

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

Кафедра комп'ютерних наук
Напрямок підготовки 122 – Комп'ютерні науки

ЗВІТ
Про проходження виробничої практики

На базі: _____ ТОВ «ScalHive» _____

(згідно договору про виробничу практику)

Дата захисту: „___” _____ 2020 р. Оцінка: _____ Прийняв: _____ Допущено до захисту: „___” _____ 2020 р. З оцінкою: _____ _____	Виконав: студент групи <u>СНс-32</u> _____ <u>Лісовський Владислав Володимирович</u> Залікова книжка: _____ Керівник практики: від підприємства _____ (м.п., підпис) від ВНЗ _____ (підпис)
--	--

Тернопіль, 2020

ЗАВДАННЯ НА виробничу ПРАКТИКУ
(вид практики)

1. Назва задачі (модуля) Ефективність алгоритмів. Google Cloud Platform.
Аналіз в BigQuery.

2. Термін здачі звіту 10.09.2020

3. Дата видачі завдання 06.07.2020

4. Вхідні дані до задачі (модуля): ДСТУ з опрацювання інформації, літературні джерела, технічна документація на АІС _____, матеріали практики.

5. Перелік графічного матеріалу _____

Керівник від ВНЗ _____
(підпис) (посада, П.І.Б.)

Студент _____
(підпис) Лісовський В.В.
(П.І.Б.)

ЗМІСТ

ВСТУП	4
1 ХАРАКТЕРИСТИКА ТА ОЗНАЙОМЛЕННЯ З БАЗОЮ ПРАКТИКИ.....	5
1.1 Загальний огляд бази практики	5
1.2 Структура компанії	5
1.3 Характеристика послуг.....	6
1.4 Оснащення технічного і програмного забезпечення робочого місця .	7
2 АНАЛІЗ ТА ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ	10
2.1 Постановка завдання.....	10
2.2 Розгляд платформи Onlizer	10
2.3 Алгоритм. Ефективність алгоритму. Метод Карацуби.....	13
2.4 Розгляд хмарної платформи Google Cloud Platform	15
2.4.1 Хмарні служби	15
2.4.2 Сервіси GCP.....	19
2.5 Обробка набору аналітичних даних.....	22
ВИСНОВОК.....	26
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	27
ДОДАТКИ.....	29

ВСТУП

Отримання практичних знань для студента являється однією з найважливіших умов будь-якого навчального закладу. Виробнича практика і є однією з таких засобів професійної підготовки майбутнього спеціаліста.

Метою практики є вивчення і оволодіння студентами формами та методами роботи з комплексами задач, з організаціями обраної сфери праці, використанням комп'ютерної техніки та інформаційних технологій для розв'язання поставлених завдань. А також для поглиблення та закріплення теоретичних знань, отриманих під час навчання, і для набуття професійних вмінь та навичок.

Під час проходження виробничої практики студент працює над поставленим індивідуальним завданням на своєму робочому місці, дотримуючись усіх правил, інструкцій та вимог з техніки безпеки та охорони праці.

Студент знайомиться з обраним підприємством, його структурою та сферою діяльності, аналізує і закріплює весь процес виготовлення продукції. Здобуває досвід швидкого і раціонального використання доступних інформаційно-технологічних та інтелектуальних ресурсів для виконання поставленої цілі.

1 ХАРАКТЕРИСТИКА ТА ОЗНАЙОМЛЕННЯ З БАЗОЮ ПРАКТИКИ

1.1 Загальний огляд бази практики

Базою для проходження практики була обрана тернопільська компанія ScalHive, повна назва – Товариство з обмеженою відповідальністю ScalHive, засноване у 2015 році. Адреса бази практики – вулиця Бережанська, 10, м. Тернопіль. Веб-сайт знаходиться за посиланням [9].

Компанія займається наданням аутсорсингових ІТ послуг і зосереджує увагу на розробці індивідуальних рішень, які відповідають унікальним бізнес-вимогам клієнтів.

Логотип, який виступає основною торгівельною маркою, зображений на рисунку 1.1.



Рисунок 1.1 – Логотип «ScalHive»

1.2 Структура компанії

Структура «ScalHive» складається з наступних відділів:

- адміністративного – відповідає за організацію і функціонування інфраструктури підприємства;
- фінансового – відповідає за організацію, контроль та розвиток фінансів підприємства;
- стратегічного – відповідає за визначення цілей, планів та можливостей підприємства;

- інженерного – відповідає за обслуговування пристроїв, їх роботу та тестування;
- комерційного – відповідає за розробку стратегії продажів і тактики для отримання максимального прибутку, а також налагодження та підтримку стосунків з потенційними та існуючими клієнтами;
- інформаційного – відповідає за внутрішню ефективність компанії, якісні та стійкі бізнес покращення;
- інформаційно-технологічного – відповідає за розробку і покращення продуктів, архітектур, інфраструктури та платформ;
- кадрів – відповідає за якісні умови роботи персоналу, їх задоволення, мотивації та нагород, і за набір нових кадрів.

1.3 Характеристика послуг

Компанія «ScalHive» являється аутсорсинговою, тобто вона надає замовникам послуги з розробки програмних продуктів, консультування та забезпечення контролю якості.

До послуг компанії належать:

- з програмної інженерії;
- з технічної консультації;
- з перевірки якості;
- з кібербезпеки.

В якості послуг з програмної інженерії «ScalHive» пропонує розробку для вирішення проблем всіх типів платформ – веб, мобільних пристроїв та настільних ПК. А саме створення програмного забезпечення з ідеї, покращення існуючого з додавання модифікацій або модернізації самої системи. «ScalHive» забезпечує весь стек розробки продукту, його впровадження на ринок, підтримку та обслуговування.

Однією з найбільш популярних послуг є перевірка якості та тестування. Компанія має окрему команду з QA інженерів, які забезпечать надійну і якісну перевірку стороннього забезпечення на його ефективність та захищеність. Перевірка здійснюється на всіх платформах – мобільних, настільних ПК, веб-додатків та клієнт-серверних додатків. «ScalHive» пропонує зручну і ефективну автоматизацію перевіркової рутини.

