

Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота бакалавра

на тему: «Розробка програмно-аналітичного комплексу аналізу ринку малого та середнього бізнесу на основі даних соціальної мережі Instagram»

Виконав: студент групи ІПЗ20-2

Спеціальність: 121 «Інженерія програмного забезпечення»

_____Бабенко А. С. _____
(прізвище та ініціали)

Керівник к.ф.-м.н. доцент Рудянова Т. М.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент Університет митної справи та фінансів
(місце роботи)

в.о. зав. каф. Кібербезпеки та інформаційних технологій
(посада)

к.т.н., доцент Прокопович-Ткаченко Д. І.
(науковий ступінь, вчене звання, прізвище та ініціали)

АНОТАЦІЯ

Бабенко А. С. Розробка програмно-аналітичного комплексу аналізу ринку малого та середнього бізнесу на основі даних соціальної мережі Instagram.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення». – Університет митної справи та фінансів, Дніпро, 2024.

У даній роботі розглянуто розробку програмно-аналітичного комплексу для аналізу ринку малого та середнього бізнесу на основі даних з Instagram, однієї з найпопулярніших соціальних медіа-платформ. Малий та середній бізнес, відіграючи ключову роль в економіці України, забезпечує значну частку робочих місць і доданої вартості, тому застосування сучасних технологій для аналізу даних із соціальних мереж має потенціал підвищити ефективність їх діяльності та конкурентоспроможність. У дослідженні проведено огляд методів аналізу даних, таких як моніторинг мереж, аналіз тональності, кластеризація та візуалізація, а також різних комерційних сервісів, включаючи Brandwatch, Hootsuite, Sprout Social, що пропонують інструменти для аналізу соціальних медіа. Програмно-аналітичний комплекс включає серверний застосунок на Python та кросплатформений застосунок на Rust з фреймворком Tauri, що забезпечує продуктивність та гнучкість. Особлива увага приділяється методам машинного навчання та обробки природної мови, що дозволяє отримувати глибокі інсайти та прогнозувати ринкові тенденції. Результати дослідження свідчать, що використання таких інструментів дозволяє малому та середньому бізнесу швидко реагувати на ринкові зміни, оптимізувати маркетингові стратегії, покращувати взаємодію з клієнтами та підвищувати конкурентоспроможність, сприяючи зростанню та розвитку підприємств. Робота демонструє ефективність використання даних з соціальних мереж для аналітики та прийняття рішень у сфері малого та середнього бізнесу.

Ключові слова: РОЗРОБКА, ВІЗУАЛІЗАЦІЯ ДАНИХ, ЗБІР ДАНИХ, TAURI, PYTHON, RUST, JS, INSTAGRAM.

ANNOTATION

Babenko A. S. Development of a software and analytical complex for analyzing the market of small and medium-sized businesses based on Instagram data.

Qualification work for a bachelor's degree in specialty 121 "Software Engineering." - University of Customs and Finance, Dnipro, 2024.

This paper discusses the development of a software and analytical complex for analyzing the SME market based on data from Instagram, one of the most popular social media platforms. Small and medium-sized businesses, playing a key role in the Ukrainian economy, provide a significant share of jobs and added value, so the use of modern technologies to analyze data from social networks has the potential to increase their efficiency and competitiveness. The study reviews data analysis methods such as network monitoring, tone analysis, clustering, and visualization, as well as various commercial services, including Brandwatch, Hootsuite, and Sprout Social, that offer tools for social media analysis. The software and analytical complex includes a server-side Python application and a cross-platform Rust application with the Tauri framework, which ensures performance and flexibility. Particular attention is paid to machine learning and natural language processing methods, which allows to obtain deep insights and predict market trends. The study results show that the use of such tools allows small and medium-sized businesses to respond quickly to market changes, optimize marketing strategies, improve customer experience, and increase competitiveness, contributing to the growth and development of enterprises. The paper demonstrates the effectiveness of using data from social networks for analytics and decision-making in the field of small and medium-sized businesses.

Keywords: DEVELOPMENT, DATA VISUALIZATION, DATA COLLECTION, TAURI, PYTHON, RUST, JS, INSTAGRAM.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ	8
1.1 Огляд існуючих методів та підходів для аналізу даних соціальних мереж.....	8
1.2 Аналіз функціональності та переваг існуючих інструментів аналізу соціальних мереж	10
1.3 Постановка завдання	14
1.4 Висновки до першого розділу	16
РОЗДІЛ 2. АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО КОМПЛЕКСУ.....	19
2.1 Огляд існуючих методів аналізу соціальних мереж	19
2.2 Вибір технологій та засобів розробки кросплатформеного застосунку.	21
2.3 Аналіз середовищ розробки	25
2.4 Аналіз мов програмування, розмітки та стилів	27
2.5 Висновки до другого розділу	32
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАДАЧІ.....	34
3.1 Розробка парсера з орієнтацією на роботу на серверах під управлінням операційної системи Linux з використанням мови програмування Python та бібліотеки Instaloader	34
3.2 Розробка візуалізатора з використанням Tauri та HTML, CSS, JavaScript та Chart.js.....	36
3.3 Створення логотипу для програмного комплексу PostPrism	40
3.4 Висновки до третього розділу	49
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДОДАТОК А	59
ДОДАТОК Б.....	62
ДОДАТОК В	66
ДОДАТОК Г.....	70

ВСТУП

Актуальність дослідження. Важливість малого та середнього бізнесу неможливо недооцінити: він створює робочі місця, надає послуги створює товари. На період з 2020 по 2022 роки малий та середній бізнес займав провідне значення в економіці України [1], понад 60% доданої вартості та більше 80% робочих місць були створені саме завдяки малому та середньому бізнесу. Цей сектор виявився надзвичайно гнучким і адаптивним до змінних умов, демонструючи високу стійкість у кризові періоди. Малий та середній бізнес відіграє ключову роль у стимулюванні інновацій, підвищенні конкурентоспроможності та розвитку регіональних економік. Багато підприємців активно впроваджують нові технології та бізнес-моделі, що сприяє загальному прогресу країни.

Зважаючи на важливість малого та середнього бізнесу, виникає необхідність у створенні інструментів, що дозволять ефективно аналізувати та використовувати дані для підвищення його конкурентоспроможності. Один з перспективних напрямків є розробка аналітичного застосунку, який базуватиметься на даних з Instagram. Ця соціальна медіа-платформа забезпечує величезний обсяг інформації [2] про споживацькі вподобання, тренди та конкурентну активність. Аналіз таких даних дозволить отримати глибоке розуміння ринкових тенденцій, поведінки клієнтів та ефективності різних маркетингових стратегій. Використання цих інсайтів дасть змогу виконавцям приймати обґрунтовані стратегічні рішення, які сприятимуть зростанню та розвитку бізнесу. Застосунок допоможе покращити ключові маркетингові метрики, такі як конверсія, залучення аудиторії та лояльність клієнтів. Окрім того, інтеграція отриманих даних у бізнес-процеси підвищить їхню ефективність, що дозволить оптимізувати витрати, підвищити продуктивність і загалом збільшити прибутковість підприємства.

Для розробки аналітичного застосунку на основі даних з Instagram необхідно використовувати комплексний підхід, що поєднує різні методи та інструменти. Першим кроком буде проведення ретельного аналізу вимог та визначення

ключових функцій, які мають бути реалізовані в застосунку. Це дозволить забезпечити, що кінцевий продукт повністю відповідатиме потребам бізнесу.

Наступним етапом стане збір та обробка відповідних даних з Instagram. Для цього буде використано API Instagram [3], що дозволить отримувати структуровані дані про публікації, профілі користувачів та інші важливі показники. Зібрані дані будуть ретельно очищені та перетворені у формат, придатний для аналізу.

Методологія дослідження: метод теорії інформації, методи веб-скрейпінгу, обробка та аналіз інформації, методи візуалізації даних, методології гнучкої розробки, методи проектування програмного забезпечення, методи верифікації та тестування програмного забезпечення.

Об'єктом дослідження є процес розробки програмно-аналітичного комплексу аналізу ринку малого та середнього бізнесу на основі даних соціальної мережі Instagram.

Предметом дослідження є створення програмного комплексу: серверного застосунку на Python [4] та кросплатформеного застосунку на rsut [5] з використанням фреймворку tauri [6] та html [7], css [8], java script [9]. Дослідження фокусується на розробці та інтеграції цих компонентів для створення повноцінного програмного рішення, яке забезпечує ефективну взаємодію користувача з великим масивом даних.

В результаті виконання завдання можна підтвердити наступні програмні результати:

ПРО1 – аналіз, цілеспрямований пошук і вибір необхідних для вирішення професійних завдань інформаційно-довідникових ресурсів і знань з урахуванням сучасних досягнень науки і техніки;

ПРО2 – знання кодексу професійної етики, розуміння соціальної значимості та культурних аспектів інженерії програмного забезпечення і дотримання їх в професійній діяльності;

ПРО6 – уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;

ПРО8 – вміння розробляти людино-машинний інтерфейс;

ПР09 – знання та вміння використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення;

ПР10 – проведення передпроектного обстеження предметної області, системного аналізу об'єкта проектування;

ПР12 – застосування на практиці ефективних підходів щодо проектування програмного забезпечення;

ПР15 – мотивований вибір мов програмування та технологій розробки для розв'язання завдань створення і супроводження програмного забезпечення.

Досягнення поставлених завдань дозволить створити зручний та функціональний програмно-аналітичний комплекс, призначений для аналізу ринку малого та середнього бізнесу на основі інформації із соціальної мережі Instagram, який складатиметься з консольного парсера та крос-платформеного застосунку. Використання сучасних технологій разом із передовими методиками розробки сприятиме високій якості кінцевого продукту, що характеризуватиметься простим у використанні програмним комплексом зі зручним та інтуїтивним інтерфейсом. В цілому, цей комплекс стане надійним інструментом для аналізу ринку, допомагаючи користувачам ефективно досягати своїх бізнес-цілей завдяки вичерпному і точному аналізу отриманих даних.

Структура роботи: 3 розділи, висновків, перелік використаних джерел, 4 додатки. Обсяг роботи – 55 сторінок, робота містить 11 рисунків, перелік використаних джерел складається з 31 посилання, додатки розміщено на 13 сторінках.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

1.1 Огляд існуючих методів та підходів для аналізу даних соціальних мереж

Аналіз відкритих даних з соціальних мереж (Social Media Data Analysis, SMDA) є активною галуззю досліджень у сфері обробки природної мови, машинного навчання та аналізу даних. Ця область передбачає збирання, очищення, інтеграцію та аналітичну обробку величезних обсягів неструктурованих даних, які генеруються користувачами соціальних медіа, таких як Facebook, X(Twitter), Instagram та інших платформ. Метою SMDA є виявлення корисних закономірностей, трендів, настроїв та поведінкових моделей, що можуть бути використані в різних галузях, як-от маркетинг, політика, охорона здоров'я, безпека тощо.

Існує кілька основних методів та підходів до аналізу даних соціальних мереж.

Аналіз тональності та настрою (Sentiment Analysis) [10]. Це процес визначення емоційного забарвлення тексту, наприклад, чи є певний текст позитивним, негативним або нейтральним. Для цього використовуються методи машинного навчання, лексичні ресурси та правила.

Аналіз тем та кластеризація (Topic Modeling & Clustering). Ці методи дозволяють виявляти основні теми та групувати повідомлення за подібністю змісту. Популярними алгоритмами є Латентний розподіл Діріхле (LDA), k-means кластеризація та аналіз семантичної близькості.

Аналіз мереж та графів (Network & Graph Analysis). Соціальні мережі можна представити у вигляді графів, де вузли - це користувачі, а ребра - зв'язки між ними. Аналіз графів дозволяє виявляти впливових акторів, визначати центри спільнот та вивчати розповсюдження інформації.

Виявлення подій та трендів (Event & Trend Detection). Ці методи спрямовані на ідентифікацію "гарячих" тем, новин та трендів, що з'являються в соціальних

мережах у режимі реального часу. Вони часто базуються на аналізі часових рядів, виявленні різких змін активності та кластеризації повідомлень.

Аналіз зв'язаних даних (Linked Data Analysis). Оскільки дані соціальних мереж містять посилання на різні сутності (люди, організації, місця тощо), важливим є зв'язування цих сутностей з відкритими базами знань, такими як DBpedia чи Wikidata. Це дозволяє збагатити дані семантичною інформацією та проводити складніші аналітичні запити.

Аналіз мультимодальних даних (Multimodal Data Analysis). Сучасні соціальні мережі містять не лише текст, але й зображення, відео та інші типи даних. Методи обробки мультимодальних даних комбінують техніки комп'ютерного зору, розпізнавання мовлення та обробки природної мови для більш повного розуміння контексту.

Візуалізація даних (Data Visualization). Ефективна візуалізація даних дозволяє краще зрозуміти результати аналізу та виявити закономірності, які важко помітити в "сирих" даних. Для цього використовуються методи інформаційної візуалізації, геовізуалізації та візуалізації мереж.

Існує кілька популярних сервісів та платформ, які спеціалізуються на аналізі даних із соціальних мереж:

Brandwatch – одна з провідних платформ для моніторингу та аналізу соціальних мереж. Вона пропонує інструменти для відстеження згадувань бренду, аналізу тональності, виявлення впливових осіб та трендів у соціальних мережах.

Hootsuite - всеохоплююча платформа для управління соціальними мережами, яка також має потужні функції аналітики. Вона дозволяє відстежувати ключові показники ефективності, проводити аналіз настроїв та залучати аудиторію.

Sprout Social – платформа, орієнтована на малий та середній бізнес, надає інструменти для моніторингу згадувань бренду, аналізу залученості аудиторії та створення звітів про ефективність у соціальних мережах.

Synthesio – спеціалізований сервіс для аналізу соціальних даних, який пропонує глибоку аналітику щодо думок клієнтів, настроїв ринку та конкурентного аналізу.

Crimson Hexagon – платформа для аналізу соціальних мереж, яка забезпечує всебічний аналіз тексту, візуалізацію даних та виявлення впливових авторів.

NetBase – сервіс для моніторингу соціальних мереж, який спеціалізується на аналізі тональності, виявленні трендів та проведенні досліджень ринку.

Clarabridge – потужна платформа для аналізу голосу клієнта, яка обробляє дані з соціальних мереж, електронної пошти, опитувань тощо для вдосконалення клієнтського досвіду.

SocialBakers – комплексний набір інструментів для аналізу даних соціальних мереж, включаючи відстеження показників залученості, порівняння з конкурентами та оптимізацію контенту.

Крім комерційних сервісів, існують також відкриті бібліотеки та фреймворки для аналізу даних соціальних мереж, такі як AYLIEN Text Analysis, Apache TinkerPop, NetworkX та інші. Вони можуть бути корисними для дослідників, розробників та компаній, які хочуть створювати власні рішення для аналізу соціальних даних.

Важливо зазначити, що більшість цих сервісів пропонують багато можливостей, від простого моніторингу згадувань до складних аналітичних рішень, таких як машинне навчання та візуалізація даних. Вибір конкретного сервісу залежить від потреб організації, бюджету та необхідного рівня деталізації аналітики.

1.2 Аналіз функціональності та переваг існуючих інструментів аналізу соціальних мереж

Світ соціальних мереж рясніє корисною інформацією, яка може бути цінною для бізнесу та дослідників. Проте, видобуток та аналіз цих даних може бути складним завданням. Ось де на допомогу приходять інструменти аналізу даних соціальних мереж.

Різні інструменти пропонують широкий спектр функцій та переваг, тому важливо вибрати той, який найкраще відповідає вашим потребам. Ось короткий огляд деяких популярних інструментів:

1. Brandwatch:

- всеохоплююча платформа для моніторингу всіх основних соціальних мереж, новинних сайтів, блогів та форумів;
- потужні інструменти аналізу тональності, виявлення впливових осіб та відстеження трендів;
- гнучкі можливості сегментації даних та створення персоналізованих панелей;
- інтеграція з crm-системами та іншими бізнес-додатками;
- перевага: комплексне рішення для аналізу даних соціальних мереж.

2. Hootsuite:

- зручний інтерфейс для публікації контенту та управління кількома акаунтами соціальних мереж;
- аналітичні інструменти для відстеження показників ефективності, настроїв та залученості аудиторії;
- можливість планування публікацій та автоматизації процесів;
- інтеграція з рекламними платформами соціальних мереж;
- перевага: всеохоплююча платформа для управління присутністю в соціальних мережах.

3. Sprout Social:

- зручні інструменти для командної роботи та співпраці в соціальних мережах;
- аналітика залученості аудиторії, настроїв та тенденцій;
- можливість створювати звіти та візуалізації для різних цілей;
- інтеграція з crm-системами та інструментами для роботи з клієнтами;
- перевага: орієнтована на малий та середній бізнес, зручна для командної роботи.

