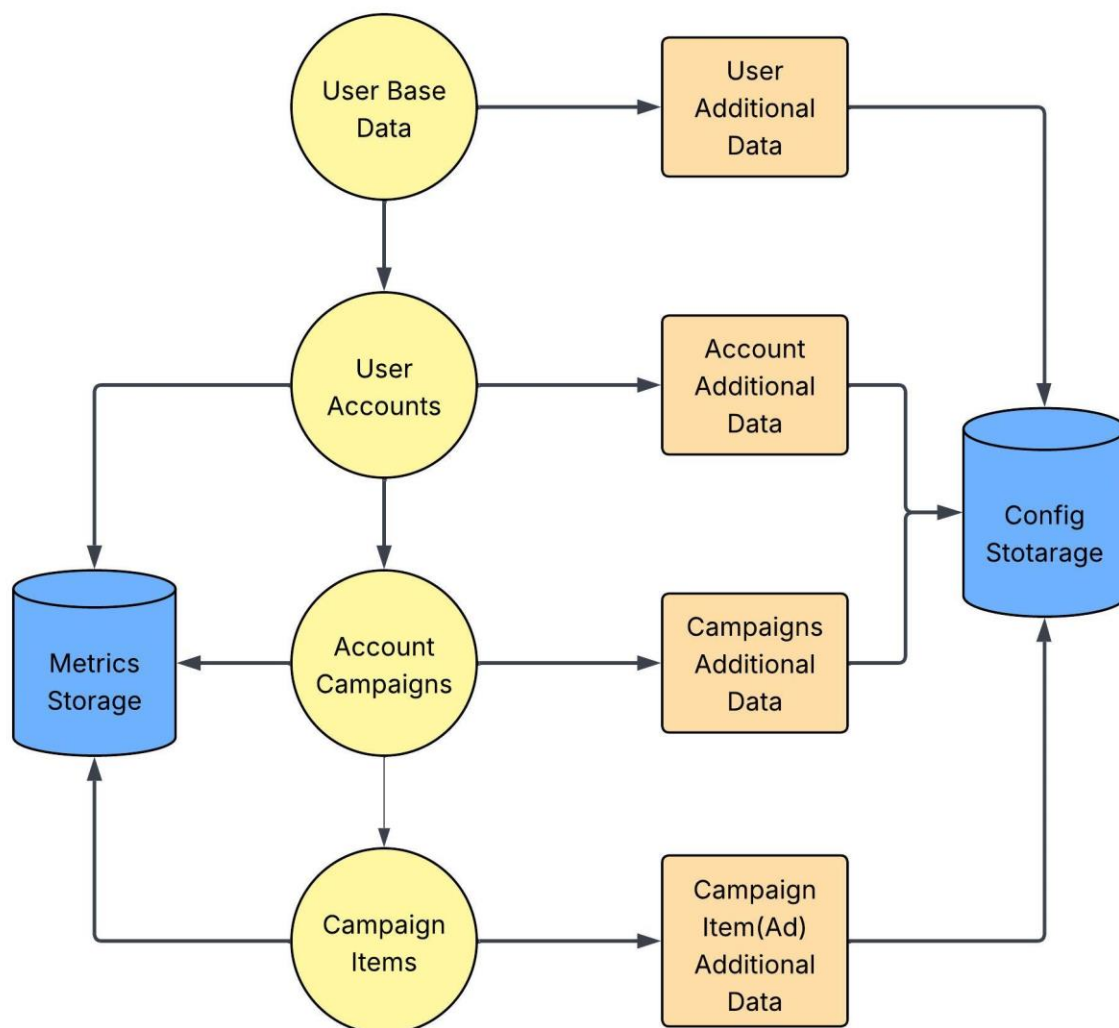
5. Таймаути та політика повторних спроб.

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.core.retry.RetryPolicy;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import java.time.Duration;

S3Client s3Client = S3Client.builder()
    .region(Region.US_EAST_1)
    .overrideConfiguration(ClientOverrideConfiguration.builder()
        .apiCallTimeout(Duration.ofSeconds(10))
        .apiCallAttemptTimeout(Duration.ofSeconds(5))
        .retryPolicy(RetryPolicy.defaultRetryPolicy())
        .build())
    .build();
```

Додаток Г

Приклад схеми рекламної кампанії у базі даних



Додаток Д
Узагальнена модель структури бази даних для рекламних кампаній