Іншою послугою являється надання технічної консультації. «ScalHive» пропонує:

- надання технічної консультації;
- огляд архітектури
- огляд коду;
- огляд дизайну;
- тренування та наставництво з Scala та реактивної платформи Lightbend.

Ще одним пунктом являється надання послуг щодо захисту продукції від кіберзагроз, шлях моніторингу захисту системи, тестуванням існуючого захисту на безпечність, визначенням і усуненням всіх ризиків безпеки та вимог до відповідності.

1.4 Оснащення технічного і програмного забезпечення робочого місця

Представлене робоче місце оснащено комп'ютерною технікою.

Комп'ютерна техніка — це пристрої та системи пристроїв, які є компонентами ПК, а також самі персональні комп'ютери та обладнання, яке працює спільно з ними і забезпечує деяку додаткову функціональність.

Характеристика обладнання, яке застосовується на робочому місці:

1. Монітор (див. рис. 1.2):



Рисунок 1.2 – Зовнішній вигляд монітора

Таблиця 1.1 – Характеристики монітора

Модель	PHILIPS 242V8A/00
Тип матриці	IPS
Діагональ дисплея	23.8
Роздільна здатність	1920x1080
Інтерфейс підключення	HDMI , DisplayPort , Аудіо вхід , VGA
Час відгуку матриці	4 мс
Кути огляду, гор/верт	178

2. Системний блок (див. рис. 1.3):



Рисунок 1.3 – Зовнішній вигляд системного блоку

Таблиця 1.2 – Характеристики системного блоку

Модель	Everest Home&Office 1030
Материнська плата	B250M PRO-VH
Процесор	4х ядерний Intel Core i5-7400, 3 ГГц
Оперативна пам'ять	12 ГБ 2400 МГц, 2 слоти DDR4
Відеокарта	MSI NVIDIA GTX 1660 Ti, 6 Гб
Жорсткий диск	Toshiba HDWD110, 1 TB
SSD	TeamElite 256 Гб
Чіпсет материнської плати	Intel H110
Слоти розширення	1 x PCI-E 3.0 x16, 2 x PCI-E 3.0 x1
Потужність БЖ	400 Вт

2 АНАЛІЗ ТА ВИКОНАННЯ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

2.1 Постановка завдання

Для проходження практики в компанії ScalHive були надані такі завдання:

- Аналіз програмного забезпечення Onlizer.
- Проходження курсу Google Cloud Certified Professional Cloud Architect.
- Проходження першого тижня курсу «Як розуміти ефективність алгоритмів» і виконання практичного завдання – реалізації алгоритму Карацуби.
- Розгляд аналітичних даних запропонованого набору даних, створення SQL-скрипта для обробки даних, що містять спеціальні параметри, в Google Cloud Platform.

Суть завдань – покращення навичок програмування, розробок ефективних алгоритмів, вивчення поняття ефективності алгоритму, ознайомлення з хмарною платформою Google Cloud Platform, робота з датасетами, їх аналіз та обробка в BigQuery.

2.2 Розгляд платформи Onlizer

Onlizer – це платформа для інтеграції сервісів і автоматизації процесів, не використовуючи коду, або з використанням псевдокоду. Схожий клас програмних рішень називають Cloud-based Integrated.

Підтримує наступні сервіси:

- Amazon DynamoDB, RedShift, S3, SES.
- Apache HBase, Hive.
- Azure Blob Storage, DocumentDB, Notifications Hubs, Scheduler, Table Storage.

- Bitrix24.
- Crowdin.
- Data Storage, Date and Time Operations, CSV files, FTP, Git.
- Dropbox, Evernote.
- Ebay.
- Google Services.
- LinkedIN, Telegram, Messenger, Facebook.
- Microsoft Services.
- DBs – MySQL, MongoDB.
- І багато інших сервісів.

Приклад доступних сервісів та їх можливих зв'язків зображено на рисунку 2.1.