4. Synthesio:

- глибокий аналіз тональності та настроїв аудиторії;
- можливість виявлення ключових тем, що цікавлять аудиторію;
- інструменти для відстеження конкурентів та порівняння з ними;
- потужна візуалізація даних для кращого розуміння тенденцій;
- перевага: спеціалізований на аналізі думок споживачів та конкурентному аналізі.

5. Crimson Hexagon:

- передові технології машинного навчання для аналізу тексту та візуалізації даних;
- можливість виявлення впливових авторів та ключових тем;
- аналіз зображень та відео в соціальних мережах;
- інтеграція з системами бізнес-аналітики для поглибленого аналізу;
- перевага: сучасні технології аналізу, підходить для складних аналітичних завдань.

6. NetBase:

- швидке виявлення трендів та актуальних тем в соціальних мережах;
- інструменти для відстеження репутації бренду та управління кризовими ситуаціями;
- можливість сегментації даних за демографічними та географічними ознаками;
- інтеграція з системами управління досвідом клієнтів;
- перевага: орієнтована на швидке реагування та управління репутацією.

7. Clarabridge:

- комплексна платформа для аналізу голосу клієнта з різних джерел;
- глибокий аналіз тональності та виявлення ключових факторів задоволеності клієнтів;
- можливість створення персоналізованих подорожей клієнтів;
- інтеграція з системами CRM та обслуговування клієнтів;
- перевага: всеохоплююче рішення для вдосконалення клієнтського досвіду.

8. SocialBakers:

- зручні інструменти для порівняння показників ефективності з конкурентами;
- аналітика залученості аудиторії та оптимізація контенту;
- можливість відстежувати ефективність рекламних кампаній у соціальних мережах;
- інтеграція з рекламними платформами соціальних мереж;
- перевага: зосереджена на бенчмаркінгу та оптимізації контенту.

Ці інструменти пропонують широкий спектр функцій, від базового моніторингу до глибокої аналітики та візуалізації даних. Вибір конкретного інструменту залежить від цілей аналізу, бюджету та специфічних потреб організації. Деякі інструменти можуть бути більш доцільними для великих підприємств, тоді як інші краще підходять для малого та середнього бізнесу. Важливо також враховувати інтеграцію з існуючими системами та можливості автоматизації процесів, щоб забезпечити максимально ефективне використання обраного рішення, у таблиці 1.1 наведено порівняння функціоналу та особливостей.

Кожен з перелічених інструментів має свої сильні та слабкі сторони, що визначають його придатність для конкретних завдань. Brandwatch виділяється здатністю охоплювати широкий спектр соціальних мереж та онлайн-ресурсів, потужними можливостями аналізу тональності та виявлення впливових осіб, а також інтеграцією з CRM-системами. Hootsuite пропонує зручний інтерфейс для управління кількома акаунтами, планування публікацій та аналітичні інструменти для оцінки ефективності контенту. Sprout Social відзначається зручними інструментами для командної роботи та аналітикою залученості аудиторії, а Synthesio спеціалізується на глибокому аналізі тональності та настроїв аудиторії. Crimson Hexagon використовує машинне навчання для глибокого аналізу тексту та візуалізації даних, NetBase орієнтована на швидке виявлення трендів та управління репутацією, Clarabridge аналізує голос клієнта для покращення взаємодії, а SocialBakers зосереджується на бенчмаркінгу та оптимізації контенту.

Таблиця 1.1

Функціональність та переваги популярних інструментів для аналізу даних
соціальних мереж

Інструмент	Моніторинг мереж	Аналіз тональності	Виявлення трендів	Аналіз конкурентів	Візуалізація даних	Основні переваги
Brandwatch	Так	Так	Так	Так	Так	Комплексне рішення, інтеграція з CRM
Hootsuite	Так	Так	Так	Частково	Так	Управління акаунтами, планування публікацій
Sprout Social	Так	Так	Так	Частково	Так	Зручна для командної роботи, орієнтована на МСБ
Synthesio	Так	Так	Так	Так	Так	Глибокий аналіз думок споживачів

1.3 Постановка завдання

Ця робота спрямована на дослідження існуючих методологій, підходів та програмного забезпечення для аналізу соціальної мережі Інстаграм. На основі отриманих результатів планується розробити рекомендації, технічне завдання та створити новий комплексний програмний продукт для аналізу та візуалізації даних, який включатиме позитивні аспекти існуючих продуктів та усуватиме недоліки.

Основними етапами аналізу інформації є:

1. Парсинг інформації – процес витягування необхідних даних з різноманітних джерел, таких як файли, веб-сторінки, бази даних чи потоки даних. Мета парсингу – перетворити необроблені дані у структуровану форму, яку можна легко аналізувати. Це може включати видобуток даних з різних форматів, таких як CSV, JSON, XML, тексту тощо.

2. Очищення. Після парсингу дані часто містять помилки, пропущені значення або несумісні формати. Процес очищення даних полягає у виявленні та виправленні таких проблем, щоб дані були послідовними та коректними. Це може включати видалення дублікатів, обробку пропущених значень, перетворення форматів даних та виявлення аномалій.

3. Візуалізація. Після того, як дані були успішно очищені та структуровані, їх можна візуалізувати для кращого розуміння та аналізу. Візуалізація даних передбачає представлення даних у вигляді графіків, діаграм, карт тепла чи інших візуальних форматів. Ця стадія допомагає виявити тенденції, закономірності та відхилення в даних, що може підказати напрямки для подальшого аналізу.

Виходячи з цього, мова програмування Python з бібліотекою з відкритим вхідним кодом Instaloader є ідеальним рішенням для написання парсеру, розрахованого на роботу на сервері під управлінням операційної системи Linux. Для візуалізації, в свою чергу, буде використано фреймворк Taui, та мова програмування Rust, для написання бекенду. Для створення інтерфесу користувача буде використано html, css, java script та фреймворк chart.js.

Дослідження методів і інструментів для аналізу даних соціальних мереж показало, що ця сфера є важливою і швидко розвивається, з широким застосуванням у таких галузях, як маркетинг, політика, охорона здоров'я та безпека. Існуючі інструменти пропонують різноманітні функції: від базового моніторингу згадувань до глибокого аналізу тональності, виявлення трендів, візуалізації даних і інтеграції з іншими системами. Вибір конкретного інструменту залежить від потреб організації, бюджету та необхідного рівня деталізації аналітики. Однак більшість з цих інструментів є комерційними, що може стати проблемою для малих компаній або дослідницьких проектів з обмеженим фінансуванням. Відкриті бібліотеки можуть бути альтернативою комерційним рішенням, але їх використання вимагає додаткових навичок програмування.

З огляду на це, подальша робота має включати розробку комплексного програмного продукту для аналізу та візуалізації даних соціальної мережі Instagram, який поєднає переваги існуючих інструментів та усуне їх недоліки. Для

забезпечення гнучкості, масштабованості та економічної ефективності доцільно використовувати відкриті бібліотеки і фреймворки, такі як Instaloader, Tauri, Rust, HTML, CSS, JavaScript і chart.js. Важливо також впровадити сучасні методи аналізу даних, зокрема аналіз тональності, виявлення трендів, кластеризацію, аналіз мереж і мультимодальних даних, а також створити зручний інтерфейс для полегшення роботи з даними та візуалізації результатів. Крім цього, слід забезпечити можливість інтеграції з іншими системами та бізнес-додатками для підвищення ефективності аналізу, а також провести тестування та оптимізацію рішення для досягнення високої продуктивності та надійності при обробці великих обсягів даних. Реалізація цих кроків дозволить створити ефективний і доступний інструмент для аналізу даних Instagram, корисний як для комерційних організацій, так і для дослідницьких проектів.

1.4 Висновки до першого розділу

Перший розділ, присвячений дослідженню предметної області та постановці завдання, дозволяє детально розглянути сучасні методи та підходи до аналізу даних соціальних мереж, зокрема таких платформ, як Facebook, X(Twitter) та Instagram, акцентуючи увагу на різноманітних техніках обробки природної мови, машинного навчання та аналізу даних. SMDA включає в себе процеси збирання, очищення, інтеграції та аналітичної обробки великих обсягів неструктурованих даних, які генеруються користувачами соціальних мереж, з метою виявлення корисних закономірностей, трендів, настроїв та поведінкових моделей, що мають застосування в різних галузях, таких як маркетинг, політика, охорона здоров'я та безпека.

Аналіз тональності та настрою дозволяє визначити емоційне забарвлення тексту, використовуючи методи машинного навчання, лексичні ресурси та правила, в той час як аналіз тем та кластеризація допомагають виявляти основні теми та групувати повідомлення за подібністю змісту за допомогою алгоритмів, таких як Латентний розподіл Діріхле (LDA) та k-means кластеризація. Аналіз мереж та

графів представляє соціальні мережі у вигляді графів, де вузли – це користувачі, а ребра – зв'язки між ними, що дозволяє виявляти впливових акторів, визначати центри спільнот та вивчати розповсюдження інформації. Виявлення подій та трендів спрямоване на ідентифікацію "гарячих" тем та новин у реальному часі, використовуючи аналіз часових рядів та кластеризацію повідомлень, а аналіз зв'язаних даних збагачує дані семантичною інформацією, дозволяючи проводити складніші аналітичні запити. Мультиmodalний аналіз даних включає в себе обробку тексту, зображень та відео, комбінуючи техніки комп'ютерного зору та обробки природної мови для більш повного розуміння контексту, а візуалізація даних допомагає краще зрозуміти результати аналізу та виявити закономірності.

Існує безліч популярних сервісів та платформ, які спеціалізуються на аналізі даних соціальних мереж, серед яких Brandwatch, Hootsuite, Sprout Social, Synthesio, Crimson Hexagon, NetBase, Clarabridge та SocialBakers. Кожен з цих інструментів має свої особливості та переваги: від моніторингу згадувань та аналізу тональності до виявлення трендів, аналізу конкурентів та глибокої візуалізації даних. Вибір конкретного інструменту залежить від потреб організації, бюджету та необхідного рівня деталізації аналітики. Комерційні сервіси надають широкий спектр можливостей, але можуть бути дорогими для малих компаній чи дослідницьких проєктів з обмеженим фінансуванням, тоді як відкриті бібліотеки, такі як AYLIE Text Analysis, Apache TinkerPop та NetworkX, можуть стати альтернативою, але вимагають додаткових навичок програмування.

Постановка завдання у цьому контексті полягає у дослідженні існуючих методологій, підходів та програмного забезпечення для аналізу соціальної мережі Instagram з метою розробки рекомендацій, технічного завдання та створення нового комплексного програмного продукту для аналізу та візуалізації даних. Основними етапами аналізу є парсинг інформації, очищення даних та їх візуалізація. Парсинг передбачає витягування необхідних даних з різноманітних джерел та їх перетворення у структуровану форму. Очищення даних полягає у виявленні та виправленні помилок, пропущених значень та несумісних форматів для забезпечення послідовності та коректності даних. Візуалізація допомагає

представити дані у вигляді графіків, діаграм та інших візуальних форматів для виявлення тенденцій та закономірностей.

Розробка нового програмного продукту базується на використанні мови програмування Python з бібліотекою Instaloader для написання парсеру, розрахованого на роботу на сервері під управлінням операційної системи Linux. Для візуалізації даних буде використано фреймворк Taui та мову програмування Rust, а для створення інтерфейсу користувача – HTML, CSS, JavaScript та фреймворк chart.js. Це забезпечить гнучкість, масштабованість та економічну ефективність нового продукту, поєднуючи переваги існуючих інструментів та усуваючи їх недоліки. Реалізація таких кроків дозволить створити ефективний і доступний інструмент для аналізу даних Instagram, який буде корисним як для комерційних організацій, так і для дослідницьких проектів, забезпечуючи інтеграцію з іншими системами та бізнес-додатками для підвищення ефективності аналізу.

РОЗДІЛ 2. АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО КОМПЛЕКСУ

2.1 Огляд існуючих методів аналізу соціальних мереж

Аналіз даних соціальних мереж є однією з найактуальніших і найдинамічніших сфер сучасних досліджень, що охоплює широкий спектр галузей, таких як маркетинг, політика, охорона здоров'я та безпека. Цей вид аналізу дозволяє не лише відстежувати згадки про бренд або продукт, але й проводити глибокі аналітичні дослідження, що включають аналіз тональності, виявлення трендів, кластеризацію користувачів і аналіз соціальних мереж як складних систем. Завдяки цьому організації можуть отримати глибше розуміння потреб і поведінки своїх клієнтів, виявляти ключові тенденції та краще реагувати на зміни в соціальних настроях.

Існуючі інструменти для аналізу соціальних мереж надають різноманітний функціонал, починаючи від базового моніторингу згадувань до складних методів аналізу великих обсягів даних. Серед популярних інструментів можна виділити такі платформи, як Hootsuite, Sprout Social і Brandwatch, які дозволяють компаніям стежити за згадками про свої продукти, аналізувати настрої споживачів і відстежувати тенденції ринку. Вони також пропонують інтеграцію з іншими бізнес-системами, що дозволяє отримувати повний огляд маркетингових кампаній та їхньої ефективності. Інші інструменти, такі як Talkwalker і NetBase Quid, спеціалізуються на глибокому аналізі даних, включаючи візуалізацію складних мережових взаємозв'язків та виявлення тональності повідомлень, що робить їх корисними для більш детальних і стратегічних аналізів.

Згідно з даними на рисунку 2.1, пайдіаграма відображає ринкову частку різних інструментів для аналізу соціальних мереж, де Hootsuite отримує найбільшу частку - 25%, що підтверджує її статус як однієї з найбільш популярних платформ. Sprout Social та Brandwatch також належать до категорії популярних інструментів, займаючи відповідно 20% та 18% ринку. Інструменти для глибокого аналізу, такі як Talkwalker і NetBase Quid, мають менші, але значущі частки - 15% та 12%

відповідно. Решта 10% ринку припадає на інші інструменти, які не були згадані в тексті. Ця діаграма ілюструє домінування кількох ключових гравців у сфері аналізу соціальних мереж, а також вказує на наявність значущого сегменту ринку для спеціалізованих інструментів глибокого аналізу.



Рисунок 2.1 – Частки ринку інструментів для аналізу соціальних мереж

Проте більшість існуючих інструментів для аналізу соціальних мереж є комерційними продуктами, що може стати бар'єром для малих компаній або дослідницьких проектів з обмеженим фінансуванням. Відкриті бібліотеки та фреймворки, такі як Instaloader, Tauri, Rust, HTML, CSS, JavaScript і chart.js, пропонують альтернативу комерційним рішенням. Вони забезпечують гнучкість і масштабованість, але вимагають додаткових навичок програмування та розробки, що може бути викликом для деяких організацій. Однак ці інструменти надають можливість глибокої кастомізації та розширення функціоналу відповідно до специфічних потреб, що робить їх привабливими для дослідників і розробників.

Важливим аспектом аналізу соціальних мереж є застосування сучасних методів аналізу даних, таких як аналіз тональності, виявлення трендів, кластеризація, аналіз мереж і мультимодальних даних. Ці методи дозволяють не

тільки отримати більш детальну інформацію про поведінку користувачів, але й виявити приховані патерни та взаємозв'язки між різними елементами соціальних мереж. Використання цих методів допомагає організаціям краще розуміти своїх клієнтів і швидше реагувати на зміни в ринку. Крім того, методи машинного навчання та штучного інтелекту, що застосовуються в аналізі соціальних даних, значно підвищують точність і ефективність прогнозування майбутніх тенденцій.

Загалом, аналіз даних соціальних мереж є потужним інструментом для будь-якої організації, яка прагне отримати конкурентну перевагу на ринку. Вибір конкретних інструментів та методів аналізу залежить від потреб і ресурсів організації, але поєднання комерційних та відкритих рішень, а також використання сучасних аналітичних методів може забезпечити максимальну ефективність і результативність. Окрім цього, важливо постійно оновлювати та вдосконалювати обрані методи та інструменти відповідно до змін у технологічному середовищі та потребах користувачів, щоб залишатися на передовій у цій швидко змінюваній сфері.

Перспективи подальшого розвитку методів аналізу соціальних мереж включають розробку більш інтегрованих рішень, які об'єднують аналіз даних з різних платформ і джерел, створення більш інтуїтивних і зручних інтерфейсів для користувачів, а також впровадження нових технологій, таких як аналіз відео та аудіо контенту. Зростаюча роль соціальних медіа у формуванні громадської думки та бізнес-стратегії робить аналіз цих даних ще більш важливим, і розвиток інструментів для їх ефективного аналізу буде ключовим фактором успіху в багатьох галузях.

2.2 Вибір технологій та засобів розробки кросплатформеного застосунку.

Розробка кросплатформеного застосунку вимагає ретельного вибору технологій та засобів, які забезпечують максимальну ефективність, продуктивність і масштабованість. Ключовими аспектами, які слід враховувати при виборі технологій, є вимоги до продуктивності, зручність розробки, підтримка різних

платформ і можливість інтеграції з іншими системами. Розробники часто обирають технології, які дозволяють зменшити час розробки та полегшити підтримку застосунку, що є критично важливим для швидкої адаптації до змін у ринку та потребах користувачів.