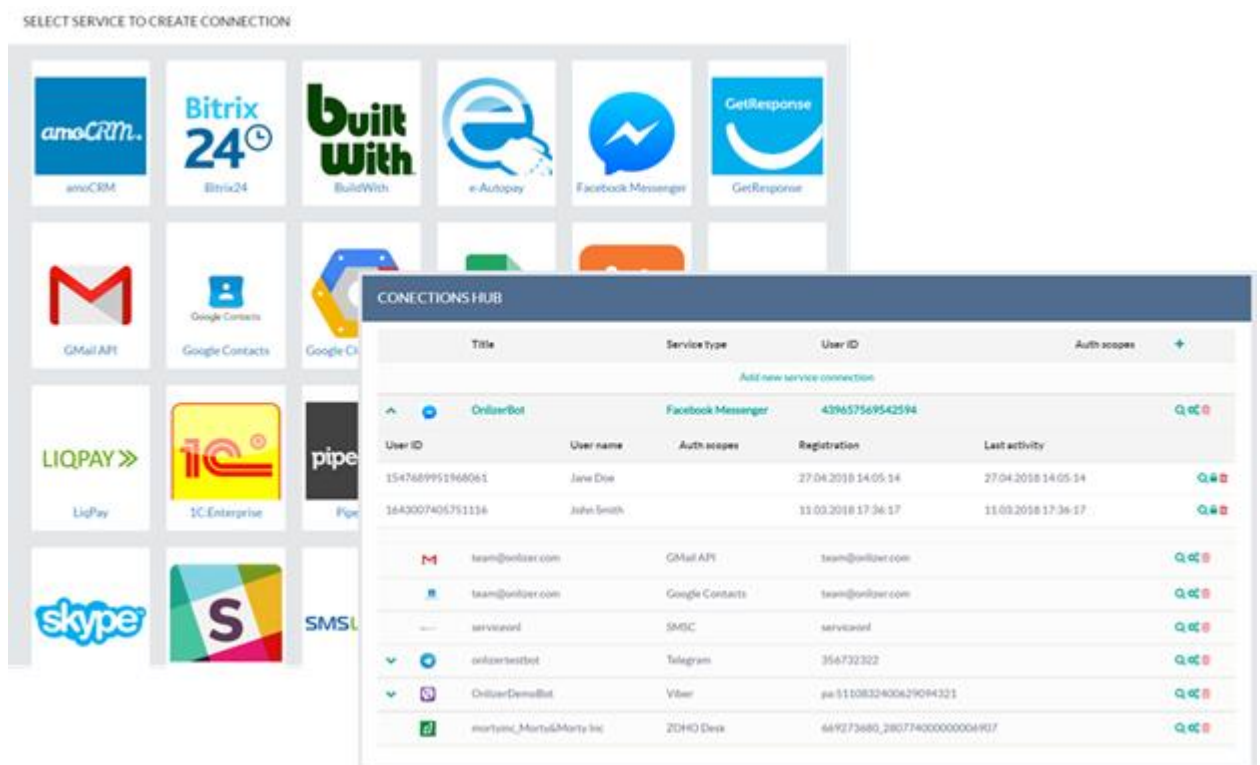


Рисунок 2.1 – Приклад доступних сервісів та їх зв'язків

У веб-додатку дає змогу створити інтеграції сервісів – об'єднати певну кількість, прив'язавши до них тригери і дії. Для прикладу, можна

об'єднати Телеграм + Пошту (див. рис. 2.2), щоб при отриманні ботом повідомлення, він надсилав його на пошту, створював подію в календарі, або виконував інші дії, які залежать від вибору сервісу.

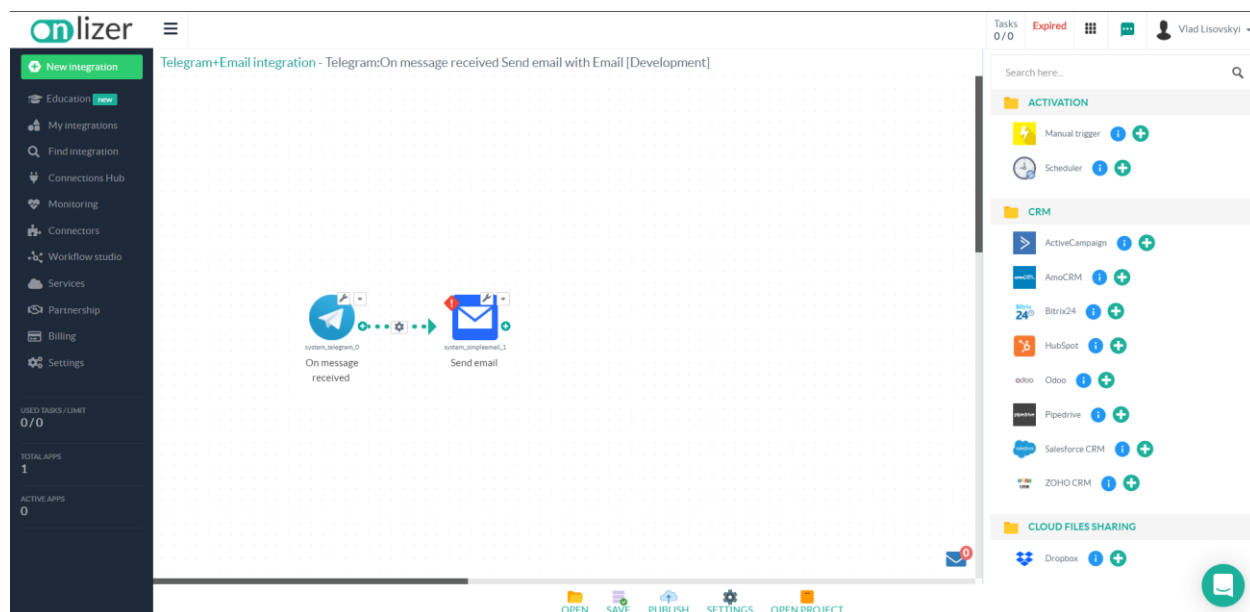


Рисунок 2.2 – Об'єднання Телеграм + Пошта

При виборі інтеграції та прив'язуванні її до іншого сервісу, збоку з'являється панель опису можливих дій (див. рис. 2.3).

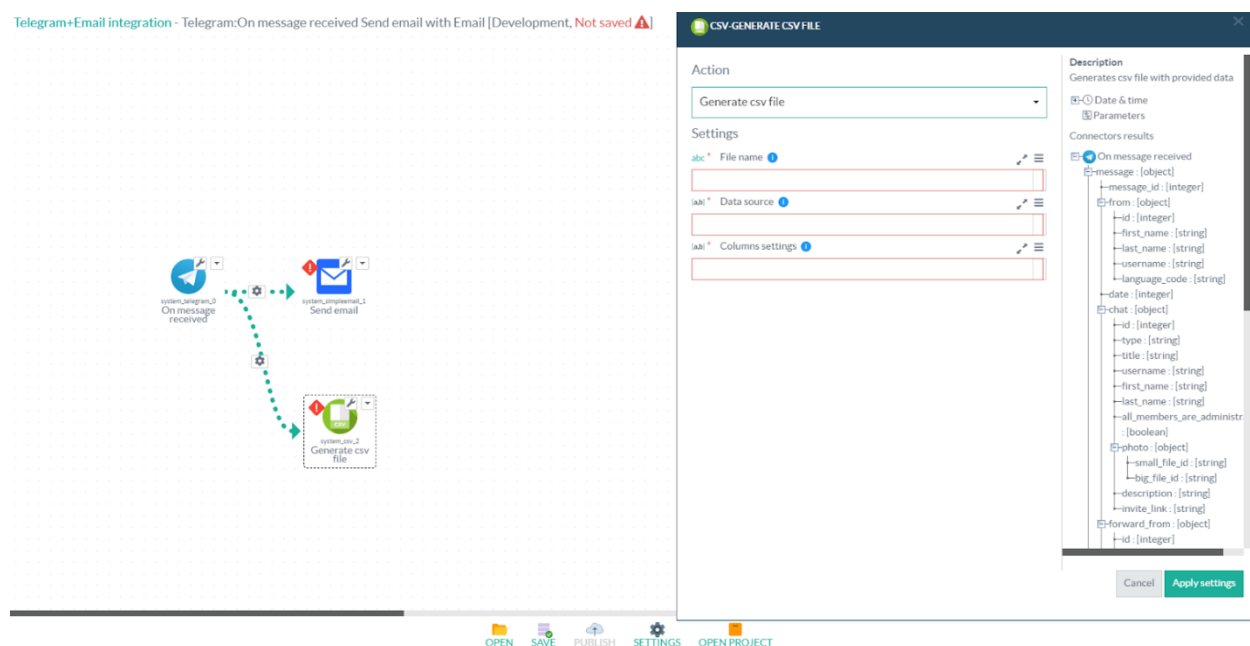


Рисунок 2.3 – Панель програмованих дій

2.3 Алгоритм. Ефективність алгоритму. Метод Карацуби

Алгоритм – набір систематизованих правил виконання обчислювального процесу, що обов'язково призводить до розв'язання певного класу задач після скінченного числа операцій.

Алгоритм – це довільна, коректна обчислювальна процедура, на вхід якої подається деяка величина або набір величин, а результатом виконання якої є вихідна величина або набір значень.

Можна навести загальні риси алгоритму:

- a. Дискретність інформації. Кожен алгоритм має справу з даними: вхідними, проміжними, вихідними. Ці дані представляються у вигляді скінчених слів деякого алфавіту.
- b. Дискретність роботи алгоритму. Алгоритм виконується по кроках та при цьому на кожному кроці виконується тільки одна операція.
- c. Детермінованість алгоритму. Система величин, які отримуються в кожний (не початковий) момент часу, однозначно визначається системою величини, які були отримані в попередні моменти часу.
- d. Елементарність кроків алгоритму. Закон отримання наступної системи величин з попередньої повинен бути простим та локальним.
- e. Виконуваність операцій. В алгоритмі немає бути не виконуваних операцій. Наприклад, не можна в програмі призначити значення змінній «нескінченність», така операція була би не виконуваною. Кожна операція опрацьовує певну ділянку у слові, яке обробляється.
- f. Скінченність алгоритму. Опис алгоритму повинен бути скінченим.
- g. Спрямованість алгоритму. Якщо спосіб отримання наступної величини з деякої заданої величини не дає результату, то має бути вказано, що треба вважати результатом алгоритму.

h. Масовість алгоритму. Початкова система величин може обиратись з деякої потенційно нескінченної множини.

Ефективність алгоритму — це властивість алгоритму, пов'язана з обчислювальними ресурсами, використовуваними алгоритмом. Алгоритм повинен бути проаналізований з метою визначення необхідних йому ресурсів. Ефективність алгоритму можна розглядати як аналог виробничої продуктивності повторюваних або безперервних процесів.

Для досягнення максимальної ефективності бажано зменшити використання ресурсів. Однак різні ресурси (такі як час і пам'ять) не можна порівняти безпосередньо, тому який із двох алгоритмів вважати більш ефективним часто залежить від того, який фактор важливіший, наприклад, вимога високої швидкості, мінімального використання пам'яті чи інша міра ефективності. [8]

Алгоритм роботи методу Карацуби зображений на рисунку 2.4.

Метод Карацуби

$$X_1 X_2 = 10^n \cdot ac + 10^{n/2} \cdot (ad + bc) + bd$$

$$(a+b) \cdot (c+d) = ac + \underline{ad + bc} + bd$$

$$ad + bc = \frac{(a+b) \cdot (c+d) - ac - bd}{\sim n/2 \quad \sim n/2 \quad \sim n/2}$$

$$X_1 X_2 = 10^n \cdot \textcircled{ac} + 10^{n/2} \cdot \left[\underline{(a+b) \cdot (c+d)} - ac - bd \right] + \textcircled{bd}$$

$$T(n) = 3n^{\log_2 3} \approx 3n^{1.585}$$

Рисунок 2.4 – Логіка методу Карацуби

Приклад рекурсивної реалізації методу Карацуби знаходиться в лістингу 2.1.

Лістинг 2.1 – Рекурсивна реалізація методу Карацуби

```
def karatsuba_tail_rec(x, y, sum_l=None):
    if sum_l is None:
        sum_l = []

    if len(str(x)) == 1 and len(str(y)) == 1:
        return x*y, sum_l

    n = max(len(str(x)), len(str(y)))
    n2 = n//2
    a, b = x//10**n2, x%10**n2
    c, d = y//10**n2, y%10**n2

    ac, _ = karatsuba_tail_rec(a, c, sum_l)
    bd, _ = karatsuba_tail_rec(b, d, sum_l)

    ab_cd, _ = karatsuba_tail_rec(a+b, c+d, sum_l)
    mult_sum = ab_cd - ac - bd
    sum_l.append(mult_sum)
    return (10**n) * ac + (10**n2) * mult_sum + bd, sum_l
```

2.4 Розгляд хмарної платформи Google Cloud Platform

Google Cloud Platform – запропонований компанією Google набір хмарних служб, які виконуються на тій же самій інфраструктурі, яку Google використовує для своїх продуктів призначених для кінцевих споживачів, таких як Google Search та YouTube. [6]

2.4.1 Хмарні служби

Хмарні служби або хмарне обчислення - модель забезпечення повсюдного та зручного доступу на вимогу через мережу до спільного пулу обчислювальних ресурсів, що підлягають налаштуванню (наприклад, до комунікаційних мереж, серверів, засобів збереження даних, прикладних програм та сервісів), і які можуть бути оперативно надані та звільнені з мінімальними управлінськими затратами та зверненнями до провайдера. [12]

Міжнародні стандарти NIST складають наступні основні характеристики хмарного обчислення [10, 4]:

1. Самообслуговування по запиту (On-Demand Self-Service) – споживач самостійно визначає і змінює обчислювальні потреби, такі як

серверний час, швидкості доступу та обробки даних, обсяг збережених даних без взаємодії з представником постачальника послуг.