Однією з основних технологій для створення кросплатформеного застосунку є фреймворк Flutter, розроблений Google. Flutter дозволяє розробляти застосунки для Android та iOS з єдиною базою коду, використовуючи мову програмування Dart. Цей фреймворк забезпечує високу продуктивність і швидкий час виконання завдяки використанню власного механізму рендерингу. Крім того, Flutter надає широкий набір віджетів, які дозволяють створювати сучасний і привабливий інтерфейс користувача.

Іншим популярним фреймворком є React Native, створений Facebook. React Native використовує JavaScript і дозволяє розробляти мобільні застосунки з використанням тих самих принципів, що й у веб-розробці. Це робить його привабливим вибором для розробників, які вже знайомі з React. React Native забезпечує високу продуктивність і дозволяє використовувати нативні компоненти, що покращує взаємодію з платформою. Однією з головних переваг React Native є велика спільнота розробників і наявність численних бібліотек і плагінів.

Xamarin, розроблений компанією Microsoft, також є потужним інструментом для створення кросплатформених застосунків. Використовуючи мову C# та фреймворк .NET, Xamarin дозволяє створювати застосунки для Android, iOS та Windows з єдиною базою коду. Цей фреймворк забезпечує високий рівень продуктивності та інтеграції з нативними API платформ. Крім того, Xamarin надає можливість використання спільного коду між різними платформами, що значно зменшує час розробки та підтримки.

Для створення кросплатформених веб-застосунків часто використовуються фреймворки Angular та Vue.js. Angular, розроблений Google, надає потужні інструменти для створення масштабованих і продуктивних веб-застосунків. Він забезпечує двосторонню прив'язку даних, модульну структуру та підтримку TypeScript, що робить розробку більш структурованою та зрозумілою. Vue.js є

легшим і більш гнучким фреймворком, який дозволяє швидко створювати інтерфейси користувача та легко інтегруватися з іншими бібліотеками та проектами.

Electron є ще одним важливим інструментом для розробки кросплатформених застосунків, орієнтованих на настільні системи. Використовуючи веб-технології, такі як HTML, CSS та JavaScript, Electron дозволяє створювати настільні застосунки для Windows, macOS та Linux з єдиною базою коду. Це робить його привабливим вибором для розробників, які вже знайомі з веб-розробкою. Electron забезпечує доступ до нативних API, що дозволяє створювати функціональні та інтерактивні настільні застосунки.

При виборі технологій для розробки кросплатформеного застосунку важливо враховувати не лише технічні можливості, але й підтримку та документацію. Фреймворки з великою спільнотою розробників та активною підтримкою часто є кращим вибором, оскільки вони забезпечують швидке вирішення проблем та доступ до широкого спектру ресурсів і бібліотек. Крім того, наявність детальної документації полегшує процес навчання та прискорює розробку.

Вибір технологій також залежить від специфічних вимог проекту та цільової аудиторії. Наприклад, якщо основною метою є створення високопродуктивного мобільного застосунку з нативним інтерфейсом, то Flutter або React Native можуть бути кращим вибором. Якщо ж пріоритетом є швидка розробка та розгортання веб-застосунку, то Angular або Vue.js можуть забезпечити необхідну гнучкість і ефективність. Для настільних застосунків Electron є очевидним лідером завдяки своїй простоті та можливостям інтеграції з нативними функціями операційної системи.

У моєму проекті я обрав використання Tauri, Rust, HTML, CSS та JavaScript для розробки кросплатформеного застосунку. Tauri є сучасним фреймворком, який дозволяє створювати настільні застосунки з використанням веб-технологій, але з меншою вагою та кращою продуктивністю, ніж Electron. Використання Rust для бекенду забезпечує високу продуктивність, безпеку та ефективність пам'яті, що робить його ідеальним вибором для створення високопродуктивних застосунків.

HTML, CSS та JavaScript використовуються для розробки фронтенду, що дозволяє створювати інтуїтивний і привабливий інтерфейс користувача з багатим функціоналом.

Ринок кросплатформеного розробки включає різні технології з різними частками ринку. Як візуалізовано на рисунку 2.2, React Native займає 22% ринку, що робить його одним з лідерів. C# та .NET, які використовуються в Xamarin, мають частки ринку 17% і 12% відповідно. Flutter займає 13% ринку, тоді як мова Dart, на якій базується Flutter, займає 6%. Angular і Vue.js мають частки 12% і 7% відповідно, тоді як Electron також займає 7%. Ці частки показують, як різноманітні технології знаходять свої ніші в кросплатформеній розробці, і допомагають розробникам обирати найбільш підходящі інструменти для їхніх потреб.



Рисунок 2.2 – Частки ринку технологій кросплатформеного розробки

Отже, вибір технологій та засобів розробки кросплатформеного застосунку є складним і багатограним процесом, який вимагає врахування багатьох факторів, включаючи продуктивність, зручність розробки, підтримку різних платформ та можливість інтеграції. Використання сучасних фреймворків, таких як Flutter, React Native, Xamarin, Angular, Vue.js, Electron, а також Tauri у поєднанні з Rust, HTML,

CSS та JavaScript, дозволяє створювати ефективні, продуктивні та зручні застосунки, які відповідають вимогам сучасного ринку та потребам користувачів.

2.3 Аналіз середовищ розробки

Середовище розробки (IDE) є критично важливим інструментом для програмістів, оскільки воно забезпечує зручний і потужний інтерфейс, у якому вони можуть створювати, редагувати, налагоджувати та керувати своїми проектами. З величезною кількістю доступних IDE на ринку, вибір правильного середовища може бути складним завданням, що потребує ретельного аналізу функціональності, продуктивності та вподобань розробника.

На ринку існує кілька провідних IDE, які користуються широкою популярністю серед розробників завдяки своїм потужним функціям та підтримці різних мов програмування. Visual Studio від Microsoft є потужним середовищем для розробки на C#, C++, VB.NET та інших мовах .NET, пропонуючи безліч інструментів для підвищення продуктивності, налагодження та керування проектами. IntelliJ IDEA, розроблена компанією JetBrains, вважається однією з найкращих IDE для Java та інших мов на платформі JVM, з акцентом на зручність використання, автозавершення коду та потужні можливості рефакторингу. Для веб-розробників дуже популярними є Visual Studio Code від Microsoft та Sublime Text, які пропонують легкість, гнучкість та велику екосистему розширень.

Одна з ключових переваг сучасних IDE – їх розширюваність. Розробники можуть встановлювати плагіни, додатки та розширення, які розширюють функціональність IDE та адаптують її до своїх конкретних потреб і вподобань. Крім того, багато IDE тісно інтегруються з системами контролю версій, такими як Git або Subversion, спрощуючи роботу з репозиторіями коду, відстеженням змін та злиттям гілок.

Ефективне налагодження та відстеження помилок є критично важливим для швидкого усунення дефектів та забезпечення стабільності програмного забезпечення. Більшість сучасних IDE пропонують вбудовані інструменти для

налагодження, такі як крокове виконання, перегляд змінних, встановлення точок зупинки та візуалізація стеків викликів. Деякі IDE також інтегруються з інструментами для статичного аналізу коду, які допомагають виявляти потенційні проблеми, порушення стандартів кодування та антипатерни до запуску програми, забезпечуючи більш ефективний процес пошуку та усунення помилок.

Автозавершення коду та рефакторинг – це функції, які значно підвищують продуктивність розробників, допомагаючи їм писати код швидше та ефективніше. Автозавершення коду пропонує можливі варіанти завершення під час введення, базуючись на синтаксисі мови програмування, імпортованих бібліотеках та контексті коду, що прискорює процес написання та зменшує кількість помилок. Рефакторинг дозволяє реорганізувати існуючий код без зміни його функціональності, покращуючи його структуру, читабельність та відповідність кращим практикам програмування.

З поширенням хмарних технологій, деякі IDE пропонують безшовну інтеграцію з хмарними середовищами розробки, такими як AWS Cloud9, Microsoft Visual Studio Online або GitHub Codespaces. Ця інтеграція дозволяє розробникам працювати над своїми проектами з будь-якого пристрою, підключеного до Інтернету, не обмежуючись конкретним робочим місцем. Крім того, деякі IDE забезпечують можливості для віддаленої співпраці, дозволяючи кільком розробникам одночасно редагувати й налагоджувати код у режимі реального часу, підвищуючи ефективність командної роботи.

Хоча деякі IDE оптимізовані для певних мов програмування або технологічних стеків, інші пропонують широку підтримку різноманітних мов і фреймворків. Наприклад, Visual Studio Code та JetBrains IntelliJ IDEA пропонують плагіни, розширення та спеціалізовані редактори для багатьох мов програмування та технологій, дозволяючи розробникам працювати в одному середовищі з різними мовами, фреймворками та бібліотеками, забезпечуючи безшовну інтеграцію та уніфікований досвід розробки.

Сучасні IDE часто інтегруються з системами збірки та тестування, такими як Maven, Gradle, MSBuild або Bazel. Ця інтеграція дозволяє розробникам запускати

процеси збірки, виконувати юніт-тести та перевіряти коди стилю прямо з середовища розробки, спрощуючи процес розробки та забезпечуючи безперервну інтеграцію. Деякі IDE також інтегруються з інструментами для безперервної інтеграції та безперервного розгортання, такими як Jenkins, Travis CI або CircleCI, дозволяючи автоматизувати процеси збірки, тестування та розгортання програмного забезпечення.

Visual Studio Code (VS Code) від Microsoft стала однією з найпопулярніших і найшвидше зростаючих IDE для веб-розробників, розробників на різних мовах програмування та ентузіастів відкритого програмного забезпечення. Її перевагами є легкість, гнучкість, розширюваність за допомогою величезної екосистеми розширень, вбудована підтримка Git, інтегрований термінал, потужні можливості налагодження та багато іншого. Завдяки своїй відкритій архітектурі, активній підтримці спільноти та безперервному вдосконаленню, VS Code швидко перетворилася на улюблене середовище для багатьох розробників, пропонуючи безповний досвід розробки на різних платформах, операційних системах і для різноманітних технологій і проєктів. Зокрема, VS Code має чудову підтримку веб-технологій, таких як JavaScript, TypeScript, React, Angular та Vue.js, завдяки своїм потужним вбудованим інструментам та розширенням. Крім того, вона пропонує відмінну інтеграцію з хмарними середовищами розробки та системами безперервної інтеграції, що робить її ідеальним вибором для командної роботи та розробки у хмарі. З величезною і активною спільнотою, а також постійним потоком оновлень і вдосконалень, VS Code, безсумнівно, зміцнить свої позиції як одне з провідних середовищ розробки для широкого кола розробників у найближчому майбутньому.

2.4 Аналіз мов програмування, розмітки та стилів

Мови розмітки та стилів відіграють ключову роль у створенні веб-сторінок, забезпечуючи структуру, контент та візуальне оформлення. HTML (HyperText Markup Language) є основною мовою розмітки, яка використовується для

визначення структури веб-сторінки та її елементів, таких як заголовки, абзаци, посилання, зображення та інших. Ця мова забезпечує семантичний каркас для представлення контенту.

Доповнюючи HTML, CSS (Cascading Style Sheets) відповідає за стилізацію та візуальне оформлення веб-сторінок. За допомогою CSS можна задавати кольори, шрифти, розміри, відступи, фони та інші стилі для елементів HTML. Ця мова забезпечує гнучкість у створенні привабливих та естетичних інтерфейсів користувача, дозволяючи розробникам легко змінювати зовнішній вигляд веб-сторінок.

У зазначеному проєкті для створення адаптивного та кросбраузерного інтерфейсу користувача застосовується звичайний flexbox. Flexbox є потужним інструментом CSS для створення гнучких та адаптивних макетів. Він дозволяє легко розміщувати та вирівнювати елементи в контейнері, а також управляти їх розмірами та порядком відображення. Використання flexbox забезпечує відмінну адаптивність для різних розмірів екранів та пристроїв, що є критично важливим для створення сучасних та зручних для користувача веб-інтерфейсів.

Крім HTML, CSS та flexbox, можуть також використовуватися додаткові мови та технології для розширення функціональності та покращення досвіду користувача. Наприклад, JavaScript дозволяє додавати інтерактивність та динамічну поведінку на веб-сторінки. Фреймворки та бібліотеки, такі як React, Angular або Vue.js, надають зручні абстракції та компоненти для створення складних веб-додатків.

Важливо зазначити, що HTML, CSS та flexbox є ключовими технологіями для створення структури, стилізації та макетів веб-сторінок, але для повноцінного веб-розвитку потрібно також використовувати інші інструменти та технології. Наприклад, для забезпечення кросбраузерної сумісності та коректного відображення на різних пристроях часто використовуються CSS-препроцесори та постпроцесори, такі як Sass, Less або PostCSS.

Підсумовуючи, мови розмітки та стилів, такі як HTML та CSS, є основою для створення веб-сторінок та інтерфейсів користувача. Вони забезпечують структуру,

контент та візуальне оформлення, тоді як інструменти, такі як flexbox, надають гнучкість та адаптивність для різних пристроїв та розмірів екранів. Розробники часто поєднують ці технології з іншими інструментами та фреймворками для створення повноцінних веб-додатків, що відповідають сучасним вимогам та очікуванням користувачів.

Мови програмування є ключовим інструментом для розробників, оскільки вони забезпечують можливість створювати різноманітні програми та додатки. Кожна мова має свої унікальні особливості, переваги та сфери застосування. У цьому проєкті ми розглянемо такі мови, як Rust, JavaScript, C# та Kotlin.

Rust – це система програмування з акцентом на безпеку, продуктивність та паралельні обчислення. Вона була розроблена Mozilla Research для створення надійних та ефективних систем. Rust пропонує унікальний підхід до керування пам'яттю, гарантуючи безпеку та запобігаючи витокам пам'яті без використання автоматичного збирача сміття. Це робить її ідеальною для розробки системного програмного забезпечення, драйверів пристроїв, веб-браузерів та інших критичних для продуктивності додатків.

JavaScript, з іншого боку, є однією з найпопулярніших мов для веб-розробки. Спочатку створена для додавання інтерактивності на веб-сторінки, JavaScript зараз використовується як для фронтенду, так і для бекенду. Завдяки Node.js, розробники можуть використовувати JavaScript для створення серверних додатків, API та різноманітних інструментів. На фронтенді JavaScript використовується для створення динамічних інтерфейсів користувача, взаємодії з DOM та забезпечення інтерактивності веб-сторінок.

C# – це об'єктно-орієнтована мова програмування, розроблена Microsoft, яка широко використовується для створення різноманітних додатків, включаючи веб-додатки, ігри, мобільні додатки та корпоративні системи. Вона є частиною екосистеми .NET і забезпечує потужні можливості для розробки кросплатформених рішень. C# відомий своєю строгою типізацією, зручністю використання та широкою підтримкою різноманітних бібліотек та фреймворків.

Kotlin – це сучасна мова програмування, розроблена JetBrains, яка поєднує об'єктно-орієнтоване та функціональне програмування. Вона була спочатку створена для вдосконалення розробки Android-додатків, але зараз також використовується для розробки серверних додатків, веб-сервісів та інших проєктів на платформі JVM. Kotlin відомий своєю лаконічністю, безпекою та сумісністю з Java, що робить його привабливим вибором для розробників у різних областях.

Під час створення додатків на Tauri, Rust та JavaScript відіграють ключову роль. Rust може використовуватися для розробки ядра додатку, забезпечуючи високу продуктивність, безпеку та контроль над пам'яттю. Його можливості паралельних обчислень та низькорівнева природа роблять його ідеальним вибором для критичних для продуктивності компонентів.

З іншого боку, JavaScript може використовуватися для створення інтерфейсу користувача та забезпечення інтерактивності додатку. Завдяки своїй гнучкості та екосистемі бібліотек, таких як React або Angular, JavaScript дозволяє розробникам створювати динамічні та привабливі інтерфейси користувача. Крім того, JavaScript також може використовуватися на стороні сервера з Node.js для розробки REST API та інших бекенд-компонентів.

Поєднання Rust та JavaScript забезпечує потужний стек для створення високопродуктивних та інтерактивних додатків на таорі. Rust забезпечує надійність та ефективність ядра, тоді як JavaScript відповідає за створення зручного та динамічного інтерфейсу користувача. Ця комбінація дозволяє розробникам використовувати переваги обох мов, створюючи додатки, які поєднують високу продуктивність та привабливий користувацький досвід.

У світі обробки великих масивів даних існує низка потужних мов програмування, які забезпечують необхідну гнучкість, швидкість та ефективність для вирішення складних задач. Однією з найбільш визнаних та широко використовуваних мов у цій сфері є Python, який завдяки своїй простоті, зрозумілості та різноманітним бібліотекам став улюбленим інструментом для роботи з великими обсягами даних.