2. Багатоклієнтський загальний пул ресурсів (Resources Pooling) – постачальник послуг об'єднує ресурси для обслуговування великої кількості споживачів в єдиний пул для динамічного перерозподілу потужностей між споживачами в умовах постійної зміни попиту на потужності; при цьому споживачі контролюють тільки основні параметри послуги (наприклад, обсяг даних, швидкість доступу), але фактичний розподіл ресурсів, що надаються споживачеві, здійснює постачальник (в деяких випадках споживачі все ж можуть керувати деякими фізичними параметрами перерозподілу, наприклад, вказувати бажаний центр обробки даних з міркувань географічної близькості).

3. Миттєва еластичність (Rapid Elasticity) – послуги можуть бути надані, розширені, звужені в будь-який момент часу, без додаткових витрат на взаємодію з постачальником, як правило, в автоматичному режимі.

4. Універсальний доступ до мережі (Broad Network Access) – послуги доступні споживачам через мережу передачі даних незалежно від термінального пристрою.

5. Облік споживання послуг (Measured Service) – постачальник послуг автоматично обчислює спожиті ресурси на певному рівні абстракції (наприклад, обсяг збережених даних, пропускна здатність, кількість користувачів, кількість транзакцій), і на основі цих даних оцінює обсяг наданих споживачам послуг.

Модель хмарного розгортання - це специфічна конфігурація параметрів середовища, таких як доступність та власність інфраструктури розгортання та розмір сховища. Це означає, що типи розгортання різняться залежно від того, хто контролює інфраструктуру та де вона знаходиться. [2]

Є 4 типи моделей розгортання.

1. Публічна хмара (public cloud) — це хмарна інфраструктура, яка призначена для вільного використання широким загалом. Публічна хмара

може перебувати у власності, керуванні та експлуатації комерційних, академічних (освітніх та наукових) або державних організацій (чи будь-якої їх комбінації). Публічна хмара перебуває в юрисдикції постачальника хмарних послуг.

2. Приватна хмара (private cloud) — це хмарна інфраструктура, яка призначена для використання виключно однією організацією, що включає декілька користувачів (наприклад, підрозділів). Приватна хмара може перебувати у власності, керуванні та експлуатації як самої організації, так і третьої сторони (чи деякої їх комбінації). Така хмара може фізично знаходитись як в, так і поза юрисдикцією власника.

3. Гібридна хмара (hybrid cloud) — це хмарна інфраструктура, що складається з двох або більше різних хмарних інфраструктур (приватних, громадських або публічних), які залишаються унікальними сутностями, але з'єднані між собою стандартизованими або приватними технологіями, що уможливають переносимість даних та прикладних програм (наприклад, використання ресурсів публічної хмари для балансування навантаження між хмарами).

4. Громадська хмара (community cloud) – надає виключне право на використання хмарних обчислень відповідним спільнотам, які мають загальні задачі.

Компанії надання хмарних сервісів надають свої послуги за трьома моделями обслуговування, трьома шарами абстракції (див. рис. 2.5).

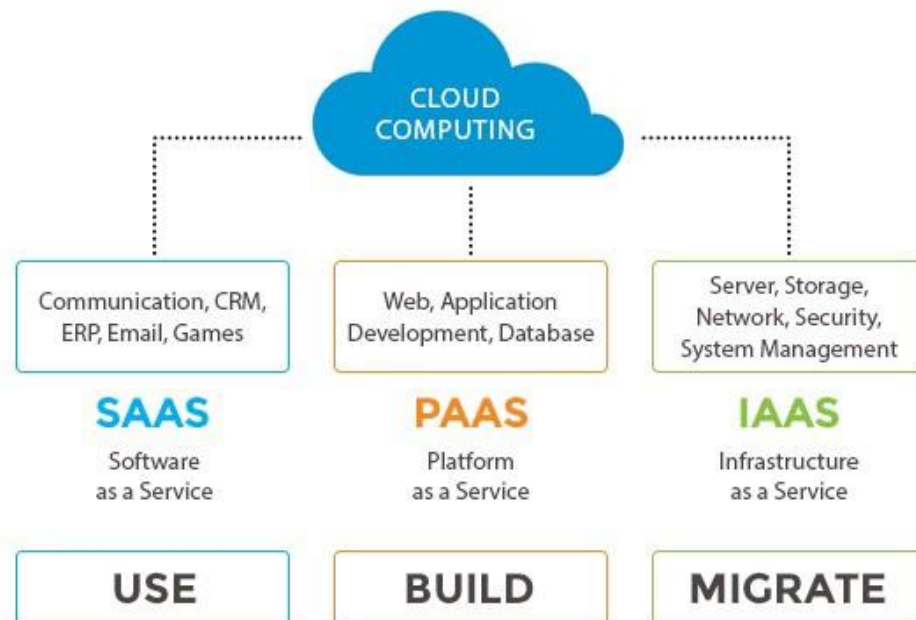


Рисунок 2.5 – Три типи моделей обслуговування

Інфраструктура як послуга (IaaS) надає бізнесу доступ до життєво важливих веб-архітектур, таких як простір для зберігання даних, сервери та підключення, без необхідності купувати та керувати цією Інтернет-інфраструктурою самостійно. Споживач отримує інформаційно-технологічні ресурси — віртуальні сервери з певною обчислювальною потужністю та обсягами пам'яті. Всім «залізом» займається провайдер. Він встановлює на нього ПЗ для створення віртуальних машин, але не займається установкою і підтримкою ПЗ користувача. Провайдер контролює тільки фізичну та віртуальну інфраструктуру.

Платформа як послуга (PaaS) створюються багато разів у хмарах IaaS спеціалістами, щоб зробити масштабованість та розгортання будь-яких додатків тривіальними та допомогти зробити ваші витрати масштабованими та передбачуваними. Головною перевагою такої послуги є те, що за найменші гроші ви можете ініціювати свою програму, не витрачаючи зусиль, крім основної розробки та, можливо, невеликого переносу, якщо ви маєте справу з існуючою програмою. Крім того, PaaS дозволяє значно масштабувати проект, оскільки він базується на хмарних обчисленнях.

В цьому випадку хмарний провайдер надає доступ до операційних систем, засобів розробки і тестування, системам управління базами даних. Провайдер контролює не тільки сервери, системи зберігання даних і обчислювальні потужності, але також пропонує користувачеві на вибір певні платформи і засоби управління ними.

SaaS (Software as a Service) – це бізнес-модель розгортання та реалізації програмного забезпечення, при якому постачальник (провайдер) розробляє додаток, ліцензує його, управляє ним, і надає споживачам (бізнес-клієнтам) доступ до ПЗ через Інтернет. SaaS – це софт + сервіс, програмне забезпечення як послуга і на вимогу.

Постачальник Software as a service надає клієнтові реалізацію бізнес-функцій, функціонала бізнес-додатків, вирішує питання інтеграції свого сервісу в ІТ-систему споживача, бере на себе всі функції з розвитку і підтримки рішень і забезпечення їх масштабування. [11, 1]

2.4.2 Сервіси GCP

Сервіси обчислень та хостингу: [3]

1. Безсерверні обчислення. Cloud Functions, функції Google Cloud як послуги (FaaS), пропонують безсерверне середовище виконання для створення та підключення хмарних служб. За допомогою хмарних функцій ви пишете прості одноцільові функції, які приєднуються до подій, що передаються з вашої хмарної інфраструктури та служб. Ваша функція спрацює, коли запускається подія, що спостерігається. Ваш код виконується в повністю керованому середовищі. Немає необхідності надавати будь-яку інфраструктуру або турбуватися про управління будь-якими серверами.

2. Платформа додатків. App Engine - це платформа Google Cloud як сервісу (PaaS). За допомогою App Engine Google виконує більшу частину управління ресурсами за споживача. Наприклад, якщо програмі потрібно більше обчислювальних ресурсів, оскільки трафік на веб-сайті збільшується, Google автоматично масштабує систему, щоб забезпечити ці ресурси. Якщо

системне програмне забезпечення потребує оновлення безпеки, це буде вирішено.

3. Контейнери. За допомогою контейнерних обчислень споживач може зосередитись на коді програми, а не на розгортанні та інтеграції в середовища хостингу. Google Kubernetes Engine (GKE), контейнери Google Cloud як послуга (CaaS), побудований на відкритій системі Kubernetes, яка надає гнучкість локальних або гібридних хмар, на додаток до загальнодоступної хмарної інфраструктури Google Cloud.

4. Віртуальні машини. Некерованою обчислювальною службою Google Cloud є Compute Engine. Про Compute Engine можна думати як про інфраструктуру як послуга (IaaS), оскільки система забезпечує надійну обчислювальну інфраструктуру, але споживач повинен вибрати та налаштувати компоненти платформи, які він хоче використовувати. Завдяки Compute Engine споживач відповідає за налаштування, адміністрування та моніторинг систем. Google гарантуватиме, що ресурси будуть доступними, надійними та готовими до використання, але їх вирішення і управління ними залежить від споживача. Перевага тут полягає в тому, що споживач має повний контроль над системами та необмежену гнучкість.

Сервіси зберігання: [3]

1. Cloud Storage. Послідовне, масштабоване зберігання даних великої ємності з доступом через посилання.

2. Постійні диски на Compute Engine для використання в якості основного сховища для екземплярів. Compute Engine пропонує як постійні диски на основі жорстких дисків, які називаються стандартними постійними, так і твердотільні постійні диски (SSD).

3. Повністю керовані файлові сервери NFS у Filestore. Екземпляри Filestore можна використовувати для зберігання даних із програм, що працюють на екземплярах Compute Engine VM або кластерах GKE.

Сервіси баз даних: [3]

1. База даних SQL в Cloud SQL, яка забезпечує бази даних MySQL або PostgreSQL.
2. Повністю керована служба реляційних баз даних у Cloud Spanner, яка пропонує узгодженість транзакцій у глобальному масштабі, схеми, запити SQL та автоматичну синхронну реплікацію для високої доступності.
3. Два варіанти зберігання даних NoSQL: Firestore - для документо-подібних даних та Cloud Bigtable - для табличних даних.

Сервіси Big Data: [3]

1. BigQuery сховище, яке використовується для масштабованого аналізу великих наборів даних
2. Dataflow — безсерверний сервіс для аналізу потокових даних з автоматизованим масштабуванням.
3. Datalab — служба, яка містить інструментарій для дослідження, аналізу, візуалізації даних, побудови моделей машинного навчання.
4. Pub/Sub — служба, що здійснює зберігання та обмін повідомленнями між іншими службами.

Сервіси машинного навчання та штучного інтелекту: [3]

1. API Video Intelligence дозволяє використовувати технологію відеоаналізу, яка забезпечує виявлення ярликів, явне виявлення вмісту та регіоналізацію.
2. Speech-to-Text дозволяє конвертувати аудіо в текст, розпізнаючи понад 110 мов та варіантів, щоб підтримати глобальну базу користувачів.
3. Cloud Vision дозволяє легко інтегрувати функції виявлення зору, включаючи маркування зображень, виявлення обличчя та орієнтирів, оптичне розпізнавання символів (OCR) та позначення явного вмісту.
4. Cloud Natural Language API дозволяє додавати аналіз тональності тексту, аналіз сутності, аналіз тональності сутності, класифікацію вмісту та аналіз синтаксису.

5. Cloud Translation дозволяє швидко перекласти вихідний текст на будь-яку з понад сотні підтримуваних мов.
6. Dialogflow дозволяє створювати розмовні інтерфейси для веб-сайтів, мобільних додатків, популярних платформ обміну повідомленнями та пристроїв IoT.

2.5 Обробка набору аналітичних даних

Для виконання аналітичної частини індивідуального завдання, був отриманий доступ до аналітичної інформації від Google Analytics про відвідування веб-сайту .

Для аналізу виключно стрічок, що містять параметри вигляду $\text{sub}\{\text{число}\}=\{\text{значення}\}$, був створений регулярний вираз $(\text{sub}\backslash\text{d})=([\^{\&\backslash\text{n}}]*)?$, який був поміщений у просунуті фільтри Google Analytics (див. рис. 2.6).