Python пропонує не лише зручний синтаксис та легкість у вивченні, а й потужні можливості для аналізу даних, машинного навчання та наукових обчислень. Численні бібліотеки, такі як NumPy, Pandas, Matplotlib та Scikit-learn, розширюють функціональність цієї мови та надають розробникам доступ до широкого спектру інструментів для ефективної обробки даних.

NumPy, наприклад, є фундаментальною бібліотекою для наукових обчислень у Python, яка забезпечує підтримку великих багатовимірних масивів та матриць, а також потужні математичні функції для роботи з ними. Ця бібліотека є основою для багатьох інших пакетів, орієнтованих на аналіз даних, і забезпечує ефективне виконання обчислень за рахунок оптимізації коду та можливості використання SIMD-інструкцій процесора.

Pandas, у свою чергу, є однією з найпопулярніших бібліотек для аналізу та маніпуляцій з табличними даними в Python. Вона пропонує зручні структури даних, такі як Series та DataFrame, які полегшують імпорт, очищення, перетворення та аналіз даних з різних джерел. Завдяки своїй потужності та гнучкості, Pandas став невід'ємною частиною багатьох проектів з обробки даних.

Matplotlib та бібліотеки візуалізації даних, такі як Seaborn та Plotly, також відіграють важливу роль у роботі з великими масивами даних у Python. Вони дозволяють створювати переконливі та інформативні візуалізації, які сприяють кращому розумінню структури та тенденцій у даних, полегшуючи процес аналізу та прийняття рішень.

Крім того, Python тісно інтегрується з бібліотеками для машинного навчання, такими як Scikit-learn та TensorFlow. Ці бібліотеки забезпечують потужні інструменти для розробки та впровадження моделей машинного навчання, дозволяючи ефективно обробляти та аналізувати величезні обсяги даних для вирішення складних задач, таких як класифікація, кластеризація та прогнозування.

Однією з ключових переваг Python для роботи з великими масивами даних є його багата екосистема та активна спільнота розробників. Завдяки відкритому коду та широкій підтримці, Python постійно розвивається та вдосконалюється,

забезпечуючи користувачів новими функціями, бібліотеками та інструментами для вирішення складних завдань з обробки даних.

Незважаючи на те, що інші мови програмування, такі як R, Java або Scala, також часто використовуються для аналізу даних, Python залишається одним з найпопулярніших та найбільш універсальних варіантів завдяки своїй простоті, потужності та гнучкості. Його широка підтримка в різних галузях, зокрема в науці, фінансах, маркетингу та технологічних дослідженнях, робить Python незамінним інструментом для фахівців з обробки даних у найрізноманітніших сферах.

2.5 Висновки до другого розділу

Підсумовуючи викладений матеріал, слід наголосити на важливості ретельного аналізу та вибору відповідних технологій, методів та інструментів для ефективного аналізу даних соціальних мереж і розробки кросплатформеного застосунку. У сфері аналізу соціальних мереж існує різноманітний спектр комерційних та відкритих рішень, кожне з яких має свої переваги та недоліки, що вимагає зваженого підходу до вибору інструментів відповідно до конкретних потреб та ресурсів організації.

Крім того, застосування сучасних методів аналізу даних, таких як аналіз тональності, виявлення трендів, кластеризація, аналіз мереж і мультимодальних даних, є критично важливим для отримання глибокого розуміння поведінки користувачів та швидкого реагування на зміни в ринку. У розвитку методів аналізу соціальних мереж очікується впровадження більш інтегрованих рішень, інтуїтивних інтерфейсів та аналізу відео- і аудіоконтенту, що відкриває нові перспективи для організацій у різних галузях.

У процесі розробки кросплатформеного застосунку вибір технологій та засобів розробки відіграє ключову роль, оскільки впливає на продуктивність, зручність розробки, підтримку різних платформ та можливість інтеграції з іншими системами. Фреймворк Taui, який поєднує переваги веб-технологій з високою продуктивністю та ефективністю Rust, постає привабливим вибором для створення

кроссплатформених настільних додатків. Використання Rust для розробки ядра забезпечує надійність, безпеку та контроль над пам'яттю, тоді як HTML, CSS і JavaScript дозволяють створювати інтуїтивний та інтерактивний інтерфейс користувача.

Важливим аспектом є також вибір середовища розробки (IDE), яке забезпечує зручний та ефективний інтерфейс для створення, редагування, налагодження та керування проектами. Середовища з розширеною функціональністю, такі як Visual Studio Code, набувають дедалі більшої популярності завдяки своїй легкості, гнучкості, розширюваності та чудовій підтримці веб-технологій.

Мови розмітки та стилів, зокрема HTML та CSS, відіграють ключову роль у створенні структури, контенту та візуального оформлення веб-сторінок та інтерфейсів. Використання таких інструментів, як flexbox, забезпечує гнучкість та адаптивність макетів для різних розмірів екранів та пристроїв, що є критично важливим для створення сучасних веб-інтерфейсів.

Вибір мов програмування також є визначальним фактором для успіху кроссплатформеного проекту. Поєднання Rust для розробки ядра додатку та забезпечення високої продуктивності з JavaScript для створення інтерактивного інтерфейсу користувача дозволяє досягти балансу між швидкістю та зручністю використання.

Python є однією з найбільш потужних та універсальних мов програмування для обробки великих масивів даних, завдяки своїй простоті, ефективності та гнучкості. Завдяки численним бібліотекам, таким як NumPy, Pandas, Matplotlib, Scikit-learn та TensorFlow, Python надає розробникам широкий спектр інструментів для аналізу, візуалізації та моделювання даних.

Його активна спільнота розробників та постійне оновлення роблять Python важливим інструментом для фахівців з обробки даних у різних галузях, від науки до фінансів, забезпечуючи їм необхідні інструменти для вирішення складних завдань у сфері аналітики та прогнозування.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАДАЧІ

3.1 Розробка парсера з орієнтацією на роботу на серверах під управлінням операційної системи Linux з використанням мови програмування Python та бібліотеки Instaloader

Консольний додаток, реалізований на Python із використанням бібліотеки Instaloader, здійснює парсинг профілів Instagram і зберігає зібрані дані у форматі JSON. Додаток спроектований за мікросервісною архітектурою, що забезпечує кілька вагомих переваг, серед яких можливість горизонтального масштабування, яке дозволяє значно підвищити продуктивність через розподіл навантаження на декілька серверів. Це особливо критично в умовах роботи в реальному часі, де стабільність і швидкість обробки даних мають вирішальне значення для функціонування системи.

Основні компоненти додатка та їх функціональність включають:

Парсер профілів Instagram. Використовуючи бібліотеку Instaloader, додаток здійснює автентифікацію в Instagram і надсилає запити до API платформи для отримання детальної інформації про користувацькі профілі, включаючи пости, кількість підписників, підписок, коментарі та інші метадані. Отримані дані зберігаються у форматі JSON, що спрощує їх подальше використання і інтеграцію з іншими системами, забезпечуючи універсальність і сумісність з різноманітними додатками та базами даних.

Використання регулярних виразів для фільтрування посилань та відокремлення юзернейму. Для підвищення ефективності та точності парсингу, додаток використовує регулярні вирази, що дозволяють фільтрувати посилання на Instagram та відокремлювати з них юзернейми. Регулярні вирази є потужним інструментом для пошуку і обробки текстових шаблонів, що значно спрощує виділення необхідної інформації з великих обсягів даних. У контексті даного додатка регулярні вирази дозволяють автоматично знаходити посилання на профілі

Instagram у тексті, ідентифікувати та витягувати з них юзернейми, що забезпечує точність і швидкість обробки даних.

Обробка даних та збереження. Зібрані дані підлягають попередній обробці, яка включає фільтрацію і нормалізацію, щоб забезпечити їх коректність і актуальність. Цей етап гарантує, що дані відповідають необхідним стандартам і готові до використання у різних зовнішніх додатках або зберігання у базах даних. Збереження у форматі JSON полегшує обробку великих обсягів даних та забезпечує швидкий доступ до них при потребі.

Горизонтальне масштабування. Однією з найважливіших переваг розробленого додатка є його здатність до горизонтального масштабування. Це досягається через використання розподілених систем, де кожен сервер виконує свою частину роботи з парсингу профілів. Такий підхід дозволяє значно прискорити обробку великої кількості запитів, що є критично важливим для забезпечення стабільної роботи у реальному часі. При збільшенні навантаження на систему можна додавати нові сервери, які працюватимуть паралельно з існуючими, забезпечуючи тим самим безперервну і швидку обробку даних без втрати продуктивності.

Моніторинг та логування. Для забезпечення стабільної роботи додатка передбачені модулі моніторингу та логування, які відстежують процеси парсингу, реєструють можливі помилки та забезпечують оперативне реагування на них. Це дозволяє підтримувати високий рівень продуктивності та мінімізувати простої у роботі системи. Логування надає детальну інформацію про роботу додатка, допомагаючи ідентифікувати і виправляти потенційні проблеми в найкоротші терміни, що сприяє підвищенню надійності і стабільності системи загалом.

Деталі реалізації. Парсер профілів Instagram, використовуючи Instaloader, виконує автентифікацію через запити до API, отримуючи доступ до інформації про користувачькі профілі, включаючи публікації, підписників, підписки та коментарі. Важливим аспектом є збереження зібраних даних у форматі JSON, що забезпечує легкість інтеграції з іншими додатками і системами. Крім того, обробка даних

включає їх фільтрацію та нормалізацію, що гарантує актуальність і коректність отриманої інформації, полегшуючи подальше використання.

Горизонтальне масштабування досягається через розподіл обробки запитів на декілька серверів, що дозволяє ефективно розподілити навантаження і забезпечити високу продуктивність навіть при значних обсягах даних. Моніторинг та логування є важливими компонентами системи, оскільки вони забезпечують відстеження процесів, реєстрацію помилок і швидке реагування на них, що дозволяє підтримувати стабільну роботу додатка і забезпечувати високу якість обслуговування.

Таким чином, консольний додаток на Python, який використовує бібліотеку `Instaloader` для парсингу профілів Instagram, представляє собою потужний інструмент для збору, обробки та збереження даних у форматі JSON, забезпечуючи масштабованість, стабільність та високу продуктивність завдяки продуманій архітектурі та використанню сучасних технологій моніторингу і логування. Використання регулярних виразів додає рівень ефективності та точності у процесі парсингу, дозволяючи швидко і точно витягувати необхідні дані з великих текстових масивів.

3.2 Розробка візуалізатора з використанням Tauri та HTML, CSS, JavaScript та Chart.js

Програма, створена на Tauri з використанням HTML, CSS, JavaScript та Chart.js, є важливим компонентом програмного комплексу, що забезпечує ефективну візуалізацію та аналіз даних. Вибір цих технологій обумовлений їх здатністю забезпечити високу продуктивність, крос-платформеність і сучасний інтерфейс користувача, що є ключовими вимогами до сучасних програмних продуктів.

Основні компоненти та їх функціональність:

Інтерфейс користувача (UI). Використовуючи HTML та CSS, створено зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко

взаємодіяти з додатком. Основні елементи інтерфейсу включають панелі навігації, форми для введення даних та таблиці для відображення результатів. HTML забезпечує структуру веб-сторінки, тоді як CSS відповідає за її стилізацію, забезпечуючи привабливий і функціональний дизайн. Крім того, адаптивний дизайн забезпечує коректне відображення інтерфейсу на різних пристроях, включаючи настільні комп'ютери, планшети та смартфони.

Логіка клієнтської частини. JavaScript забезпечує динамічність і інтерактивність інтерфейсу, дозволяючи здійснювати асинхронні запити до серверу для отримання даних, обробляти відповіді та оновлювати UI без необхідності перезавантаження сторінки. Це значно покращує користувацький досвід, забезпечуючи швидку та зручну роботу з додатком. JavaScript також використовується для реалізації різноманітних анімацій та ефектів, що робить взаємодію з додатком більш приємною та інтуїтивно зрозумілою.

Візуалізація даних. Бібліотека Chart.js використовується для створення різноманітних графіків і діаграм, що дозволяють користувачам наочно аналізувати зібрані дані. Завдяки цьому можна швидко отримати уявлення про тенденції та закономірності, що особливо корисно при аналізі великих обсягів інформації. Chart.js підтримує різні типи графіків, включаючи лінійні, стовпчасті, кругові діаграми та багато інших, що забезпечує гнучкість і універсальність у візуалізації даних. Крім того, бібліотека дозволяє налаштовувати вигляд графіків, забезпечуючи відповідність їх стилю загальному дизайну додатка.

Інтеграція з серверною частиною. Використовуючи API, клієнтська частина взаємодіє з серверною частиною додатка, що здійснює парсинг даних. Це забезпечує безшовну інтеграцію між різними компонентами системи та дозволяє ефективно використовувати зібрані дані для візуалізації та аналізу. Серверна частина, реалізована на Python з використанням бібліотеки Instaloader, виконує складну обробку і підготовку даних, які потім передаються на клієнтську частину для відображення. Такий підхід забезпечує високу ефективність та надійність роботи додатка.

Крос-платформеність. Завдяки використанню Tauri, додаток може працювати на різних операційних системах (Windows, macOS, Linux), що значно розширює його аудиторію. Крім того, Tauri забезпечує високий рівень безпеки та продуктивності, зменшуючи розмір виконуваних файлів та оптимізуючи використання системних ресурсів. Це досягається завдяки тому, що Tauri використовує рідні компоненти операційної системи для виконання додатка, що дозволяє значно зменшити обсяг споживаної пам'яті та ресурсів CPU у порівнянні з традиційними веб-технологіями.

Переваги використання Tauri. Однією з основних переваг використання Tauri є її легковагість та висока продуктивність. У порівнянні з іншими крос-платформеними фреймворками, такими як Electron, Tauri створює набагато менші виконувані файли, що дозволяє економити дисковий простір і зменшити час завантаження. Це особливо важливо для користувачів з обмеженими ресурсами та для розповсюдження додатка на різноманітних платформах.

Безпека також є ключовим аспектом Tauri. Завдяки тому, що додаток використовує рідні компоненти операційної системи, він має доступ до широкого спектру інструментів для забезпечення безпеки даних і користувачів. Це включає використання сучасних стандартів шифрування, захист від атак типу "людина посередині" (MITM) і забезпечення цілісності даних.

Сучасний інтерфейс та користувацький досвід. Інтерфейс додатка, створений за допомогою HTML та CSS, забезпечує сучасний вигляд і зручність використання. Використання адаптивного дизайну гарантує, що додаток коректно відображається на всіх пристроях, забезпечуючи безшовний користувацький досвід незалежно від розміру екрану або платформи. Динамічність інтерфейсу, забезпечена JavaScript, дозволяє користувачам здійснювати різноманітні дії без необхідності перезавантаження сторінки, що значно покращує загальне враження від роботи з додатком.

Chart.js [11], як інструмент для візуалізації даних, дозволяє створювати інтерактивні та інформативні графіки, що допомагає користувачам швидко аналізувати дані та приймати обґрунтовані рішення. Завдяки широким

можливостям налаштування, Chart.js дозволяє адаптувати вигляд графіків до специфічних вимог проекту, забезпечуючи їх відповідність загальному стилю додатка.

Інтеграція та масштабованість. Інтеграція клієнтської частини з серверною за допомогою API забезпечує безперервний обмін даними між різними компонентами системи. Це дозволяє ефективно використовувати зібрані дані для аналізу та візуалізації, забезпечуючи високу точність та актуальність інформації. Мікросервісна архітектура, використана у розробці серверної частини, дозволяє легко масштабувати додаток, додаючи нові сервери для обробки зростаючої кількості запитів. Це забезпечує стабільну роботу системи навіть при високих навантаженнях.

Використання регулярних виразів. Для підвищення ефективності та точності парсингу, додаток використовує регулярні вирази, що дозволяють фільтрувати посилання на Instagram та відокремлювати з них юзернейми. Регулярні вирази є потужним інструментом для пошуку і обробки текстових шаблонів, що значно спрощує виділення необхідної інформації з великих обсягів даних. У контексті даного додатка регулярні вирази дозволяють автоматично знаходити посилання на профілі Instagram у тексті, ідентифікувати та витягувати з них юзернейми, що забезпечує точність і швидкість обробки даних.

Таким чином, клієнтська програма, розроблена на основі Taui з використанням HTML, CSS, JavaScript та бібліотеки Chart.js, є потужним і ефективним інструментом для візуалізації та аналізу даних. Використання сучасних технологій і мікросервісної архітектури забезпечує високу продуктивність, безпеку, крос-платформеність і приємний користувацький досвід. Використання регулярних виразів для фільтрування посилань на Instagram та відокремлення юзернеймів додає рівень ефективності та точності у процесі парсингу, дозволяючи швидко і точно витягувати необхідні дані з великих текстових масивів.

3.3 Створення логотипу для програмного комплексу PostPrism

Створення логотипу – це складний і багатогранний процес, який вимагає не тільки творчого підходу, але й технічної майстерності. У цьому проекті було використано Blender [12] (див. рисунок 3.1) для моделювання та рендерингу логотипу, застосовуючи PBR матеріали, реалістичне освітлення та path trace рендер, а також ортографічну камеру для досягнення унікального та цікавого вигляду. Blender дозволяє створювати високоякісні тривимірні моделі з деталізацією, яка передає текстуру та глибину матеріалів, забезпечуючи реалістичність зображення. Після завершення роботи в Blender, використано Photoshop для остаточного доопрацювання логотипу, надаючи йому професійного вигляду. Використання Photoshop дозволяє додати фінальні штрихи, такі як корекція кольорів, підвищення контрастності та додавання ефектів, що роблять логотип більш виразним і привабливим для глядача.

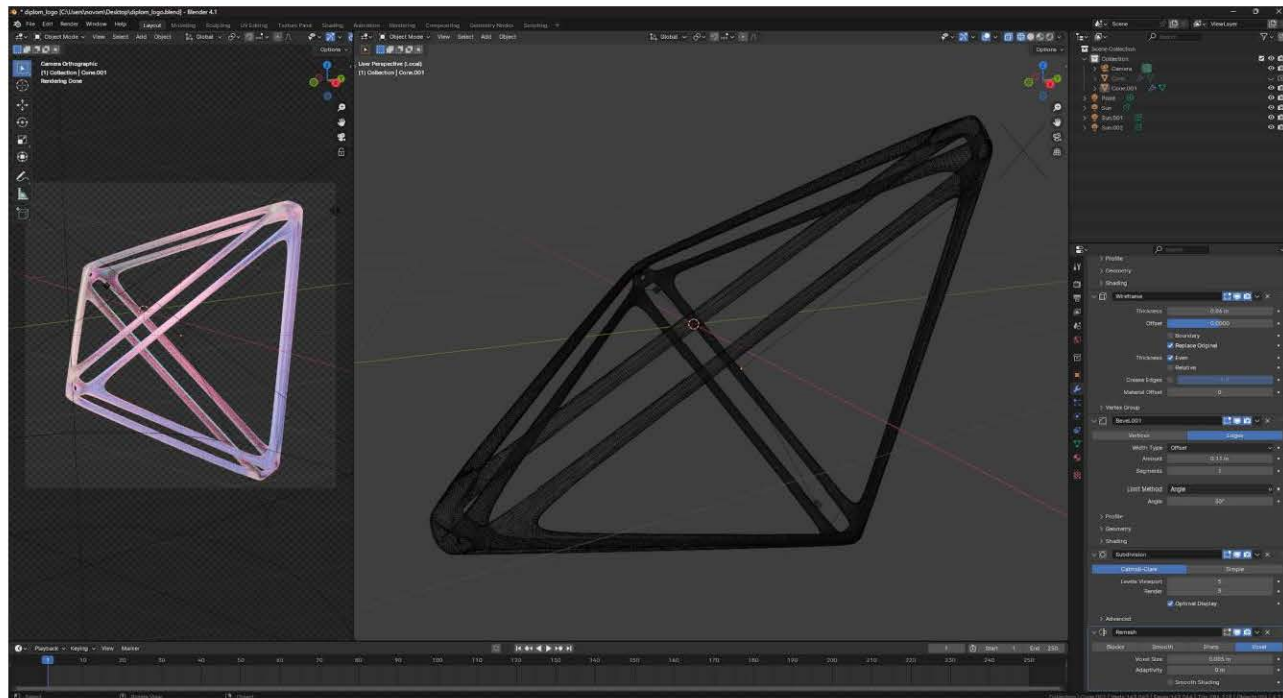


Рисунок 3.1 – Інтерфейс Blender

Створення сцени в Blender. Перша частина процесу полягала в створенні основної сцени в Blender.

Використовуючи потужні інструменти моделювання Blender, було створено базову форму логотипу, приділяючи особливу увагу його пропорціям та деталям. Для цього використано такі інструменти, як Extrude, Bevel та Subdivision Surface, що дозволили створити складну та деталізовану геометрію логотипу (див. рисунок 3.2). Крім того, важливим етапом було налаштування правильної топології моделі, що забезпечило чистоту та правильність геометрії, необхідні для подальших етапів роботи. Зокрема, ретельне опрацювання з'єднань та країв, а також відсутність н-гонів дозволило уникнути небажаних артефактів під час рендерингу. Цей етап роботи включав також тестування різних варіантів моделювання для вибору оптимальної форми, яка найкраще відображала задум дизайнера.

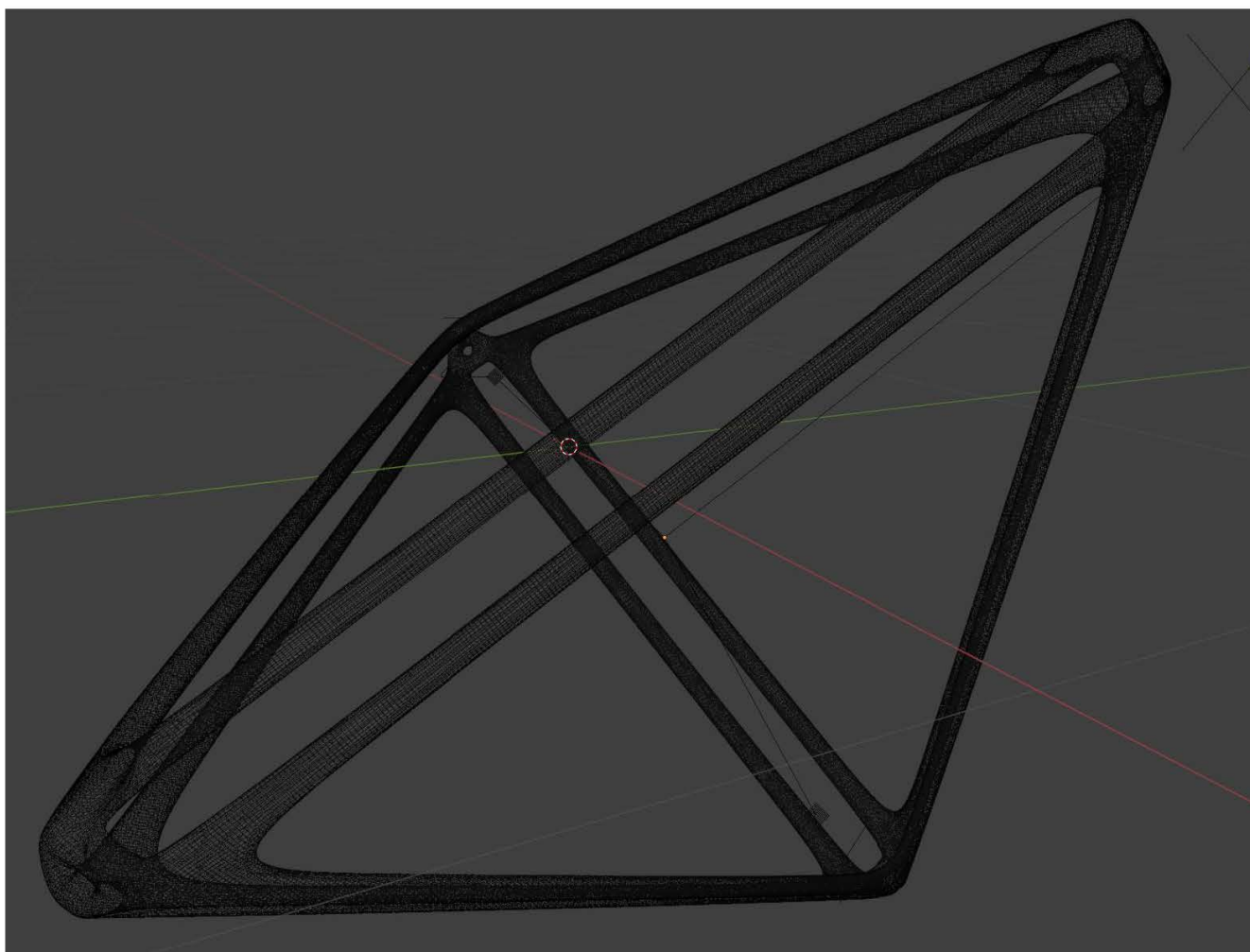


Рисунок 3.2 – Wire frame view (Геометрія моделі)

Після створення базової форми логотипу, було застосовано PBR (Physically Based Rendering) матеріалів (див рисунок 3.3). PBR матеріали дозволяють досягти високого рівня реалістичності завдяки тому, що вони враховують фізичні властивості матеріалів, такі як відбивна здатність, шорсткість і металевість. У цьому проекті було використано різні текстури, включаючи албедо, нормал, roughness та metallic мапи, щоб надати логотипу реалістичного вигляду. Особливу увагу було приділено налаштуванню кожної текстури окремо, щоб досягти оптимального результату, який би максимально точно відображав реальні матеріали. Наприклад, албедо текстура відповідає за базовий колір матеріалу, в той час як нормал мапа додає деталізації на поверхні, створюючи ефект нерівностей. Roughness мапа визначає рівень шорсткості поверхні, що впливає на те, як світло розсіюється при контакті з нею, а metallic мапа надає матеріалу металевого вигляду, що додає реалізму загальному зображенню. Поєднання цих текстур дозволило створити візуально насичений і реалістичний логотип, який виглядає правдоподібно з будь-якого кута огляду.

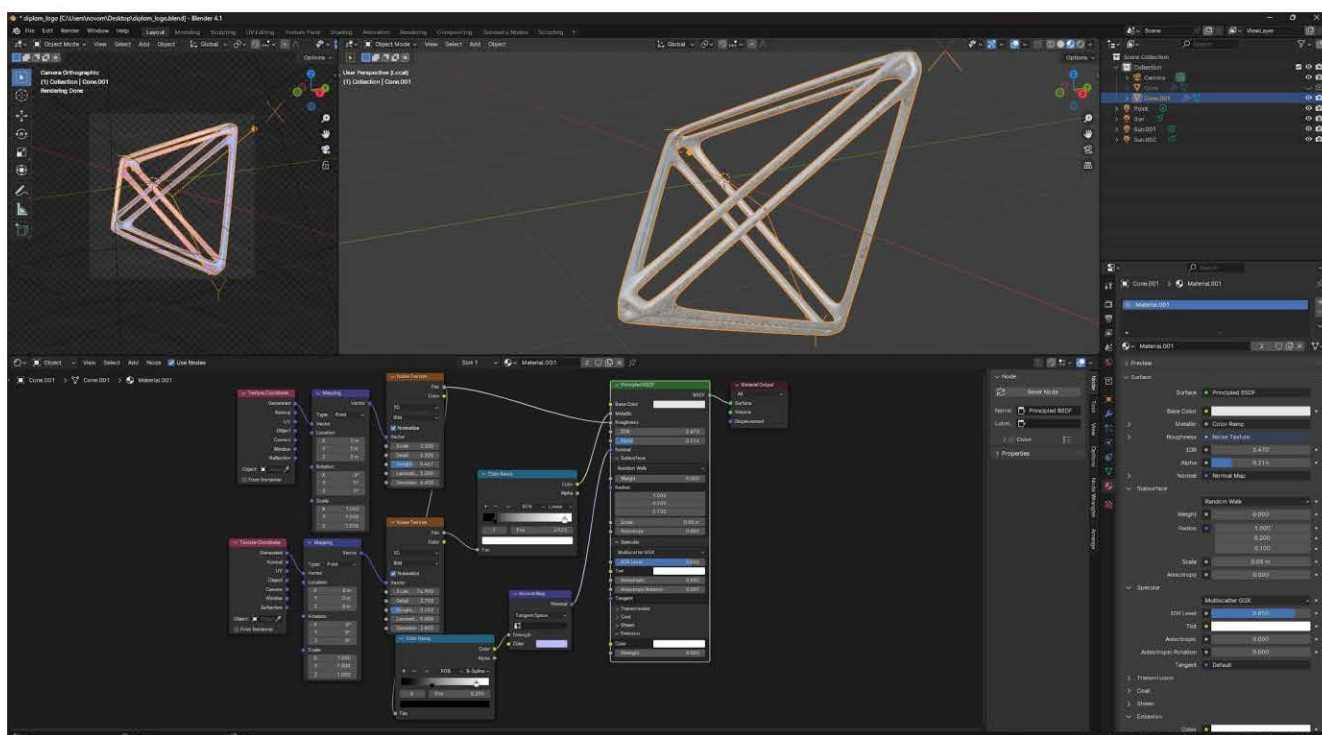


Рисунок 3.3 – налаштування PBR матеріалів

Реалістичне освітлення та path trace рендер. Для створення реалістичного освітлення було використано систему освітлення Blender. Основним джерелом світла стала HDRI-карта, яка забезпечила природне і м'яке освітлення сцени. HDRI-карта дозволяє досягти високого рівня реалістичності завдяки тому, що вона містить інформацію про світлові умови реального світу, включаючи напрямок та інтенсивність світла. Крім того, використання HDRI-карти дало можливість відтворити складні світлові ефекти, такі як відблиски та рефлексії, які додатково підсилюють глибину і реалістичність зображення.

Для додаткового підсвічування було використано декілька сфокусованих джерел світла (Spot Light), розташованих навколо логотипу. Це дозволило створити м'які тіні та підкреслити форми логотипу, додаючи йому об'єму і глибини (див рисунок 3.4). Таке розташування світильників забезпечило оптимальний баланс між освітленням і тінями, що сприяло візуальному виділенню ключових елементів логотипу. Завдяки комбінованому використанню HDRI-карти та додаткових джерел світла вдалося створити гармонійне освітлення, яке ефективно підкреслює деталі та текстури логотипу, надаючи йому професійного і завершеного вигляду.

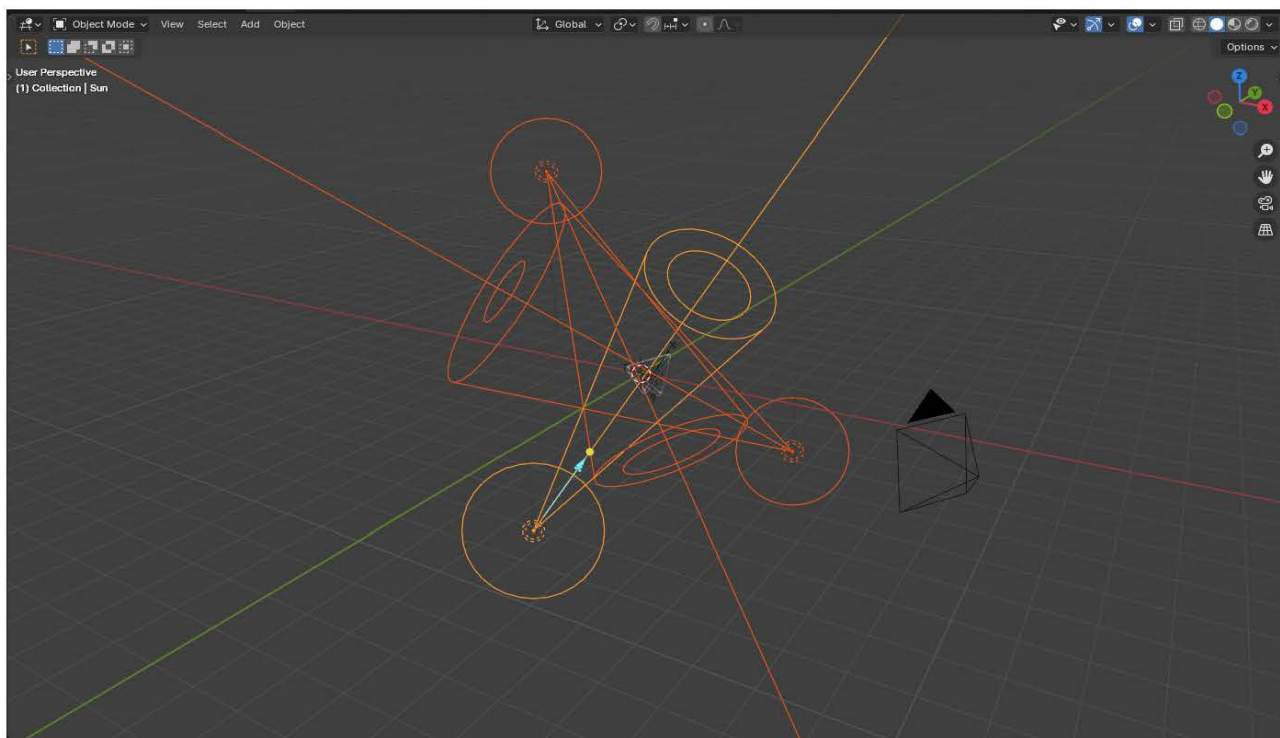


Рисунок 3.4- Схема освітлення логотипу

Важливим аспектом рендерингу було використання path trace рендеру, який є одним з найреалістичніших методів рендерингу. Path tracing симулює поведінку світла, проходячи крізь сцени і взаємодіючи з різними поверхнями. Це дозволяє досягти високої якості зображення, включаючи реалістичні відблиски, розсіювання світла та тіні. Path trace рендеринг є ресурсомістким процесом, але результати, які він дає, виправдовують витрати часу та обчислювальних ресурсів.