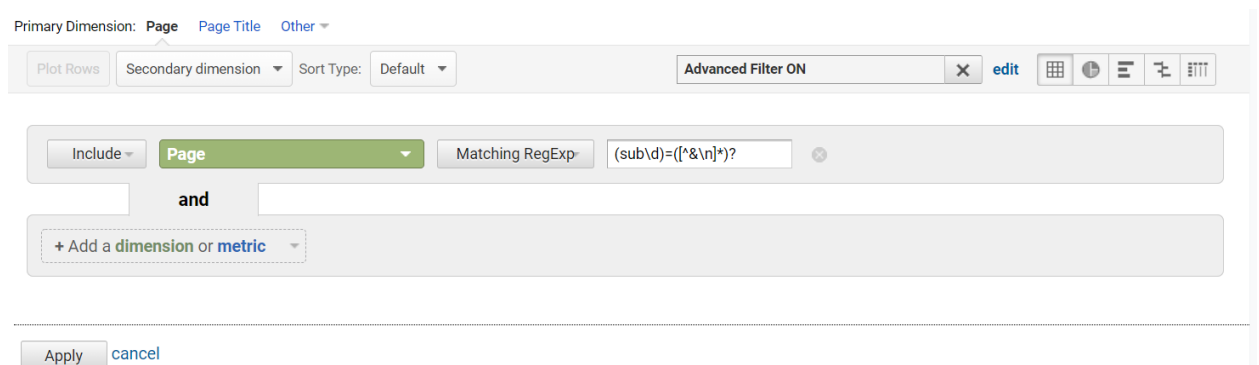


Рисунок 2.6 – Використання фільтрів Google Analytics

Дані були зібрані для періоду 2018-2020 роки. Отримана відфільтрована таблиця була збережена до списку звітів Google Analytics і також експортована у вигляді csv файлу (див. рис. 2.7).

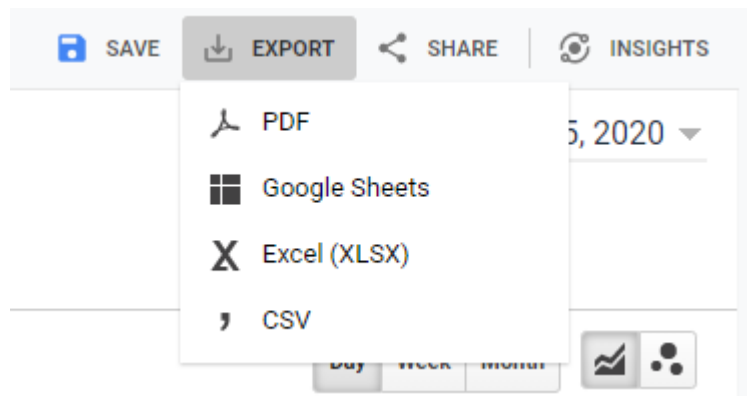


Рисунок 2.7 – Збереження і експорт даних

Для подальшої роботи з даними, було створене сховище даних в Google Storage – бакет під назвою dizmo-report (див. рис. 2.8). [5]

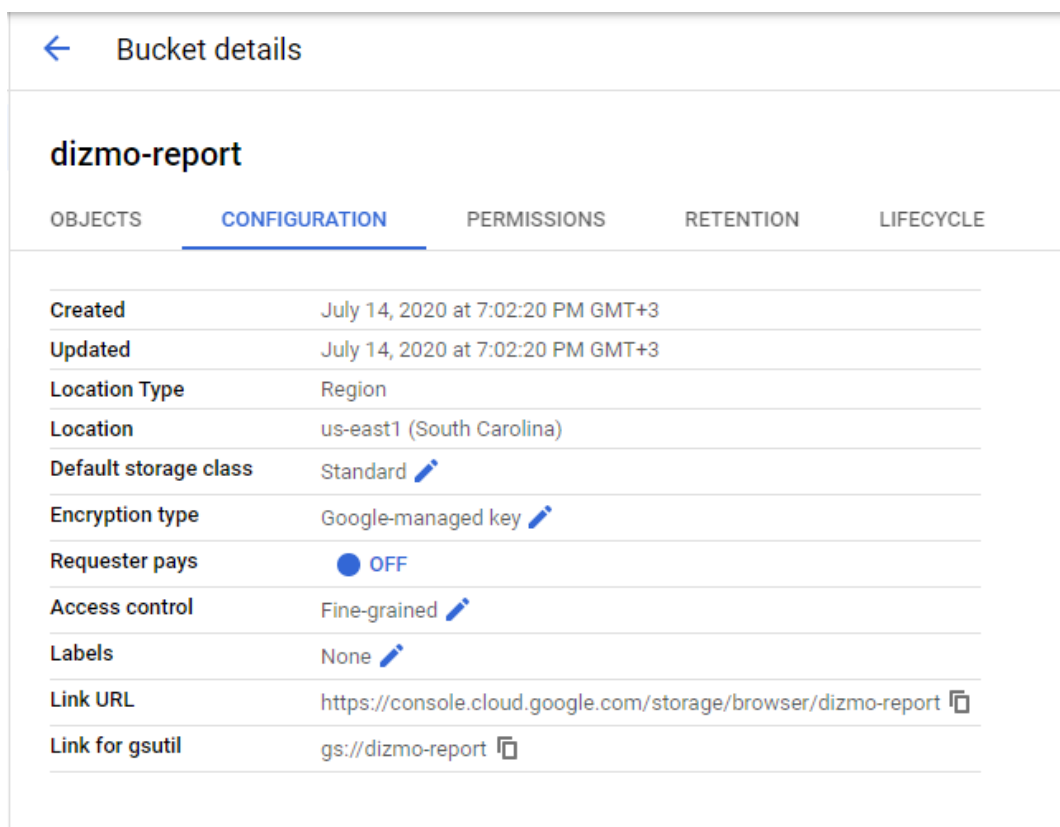



Рисунок 2.8 – Інформація про створений бакет

Всередину бакету був доданий файл з відфільтрованими звітами у форматі csv. Після цього, використовуючи інструмент імпорту BigQuery, звіт був перенесений у таблицю dizmo (див. рис. 2.9)

Schema Details Preview

Field name	Type	Mode	Policy tags 	Description
page	STRING	NULLABLE		
pageview	INTEGER	NULLABLE		
unique_pageview	INTEGER	NULLABLE		
avg_time	STRING	NULLABLE		
entrances	INTEGER	NULLABLE		
bounce_rate	STRING	NULLABLE		
exit_percentage	STRING	NULLABLE		
page_value	STRING	NULLABLE		

[Edit schema](#)

Рисунок 2.9 – Таблиця звіту в BigQuery

Для витягнення ключів та їх параметрів, а також кількості комбінацій ключ=параметр і суми кількості відвідувань був створений SQL-скрипт (див. лістинг 2.2).