Використання ортографічної камери. Ортографічна камера (див. рисунок 3.5) була обрана для рендерингу логотипу, оскільки вона дозволяє створити вигляд без перспективних спотворень. Це особливо важливо для логотипів та графічних елементів, де точність і чистота ліній мають вирішальне значення. Ортографічна камера забезпечує паралельні лінії і дозволяє зосередити увагу на деталях та формах логотипу, надаючи йому більш технічного і професійного вигляду.

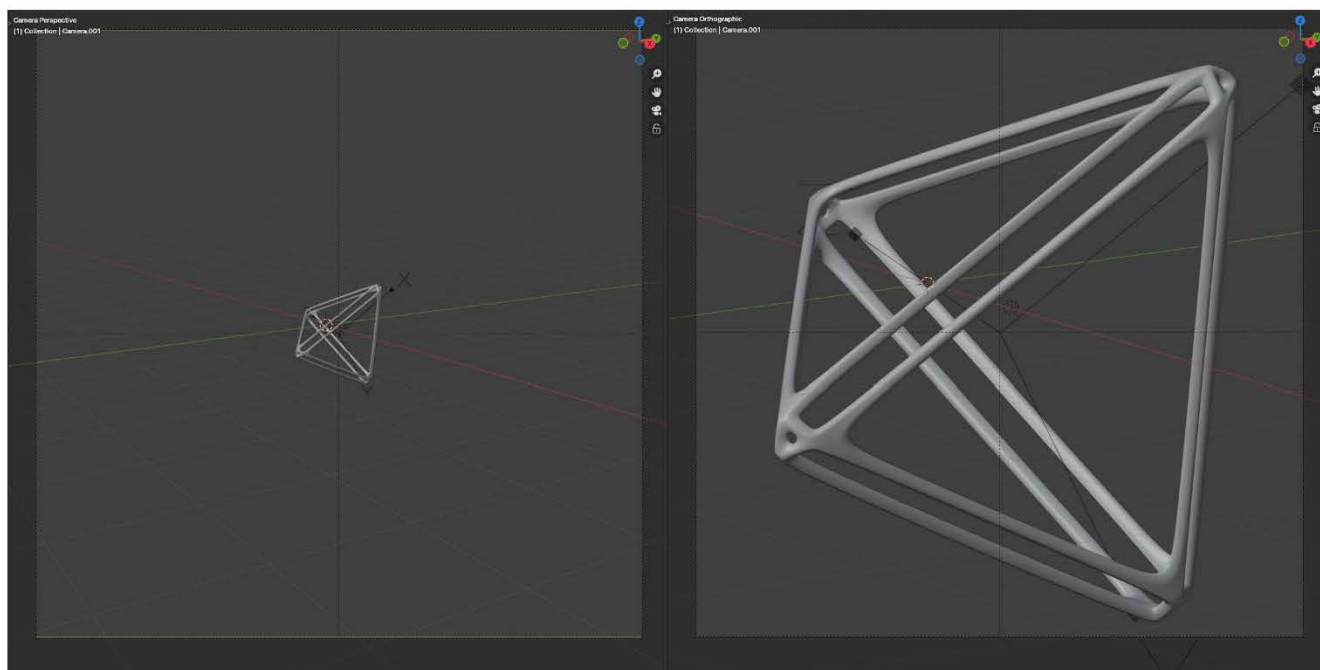


Рисунок 3.5- порівняння звичайної камери (фокусна відстань 50мм) та ортографічної

Постобробка в Photoshop. Після завершення рендерингу в Blender (див. рисунок 3.6), було зроблено постобробку зображення в Photoshop. Постобробка дозволяє додатково покращити вигляд логотипу, додаючи фінальні штрихи та

покращити кольори та читаємість форм . У Photoshop я використав кілька важливих інструментів та технік, щоб надати логотипу завершеного вигляду. Процес постобробки включав кілька ключових етапів, які були спрямовані на максимальне покращення візуальної привабливості логотипу.

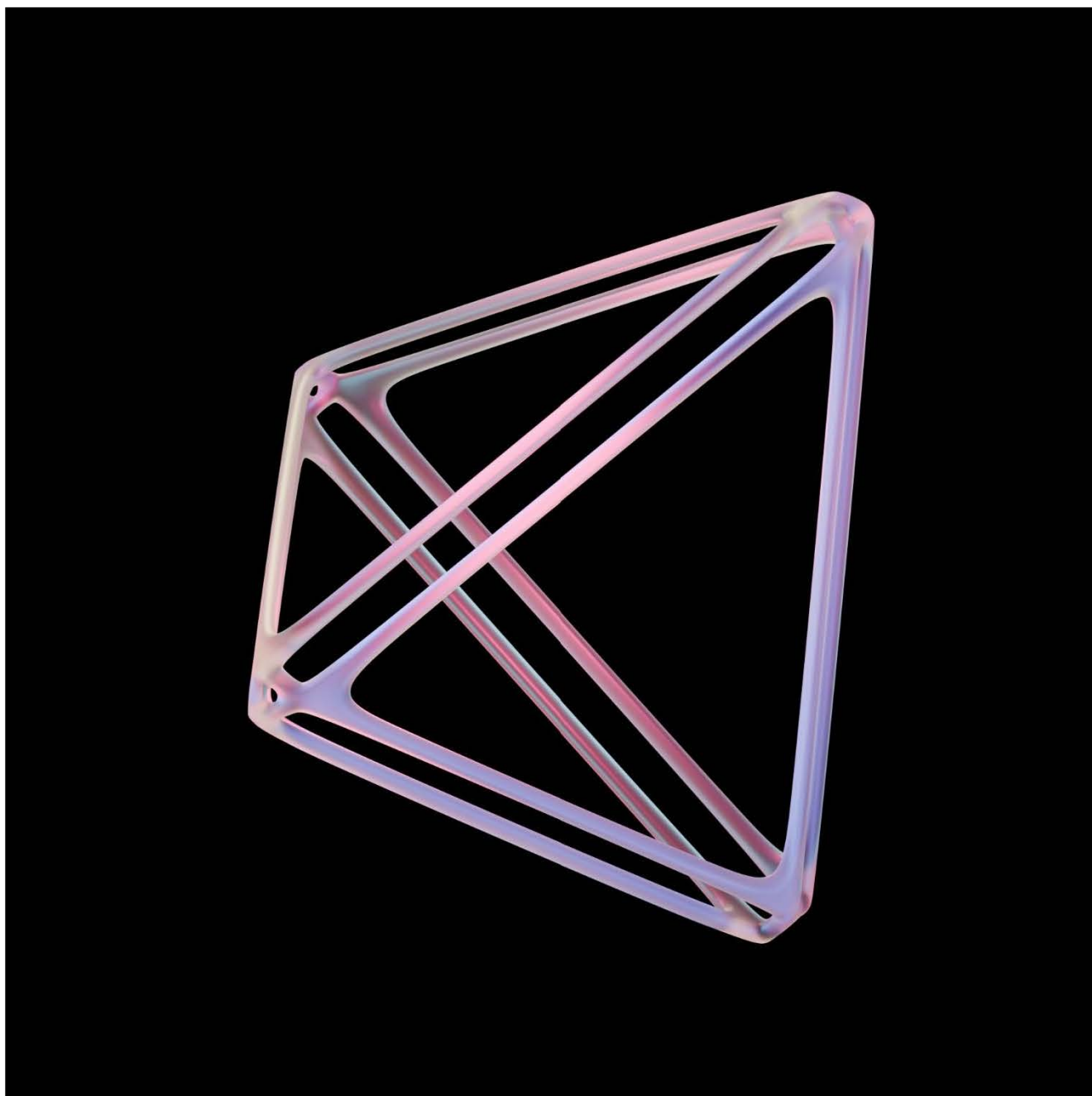


Рисунок 3.6 – Фінальний рендер

По-перше, було скореговано рівні (Levels) та криві (Curves) для покращення контрасту і яскравості зображення. Це дозволило зробити логотип більш виразним

і привертаючим увагу. Додатково, було застосовано інструменти для корекції кольору, щоб налаштувати колірну гаму логотипу, зробити його більш насиченим і гармонійним. В результаті таких змін логотип отримав більш живі та насичені кольори, що додало йому професійного вигляду і зробило його більш помітним на фоні.

Наступним кроком було додавання ефектів, таких як розмивання (Blur) і розсіювання (Glow), щоб створити м'які відблиски та підкреслити світлові ефекти. Це додало логотипу більшого об'єму та глибини, роблячи його більш реалістичним і привабливим (див. рис. 3.7). Застосування цих ефектів дозволило підсилити світлові акценти, що виникли під час рендерингу, та додати певного магічного відтінку, що робить логотип більш візуально приємним.

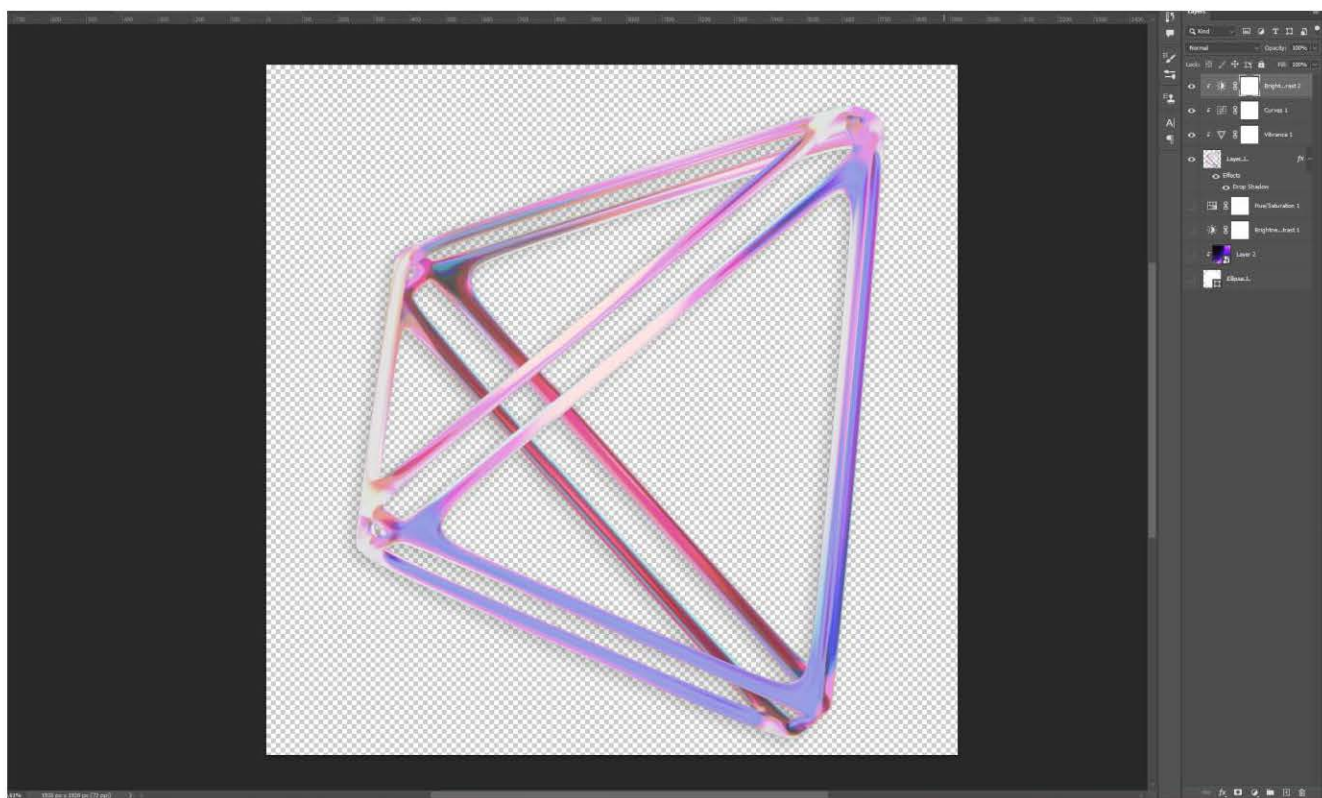


Рисунок 3.7- відредагований рендер

Використання додаткових технік у Photoshop. Додатково було використано такі інструменти, як маски (Masks) та шари корекції (Adjustment Layers), щоб локально змінювати окремі частини зображення, не впливаючи на інші. Це

дозволило досягти високої точності і контролю під час редагування логотипу. Маски дозволяють ізолювати окремі елементи логотипу для точного редагування, а шари корекції надають можливість експериментувати з різними ефектами, зберігаючи при цьому оригінальні дані зображення (див. рис. 3.8).

Використання масок дозволило виділити конкретні області логотипу для додаткового освітлення або затемнення, що допомогло створити більш збалансоване зображення. Крім того, шари корекції дозволили внести тонкі налаштування кольору та освітлення, що значно підвищило якість фінального результату. Таким чином, завдяки цим інструментам вдалося досягти максимальної деталізації і точної передачі задуму, що підкреслює унікальність та професійність створеного логотипу.

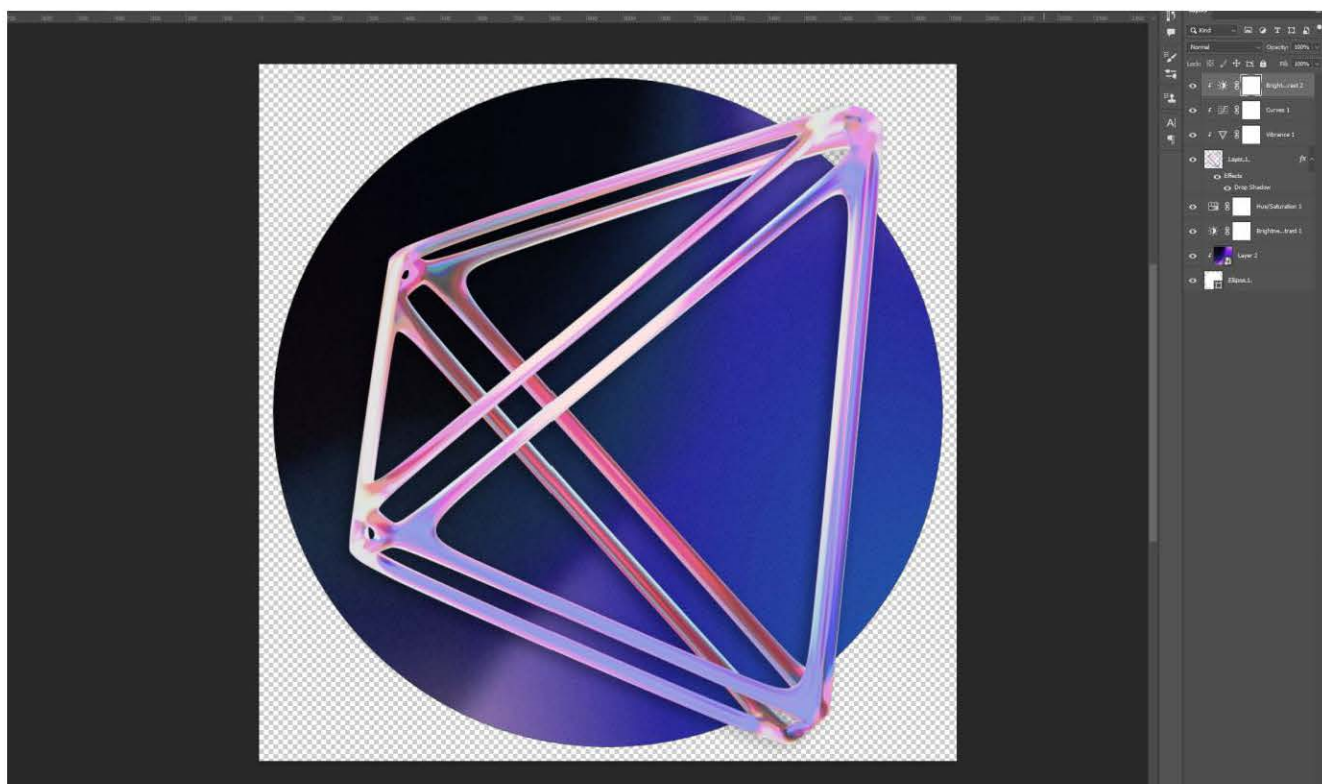


Рисунок 3.8 – Використання масок та шарів корекції

Створення логотипу (див. рисунок 3.9), за допомогою Blender та Photoshop – це багатоступеневий процес, який включає в себе моделювання, текстурювання, освітлення, рендеринг та постобробку. Використання PBR матеріалів,

реалістичного освітлення та path trace рендерингу дозволило створити високоякісне зображення, яке виглядає реалістично та професійно. Ортографічна камера забезпечила точність та чистоту ліній, що є важливим для логотипів. Постобробка в Photoshop дозволила додати завершальні штрихи та зробити логотип ще більш привабливим.

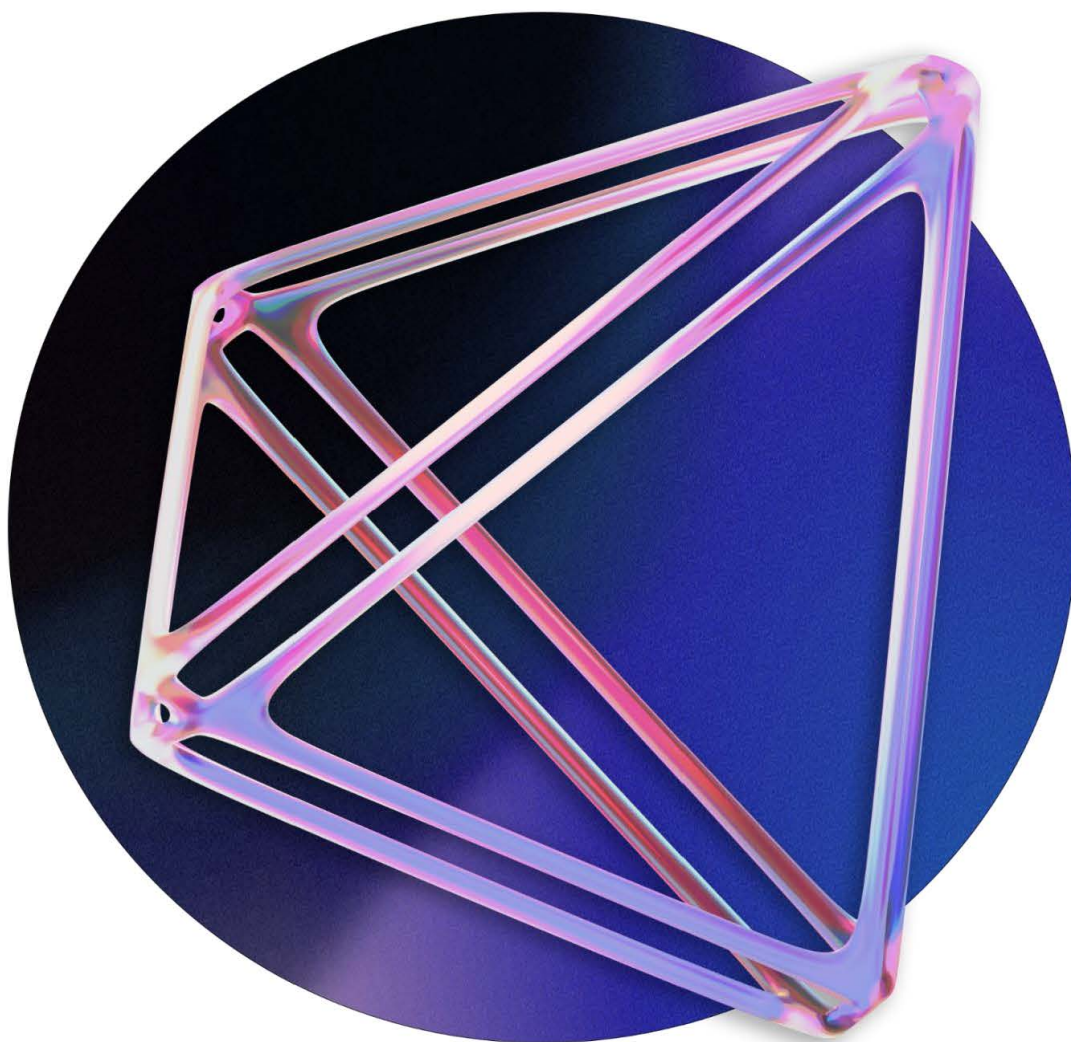


Рисунок 3.9 – Логотип PostPrism

В результаті цей процес продемонстрував, як сучасні інструменти і технології можуть бути ефективно використані для створення високоякісних графічних елементів, які можуть бути застосовані у різних контекстах, від веб-дизайну до друкованої продукції.

3.4 Висновки до третього розділу

У цьому розділі детально розглянуто два важливі компоненти програмного комплексу PostPrism: консольний парсер профілів Instagram та візуалізатор даних із сучасним інтерфейсом користувача. Крім того, описано створення логотипу програмного комплексу за допомогою 3D-модельовання та постобробки у Photoshop. Усі ці компоненти відіграють ключову роль у забезпеченні ефективного збору, обробки, візуалізації та аналізу даних, що є основною метою PostPrism.

Консольний парсер профілів Instagram, реалізований на Python із використанням бібліотеки Instaloader, є потужним інструментом для збору та обробки даних з популярної соціальної мережі. Його мікросервісна архітектура забезпечує горизонтальне масштабування, що дозволяє ефективно розподіляти навантаження між декількома серверами, підвищуючи продуктивність та стабільність системи загалом. Основні функції парсера включають автентифікацію в Instagram, надсилання запитів до API для отримання детальної інформації про користувацькі профілі, збереження даних у форматі JSON для подальшої обробки та інтеграції з іншими системами.

Особливу увагу приділено використанню регулярних виразів для фільтрування посилань на Instagram та відокремлення юзернеймів, що значно підвищує ефективність та точність парсингу. Цей потужний інструмент дозволяє автоматично знаходити необхідну інформацію у великих текстових масивах, економлячи час і зусилля. Крім того, передбачено модулі моніторингу та логування, які відстежують процеси парсингу, реєструють помилки та забезпечують оперативне реагування на них, що сприяє стабільній роботі системи та високій якості обслуговування.

Візуалізатор даних, створений за допомогою Tauri та HTML, CSS, JavaScript і Chart.js, є ключовим компонентом для ефективного відображення та аналізу зібраних даних. Використання цих сучасних технологій забезпечує високу продуктивність, крос-платформеність, сучасний інтерфейс користувача та зручну інтеграцію з серверною частиною комплексу. Інтуїтивно зрозумілий та адаптивний

інтерфейс, створений на HTML та CSS, робить взаємодію з додатком приємною та зручною для користувачів на різних пристроях. JavaScript додає динамічності та інтерактивності, дозволяючи здійснювати асинхронні запити до серверу та оновлювати інтерфейс без перезавантаження сторінки.

Візуалізація даних реалізована за допомогою Chart.js, потужної бібліотеки для створення різноманітних графіків і діаграм, що допомагає користувачам швидко виявляти тенденції та закономірності у великих обсягах інформації. Завдяки широким можливостям налаштування, Chart.js дозволяє адаптувати вигляд графіків до специфічних вимог проекту, забезпечуючи їх відповідність загальному стилю додатка.

Вибір Taui як основи клієнтської частини забезпечує високу продуктивність, безпеку та крос-платформеність додатка. Завдяки використанню рідних компонентів операційної системи, Taui створює набагато менші виконувані файли, ніж традиційні веб-технології, економлячи дисковий простір і час завантаження. Крім того, Taui надає доступ до широкого спектру інструментів для забезпечення безпеки даних і користувачів, включаючи використання сучасних стандартів шифрування та захист від атак типу "людина посередині".

Інтеграція клієнтської частини з серверною за допомогою API забезпечує безперервний обмін даними між різними компонентами системи, дозволяючи ефективно використовувати зібрані дані для аналізу та візуалізації. Мікросервісна архітектура серверної частини полегшує масштабування додатка, що є критично важливим для забезпечення стабільної роботи при зростаючих навантаженнях.

У розділі також детально описано процес створення логотипу програмного комплексу PostPrism за допомогою Blender та Photoshop. Використання PBR матеріалів, реалістичного освітлення та path trace рендерингу в Blender дозволило створити високоякісне зображення з реалістичним виглядом та професійним рівнем деталізації. Ортографічна камера забезпечила точність і чистоту ліній логотипу, що є важливим для графічних елементів такого типу.

Постобробка в Photoshop дозволила додати завершальні штрихи, скоригувати кольори та додати ефекти, такі як розмивання та розсіювання, для створення м'яких

відблисків і підкреслення світлових ефектів. Використання масок і шарів корекції надало високий рівень контролю та точності під час редагування окремих елементів логотипу.

Загалом, цей розділ демонструє, як сучасні технології та інструменти можуть бути ефективно використані для створення потужних та масштабованих програмних комплексів, здатних збирати, обробляти та візуалізувати великі обсяги даних. Мікросервісна архітектура, використання регулярних виразів, сучасних веб-технологій та 3D-моделювання забезпечують високу продуктивність, точність, візуальну привабливість та користувацький досвід. Ретельно продумана інтеграція між різними компонентами системи гарантує безперебійну роботу та можливість масштабування відповідно до зростаючих потреб. Таким чином, програмний комплекс PostPrism представляє собою передовий та надійний інструмент для аналізу даних, що відповідає сучасним вимогам та стандартам.

ВИСНОВКИ

Малий та середній бізнес в Україні відіграє надзвичайно важливу роль у загальній економіці країни, забезпечуючи понад 60% доданої вартості та більше 80% робочих місць. Сучасні тенденції розвитку малого та середнього бізнесу, що базуються на інноваційних підходах та адаптації до швидкоплинних змін, створюють сприятливе середовище для впровадження нових технологій, зокрема в сфері аналізу даних з соціальних мереж. Це дозволяє бізнесам швидше реагувати на зміни ринку, покращувати взаємодію з клієнтами, підвищувати ефективність маркетингових стратегій та збільшувати свою конкурентоспроможність.

Одним із перспективних напрямків є використання даних з Instagram для створення аналітичного застосунку, що може допомогти підприємцям краще розуміти ринкові тенденції, поведінку клієнтів та ефективність різних маркетингових стратегій. У даному дослідженні було детально розглянуто процес розробки такого програмного забезпечення, включаючи збір, обробку та аналіз даних, а також методи візуалізації та інтеграції цих даних у бізнес-процеси.

Основною метою роботи було створення програмно-аналітичного комплексу для аналізу ринку малого та середнього бізнесу на основі даних з Instagram. Для досягнення цієї мети були використані різні методи та підходи, включаючи методи веб-скрейпінгу, обробки природної мови, машинного навчання та візуалізації даних. В рамках дослідження було проведено огляд існуючих методів аналізу даних з соціальних мереж та визначено їх переваги та недоліки.

Під час дослідження були вивчені різні комерційні сервіси та платформи для аналізу даних з соціальних мереж, такі як Brandwatch, Hootsuite, Sprout Social, Synthesio, Crimson Hexagon, NetBase, Clarabridge та SocialBakers. Кожна з цих платформ пропонує різні функції та можливості для аналізу даних, включаючи моніторинг мереж, аналіз тональності, виявлення трендів, аналіз конкурентів та візуалізацію даних. Було здійснено порівняльний аналіз цих платформ, що дозволило визначити найбільш підходящі рішення для малого та середнього бізнесу в Україні.

Окрему увагу було приділено аналізу тональності та настроїв, аналізу тем та кластеризації, аналізу мереж та графів, виявленню подій та трендів, аналізу зв'язаних даних, мультимодальному аналізу даних та візуалізації даних. Було розглянуто різні алгоритми та методи, такі як Латентний розподіл Діріхле (LDA), k-means кластеризація, аналіз семантичної близькості, а також методи машинного навчання та лексичні ресурси.

На основі проведеного дослідження було розроблено технічне завдання для створення програмного комплексу, який поєднує серверний застосунок на Python та кросплатформений застосунок на Rust з використанням фреймворку Tauri та HTML, CSS, JavaScript. Було визначено ключові функції та можливості застосунку, що включають збір та обробку даних з Instagram, аналіз тональності та настроїв, виявлення трендів, аналіз конкурентів та візуалізацію даних.

Розроблений програмно-аналітичний комплекс має ряд переваг, таких як можливість швидкого реагування на зміни ринку, покращення маркетингових стратегій, підвищення залученості та лояльності клієнтів, а також оптимізація витрат та підвищення продуктивності бізнес-процесів. Інтеграція отриманих даних у бізнес-процеси дозволяє підприємцям приймати обґрунтовані стратегічні рішення, що сприяє зростанню та розвитку бізнесу.

У ході виконання кваліфікаційної роботи були виконані усі завдання, визначені на початку дослідження. Кожне з завдань було виконано наступним чином:

Проведено детальний огляд сучасних методів та підходів до аналізу даних соціальних мереж, таких як Facebook, X(Twitter) та Instagram, з акцентом на використання технік обробки природної мови, машинного навчання та аналізу даних. Цей огляд охопив процеси збирання, очищення, інтеграції та аналітичної обробки великих обсягів неструктурованих даних, що генеруються користувачами соціальних мереж, для виявлення корисних закономірностей, трендів, настроїв та поведінкових моделей. Зокрема, розглянуто методи аналізу тональності та настрою, які дозволяють визначити емоційне забарвлення тексту за допомогою машинного навчання, лексичних ресурсів та правил. Окрім цього, проаналізовано

техніки кластеризації та виявлення тем за допомогою алгоритмів Латентного розподілу Діріхле (LDA) та k-means, а також методи аналізу мереж та графів для виявлення впливових акторів та дослідження розповсюдження інформації в соціальних мережах.

Постановка завдання та визначення методологій, підходів та програмного забезпечення для аналізу даних соціальної мережі Instagram з метою розробки комплексного програмного продукту для аналізу та візуалізації даних. Основними етапами аналізу визначено парсинг інформації, очищення даних та їх візуалізація. Парсинг передбачав витягування необхідних даних з різноманітних джерел та їх перетворення у структуровану форму, тоді як очищення даних полягало у виявленні та виправленні помилок, пропущених значень та несумісних форматів для забезпечення послідовності та коректності даних. Візуалізація даних допомогла представити їх у вигляді графіків, діаграм та інших візуальних форматів для виявлення тенденцій та закономірностей.

Розробка нового програмного продукту на основі мови програмування Python з використанням бібліотеки Instaloader для написання парсеру, розрахованого на роботу на сервері під управлінням операційної системи Linux. Для візуалізації даних обрано фреймворк Taui та мову програмування Rust, а для створення інтерфейсу користувача використано HTML, CSS, JavaScript та фреймворк Chart.js. Це рішення забезпечило гнучкість, масштабованість та економічну ефективність нового продукту, поєднуючи переваги існуючих інструментів та усуваючи їх недоліки.

Розробка консольного парсера профілів Instagram, реалізованого на Python із використанням бібліотеки Instaloader, який дозволяє збирати та обробляти дані з популярної соціальної мережі. Парсер має мікросервісну архітектуру, що забезпечує горизонтальне масштабування і підвищує продуктивність та стабільність системи. Основні функції парсера включають автентифікацію в Instagram, надсилання запитів до API для отримання детальної інформації про користувачські профілі, збереження даних у форматі JSON для подальшої обробки та інтеграції з іншими системами.

Створення візуалізатора даних з використанням Tauri та HTML, CSS, JavaScript і Chart.js, що дозволяє ефективно відображати та аналізувати зібрані дані. Інтерфейс користувача, розроблений на основі цих технологій, забезпечує високу продуктивність, крос-платформеність та зручну інтеграцію з серверною частиною. Використання Chart.js дозволяє створювати різноманітні графіки і діаграми для швидкого виявлення тенденцій та закономірностей у великих обсягах інформації.

Інтеграція клієнтської частини з серверною за допомогою API, що забезпечує безперервний обмін даними між різними компонентами системи, дозволяючи ефективно використовувати зібрані дані для аналізу та візуалізації. Мікросервісна архітектура серверної частини полегшує масштабування додатка, що є критично важливим для забезпечення стабільної роботи при зростаючих навантаженнях.

Створення логотипу програмного комплексу PostPrism за допомогою Blender та Photoshop. Використання PBR матеріалів, реалістичного освітлення та path trace рендерингу в Blender дозволило створити високоякісне зображення з реалістичним виглядом та професійним рівнем деталізації. Постобробка в Photoshop додала завершальні штрихи, коригувала кольори та додала ефекти, такі як розмивання та розсіювання, для створення м'яких відблисків і підкреслення світлових ефектів.