Лістинг 2.2 – SQL-скрипт обробки звіту

```
WITH data AS (SELECT REGEXP_EXTRACT(vals, r"sub\d") as
parameter, REGEXP_EXTRACT(vals, r"=([^\&\n]*)") AS value, pageview
FROM(
  SELECT vals, pageview FROM (
    SELECT REGEXP_EXTRACT_ALL(page, r"sub\d=([^\&\n]*)") AS reg_extr,
pageview FROM dizmot.dizmo
  ), UNNEST(reg_extr) as vals
))

SELECT parameter, value, COUNT(value) AS amount, SUM(pageview)
AS pageviews FROM data GROUP BY parameter, value ORDER BY amount DESC
```

Результати виконання скрипта зображені на рисунку 2.10.

Row	parameter	value	amount	pageviews
1	sub3	40	16970	63394
2	sub1	29	10646	41709
3	sub4	798	10166	40007
4	sub1	1	7033	25973
5	sub4	797	3461	11546
6	sub3	3	2448	17853
7	sub4	796	2392	8055
8	sub1	5	1751	14116
9	sub4	1049	443	1696
10	sub5	346479	237	920
11	sub5	39928	213	780
12	sub5	313703	208	745
13	sub5	460992	205	684
14	sub5	552140	199	664
15	sub5	542847	198	668

16	sub5	43555	197	668
17	sub5	29	193	678
18	sub5	529000	192	678
19	sub5	30729	191	670
20	sub5	96203	188	628
21	sub5	558491	188	640
22	sub5	565641	185	600
23	sub5	328455	178	702
24	sub5	447716	176	620
25	sub5	539517	176	646
26	sub5	542608	175	573
27	sub5	338063	173	572
28	sub5	514549	170	588
29	sub5	553275	169	616
30	sub5	272525	166	552
31	sub5	522975	165	556
32	sub5	348377	165	569
33	sub5	210931	164	582

Рисунок 2.10 – Результативна оброблена таблиця

Також в Google Datastudio була створена візуалізація для кожної групи параметрів з частотами появи значення та кількістю відвідувань.

[7]

ВИСНОВОК

Виробнича практика є головною частиною навчального процесу. Під час практики я удосконалив професійні уміння та навички по своїй спеціальності, закріпив і розширив знання на базі вивченого.

В процесі проходження практики був проведений аналіз платформи хмарної авто інтеграції Onlizer. Було розглянуте поняття алгоритму, його властивості і ефективність. На основі методу Карацуби був розглянутий приклад покращення ефективності операції множення і була проведена програмна реалізація на Python. Окрім цього був пройдений курс Google Cloud Certified Professional Cloud Architect, який розглядав у собі хмарне обчислення, можливості платформи Google Cloud – її сервіси з виконанням практичних завдань для закріплення навичок. З використанням BigQuery, сервісу GCP, були проаналізовані дані відвідуваності посилань. На основі них була складена таблиця використання певних параметрів з частотою появи їх у комбінації з значенням і з сумарною інформацією про відвідуваність сторінок для цієї групи параметрів. Для результативної таблиці був складений візуальний дашборд, який відображав рейтинг конкретних значень.

Як результат проходження практики був складений звіт, що складається з двох частин – загальної інформації про базу проходження практики та інформації про виконане індивідуальне завдання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 3 Types of Cloud Computing Services [Електронний ресурс] – Режим доступу до ресурсу: <https://www.monitis.com/blog/3-types-of-cloud-computing-services/> - Дата доступу: 14.05.2020.

2. 4 Best Cloud Deployment Models [Електронний ресурс] – Режим доступу до ресурсу: <https://www.sam-solutions.com/blog/four-best-cloud-deployment-models-you-need-to-know/> - Дата доступу: 14.05.2020.

3. About Google Cloud services [Електронний ресурс] – Режим доступу до ресурсу: <https://cloud.google.com/docs/overview/cloud-platform-services> - Дата доступу: 14.05.2020.

4. Characteristics of Cloud Computing as per NIST [Електронний ресурс] – Режим доступу до ресурсу: <https://timesofcloud.com/cloud-tutorial/characteristics-of-cloud-computing-as-per-nist/> - Дата доступу: 14.05.2020.

5. Google Cloud Console [Електронний ресурс] – Режим доступу до ресурсу: <http://console.cloud.google.com/home/> - Дата доступу: 14.05.2020.

6. Google Cloud Platform [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Google_Cloud_Platform - Дата доступу: 14.05.2020.

7. Візуалізований звіт Google DataStudio[Електронний ресурс] – Режим доступу до ресурсу: <https://datastudio.google.com/s/iN9ZChQByVc> - Дата доступу: 14.05.2020.

8. Ефективність алгоритму [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Ефективність_алгоритму - Дата доступу: 14.05.2020.

9. Офіційний сайт ScalHive [Електронний ресурс] – Режим доступу до ресурсу: <https://scalhive.com/> - Дата доступу: 14.05.2020.

10. Характеристики хмарних обчислень [Електронний ресурс] – Режим доступу до ресурсу:

<https://coggle.it/diagram/WNtfITPg0gABaMkZ/t/характеристики-хмарних-обчислень> - Дата доступу: 14.05.2020.

11. Хмарна піраміда: IaaS, PaaS і SaaS [Електронний ресурс] – Режим доступу до ресурсу: <https://gigacloud.ua/blog/navchannja/hmarna-piramida-iaas-paas-i-saas> - Дата доступу: 14.05.2020.

12. Хмарні обчислення [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Хмарні_обчислення - Дата доступу: 14.05.2020.