Таким чином, результати даного дослідження демонструють високу ефективність використання даних з соціальних мереж для аналізу ринку малого та середнього бізнесу, а виконання кожного з зазначених завдань дозволило створити потужний та масштабований програмний комплекс PostPrism, який здатний ефективно збирати, обробляти, візуалізувати та об'єднувати аналіз великого обсягу даних, що генеруються користувачами Instagram. Запропонований програмно-аналітичний комплекс може стати потужним інструментом для українських підприємців, допомагаючи їм адаптуватися до швидкоплинних змін ринку, покращувати взаємодію з клієнтами та підвищувати свою конкурентоспроможність, оскільки мікросервісна архітектура, використання сучасних технологій та інтеграція різних компонентів системи забезпечили високу продуктивність, надійність та відповідність сучасним вимогам і стандартам у сфері аналізу даних соціальних мереж.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Урядовий портал Єдиний веб-портал органів виконавчої влади України. URL: <https://www.kmu.gov.ua/reformi/ekonomichne-zrostannya/pidtrimka-malogo-i-serednogo-pidpriyemnictva>
2. Instagram Users, Stats, Data, Trends, and More — DataReportal – Global Digital Insights, URL: <https://datareportal.com/essential-instagram-stats>
3. Instagram Graph API - Instagram-Plattform - Dokumentation - Meta for Developers. URL: <https://developers.facebook.com/docs/instagram-api/>
4. Python 3.12.3 Documentation. URL: <https://docs.python.org/3/>
5. Learn Rust. URL: <https://www.rust-lang.org/learn>
6. Guides | Tauri Apps. URL: <https://tauri.app/v1/guides/>
7. HTML: HyperText Markup Language | MDN. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>
8. CSS: Cascading Style Sheets | MDN. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>
9. JavaScript | MDN. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
10. Sentiment Analysis | AWS. URL: <https://aws.amazon.com/what-is/sentiment-analysis/>
11. Getting Started | Chart.js. URL: <https://www.chartjs.org/docs/latest/getting-started/>
12. Blender 4.1 Manual. URL: <https://docs.blender.org/manual/en/latest/>
13. Gomathy D. C. K. BIGDATA ANALYTICS IN INSTAGRAM. INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 2022. Vol. 06, no. 12. URL: <https://doi.org/10.55041/ijrsrem16821> (date of access: 02.06.2024).
14. Ukraine: Firms through the War. Washington, DC, 2024. URL: <https://doi.org/10.1596/40851> (date of access: 03.06.2024).

15. Git - Documentation. Git. URL: <https://www.git-scm.com/doc>
16. GitHub.com Help Documentation. GitHub Docs. URL: <https://docs.github.com/en>
17. Jenkins User Documentation. URL: <https://www.jenkins.io/doc/>
18. ollama/docs at main · ollama/ollama. GitHub. URL: <https://github.com/ollama/ollama/tree/main/docs>
19. Smith P. J. DATA SCIENCE for BEGINNERS: introduction to data science: python, coding, application, statistics, decision tree, neural network, and linear algebra. Independently Published, 2018.
20. Vickler A. Web development: Web development for Beginners in HTML. Ladoo Publishing LLC, 2021. 300 c.
21. Rust-101: main.rs. ralfj.de. URL: <https://www.ralfj.de/projects/rust-101/main.html>
22. Milanesi C. Beginning rust: from novice to professional. Apress, 2018. 396 c.
23. Crockford D. JavaScript: the good parts. Sebastopol, CA : O'Reilly, 2008. 153 c.
24. Simpson K. You don't know JS yet: get started. Primedia eLaunch LLC, 2020.
25. Matthes E. Python crash course, 3rd edition. No Starch Press, Incorporated, 2022.
26. Mckinney W. Python for data analysis: data wrangling with pandas, numpy, and ipython. O'Reilly Media, Incorporated, 2012.
27. Simpson A., Shovic J. Python All-in-One For Dummies. For Dummies, 2019. 704 c.
28. Wolf J. HTML und CSS: Das umfassende Handbuch zum Lernen und Nachschlagen. Inkl. JavaScript, Bootstrap, Responsive Webdesign u. v. m. Rheinwerk Verlag GmbH, 2021.

29. Duckett J. HTML & CSS: Design and build websites. 2014. 490 c.
30. Jasim M. Building Cross-Platform Desktop Applications with Electron: Create impressive cross-platform desktop applications with Electron and Node. Packt Publishing, 2017. 336 c.
31. Eshwarla P. Rust Servers, Services, and Apps. Manning Publications Co. LLC, 2022.

ДОДАТОК А

python

```

import os
import argparse
from datetime import datetime
import json
import re
import time
import instaloader
from instaloader import Profile
from instaloader.exceptions import ProfileNotExistsException

def parse_arguments():
    parser = argparse.ArgumentParser(description='Instagram profile data scraper.')
    parser.add_argument('-d', '--directory', type=str, default='./data/',
help='Directory to save the output files')
    parser.add_argument('-p', '--posts', type=int, default=50, help='Number of posts
to retrieve per profile')
    parser.add_argument('-t', '--timeout', type=int, default=5, help='Delay between
processing each profile in seconds')
    parser.add_argument('-i', '--input', type=str, default='./links.txt', help='Input
file containing Instagram profile URLs')
    return parser.parse_args()

def extract_username(url: str) -> str | None:
    """Extract username from Instagram URL."""
    pattern = r"https?:\/\/www\.instagram\.com\/([^\?\/]+)\/?.*"
    match = re.match(pattern, url)
    return match.group(1) if match else None

def get_profile_data(L, USERNAME: str, postAmount: int = 50) -> dict:
    """Get profile data and posts from Instagram."""
    try:
        profile = Profile.from_username(L.context, USERNAME)
    except ProfileNotExistsException:
        print(f"Profile '{USERNAME}' does not exist.")
        return None
    except Exception as e:
        print(f"Error retrieving profile for {USERNAME}: {e}")
        return None

    posts = profile.get_posts()
    post_count = 0
    post_data = []
    for post in posts:
        print(f"Post {post_count + 1} out of {postAmount} ({posts.count})")
        post_data.append({
            'url': post.url,
            'caption': post.caption,
            'likes': post.likes,
            'comments': post.comments,
            'pinned': post.is_pinned,
            'date': post.date,

```

```

        'is_video': post.is_video,
        'video_view_count': post.video_view_count if post.is_video else None
    })

    post_count += 1

    if post_count == postAmount:
        break

    if post_count % 10 == 0:
        print("waiting..")
        time.sleep(6)

post_data.sort(key=lambda x: x['date'], reverse=True)

total_likes = 0
for post in post_data:
    total_likes += post['likes']

return {
    'username': profile.username,
    'profile_picture': profile.profile_pic_url,
    'full_name': profile.full_name,
    'biography': profile.biography,
    'external_url': profile.external_url,
    'followers': profile.followers,
    'followees': profile.followees,
    'total_likes': total_likes,
    'is_business_account': profile.is_business_account,
    'business_category_name': profile.business_category_name,
    'mediacount': profile.mediacount,
    'userid': profile.userid,
    'present_of_scanned_posts': min(100, postAmount/(posts.count*0.01)),
    'time_of_scan': datetime.now(),
    'posts': post_data,
}

def convert_datetime(obj):
    if isinstance(obj, datetime):
        return obj.isoformat()
    raise TypeError(f"{repr(obj)} is not JSON serializable")

def main():
    args = parse_arguments()

    L = instaloader.Instaloader()

    # Load URLs from the input file
    with open(args.input, 'r') as file:
        url_list = [line.strip() for line in file if line.strip()]

    profiles = []
    total_profiles = len(url_list)

```

```

for idx, url in enumerate(url_list):
    username = extract_username(url)
    if not username:
        print(f"Invalid URL: {url}")
        continue

    print(f"Parsing profile {idx + 1} out of {total_profiles}")
    profile = get_profile_data(L, username, args.posts)
    if profile:
        profiles.append(profile)

    if (idx + 1) < total_profiles:
        time.sleep(args.timeout)

if profiles:
    os.makedirs(args.directory, exist_ok=True)
    num_profiles = len(profiles)
    filename = f"{datetime.now().strftime('%Y-%m-%d')}-{num_profiles}-1.json"
    num = 1
    while os.path.exists(os.path.join(args.directory, filename)):
        num += 1
        filename = f"{datetime.now().strftime('%Y-%m-%d')}-{num_profiles}-
{num}.json"

    with open(os.path.join(args.directory, filename), 'w', encoding='utf-8') as
f:
        json.dump(profiles, f, indent=4, default=convert_datetime,
ensure_ascii=False)

    print(f"JSON file saved to: {os.path.join(args.directory, filename)}")

if __name__ == "__main__":
    main()

```


ДОДАТОК Б

Html & css

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JSON to Divs</title>
  <link rel="stylesheet" href="styles.css">
  <script defer src="main.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script> <!--
Include Axios -->
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script> <!-- Include
Chart.js -->
</head>
<body>
  <div id="file-container">
    <input type="file" id="file-input" accept=".json">
    <button id="generate-button" disabled>Generate new report</button>
    <button id="append-button" disabled>Append</button>
  </div>
  <div id="table-container"></div>
</body>
</html>

```

```

#file-container{
  display: flex;
  justify-content: center;
  align-items: center;
}

#file-container button{
  background-color: black;
  color: #fff;
  border: none;
  margin: 10px;
  padding: 10px 15px;
  font-size: 16px;
  cursor: pointer;
  border-radius: 5px;
  transition: background-color 0.3s ease-in-out;
}

#file-container button:hover {
  background-color: #fa71cd;
  transition: background-color 0.3s ease-in-out;
}

#file-input::file-selector-button{
  background-color: black;

```

```

    color: #fff;
    border: none;
    padding: 10px 15px;
    font-size: 16px;
    cursor: pointer;
    border-radius: 5px;
    transition: background-color 0.3s ease-in-out;
}

#file-input::file-selector-button:hover {
    background-color: #fa71cd;
    transition: background-color 0.3s ease-in-out;
}

body{
    font-family: Segoe UI;
}

.row {
    display: flex;
    margin-bottom: 20px;
    border: 1px solid #ddd;
    padding: 10px;
    border-radius: 10px;
}

.column {
    flex: 1;
    padding: 10px;
}

.profile-pic {
    width: 50px;
    height: 50px;
    border-radius: 50%;
    transition: border-radius 0.3s;
}

.profile-pic:hover {
    border-radius: 0;
}

.small-text {
    font-size: 0.8em;
}

.arrow {
    display: inline-block;
    width: 20px;
    height: 20px;
    margin-right: 10px;
}

.arrow-up {

```

```

    background-color: green;
    clip-path: polygon(50% 0%, 0% 100%, 100% 100%);
}

.arrow-down {
    background-color: red;
    clip-path: polygon(50% 100%, 0% 0%, 100% 0%);
}

.post-image {
    width: 50px;
    height: 50px;
    margin: 5px;
    border: 1px solid transparent;
    transition: border 0.3s;
}

.post-image.pinned {
    border: 2px solid gold;
}

/* Column 1 */
.face{
display: flex;
flex-flow: column;
align-items: center;
justify-content: center;
flex-grow: 1;
}

.face img{
    width: 100px;
    height: 100px;
    padding: 10px 0px;
}

.scanTime{
    font-size: small;
    display: flex;
    flex-flow: column;
}

/* Column 2 */
.info{
    display: flex;
    flex-flow: column;
    align-items: center;
    justify-content: center;
    flex-grow: 1;
}

.info div{
    display: flex;
    flex-flow: column;

```

```
margin: 5px 0;
align-items: center;
}

.info span {
  text-align: center;
}

.info .present{
  font-size: small;
}

.chart{
  flex-grow: 4;
}

.gallery{
  flex-grow: 2;
}

.username {
  font-family: Segoe UI Semibold;
  text-decoration: none;
  color: #262626;
  font-size: 14px;
  font-weight: bold;
}

.username:hover {
  text-decoration: underline;
  color: #262626;
}
```

ДОДАТОК В

Java script

```

document.addEventListener('DOMContentLoaded', () => {
  const fileInput = document.getElementById('file-input');
  const generateButton = document.getElementById('generate-button');
  const appendButton = document.getElementById('append-button');
  const tableContainer = document.getElementById('table-container');
  let jsonData = null;

  fileInput.addEventListener('change', event => {
    const file = event.target.files[0];

    if (file) {
      const reader = new FileReader();

      reader.onload = (e) => {
        const content = e.target.result;

        try {
          jsonData = JSON.parse(content);
          generateButton.disabled = false; // Enable the button
          appendButton.disabled = false; // Enable the append button
        } catch (error) {
          alert('Invalid JSON file');
          generateButton.disabled = true; // Disable the button
          appendButton.disabled = true; // Disable the append button
        }
      };

      reader.readAsText(file);
    }
  });

  generateButton.addEventListener('click', () => {
    if (jsonData) {
      generateDivs(jsonData);
    }
  });

  appendButton.addEventListener('click', () => {
    if (jsonData) {
      appendDivs(jsonData);
    }
  });

  const fetchImageUrl = async (imageUrl) => {
    try {
      const { invoke } = window.__TAURI__.tauri;
      const imageData = await invoke('fetch_image', { imageUrl });
      const blob = new Blob([new Uint8Array(imageData)], { type: 'image/jpeg' });

      return URL.createObjectURL(blob);
    } catch (error) {

```



```

        console.error('Error fetching image:', error);
        return null;
    }
};

const generateDivs = async (json) => {
    if (Array.isArray(json)) {
        tableContainer.innerHTML = ''; // Clear existing content
        await createDivs(json);
    } else {
        alert('JSON must be an array of objects');
    }
};

const appendDivs = async (json) => {
    if (Array.isArray(json)) {
        await createDivs(json);
    } else {
        alert('JSON must be an array of objects');
    }
};

const createDivs = async (json) => {
    for (const item of json) {
        const rowDiv = document.createElement('div');
        rowDiv.classList.add('row');

        const profilePictureUrl = await fetchImageUrl(item.profile_picture);

        const col1 = document.createElement('div');
        col1.classList.add('column', 'face');
        col1.innerHTML = `
            <div class='scanTime'>
                <span>Time of Scan:</span>
                </span>${new Date(item.time_of_scan).toLocaleDateString('en-US', {
year: 'numeric', month: 'short', day: 'numeric' })} at ${new
Date(item.time_of_scan).toLocaleTimeString('en-US', { hour: '2-digit', minute: '2-
digit', hour12: false })}</span>
            </div>
            
            <a href="https://www.instagram.com/${item.username}/" target="_blank"
class="username">@${item.username}</a>
            <div class="small-text">
                ${item.is_business_account ? 'is business account | ' +
item.business_category_name : ''}
            </div>
        `;

        const col2 = document.createElement('div');
        col2.classList.add('column', 'info');
        col2.innerHTML = `
            <div>
                <span>Followers: ${item.followers}</span>
                <span>Followees: ${item.followees}</span>

```

```

    </div>
    <div>
      <span>Likes: ${item.total_likes}</span>
      <span class='present'>based on: ${item.present_of_scanned_posts}% of
posts</span>
    </div>
    <div>Total: ${item.mediacount} posts</div>
  `;

  const col4 = document.createElement('div');
  col4.classList.add('column', 'chart');
  const canvas = document.createElement('canvas');
  col4.appendChild(canvas);

  const reversedPosts = item.posts.slice().reverse();

  new Chart(canvas, {
    type: 'bar',
    data: {
      labels: reversedPosts.map((_, idx) => `Post ${item.posts.length -
idx}`),
      datasets: [
        {
          type: 'line',
          label: 'Likes',
          data: reversedPosts.map(post => post.likes),
          borderColor: 'blue',
          fill: false,
        },
        {
          label: 'Comments',
          data: reversedPosts.map(post => post.comments),
          backgroundColor: 'rgba(0, 123, 255, 0.5)',
        },
      ],
    },
  });

  const col5 = document.createElement('div', 'gallery');
  col5.classList.add('column');

  for (const post of item.posts.slice(0, 8)) {
    const postImageUrl = await fetchImageUrl(post.url);
    const img = document.createElement('img');
    img.src = postImageUrl;
    img.classList.add('post-image');
    if (post.pinned) img.classList.add('pinned');
    col5.appendChild(img);
  }

  rowDiv.appendChild(col1);
  rowDiv.appendChild(col2);
  rowDiv.appendChild(col4);
  rowDiv.appendChild(col5);

```

```
        tableContainer.appendChild(rowDiv);  
    }  
};  
});
```

ДОДАТОК Г

rust

```
// Prevents additional console window on Windows in release, DO NOT REMOVE!!
#![cfg_attr(not(debug_assertions), windows_subsystem = "windows")]

use tauri::Manager;
use reqwest::Client;
use serde_json::json;

#[tauri::command]
async fn fetch_image(image_url: String) -> Result<Vec<u8>, String> {
    match reqwest::get(&image_url).await {
        Ok(response) => match response.bytes().await {
            Ok(bytes) => Ok(bytes.to_vec()),
            Err(err) => Err(format!("Error reading response bytes: {}", err)),
        },
        Err(err) => Err(format!("Error fetching image: {}", err)),
    }
}

#[tauri::command]
async fn generate_description(name: String, biography: String, is_business_account:
bool, category: Option<String>, post_descriptions: String) -> Result<String, String>
{
    let client = Client::new();
    let prompt = format!(
        "Write a brief description of this Instagram profile based on its name: {}.
Biography: {}. {} Posts descriptions: {}",
        name,
        biography,
        if is_business_account {
            format!("This is a business account and its category is: {}",
category.unwrap_or_default())
        } else {
            String::new()
        },
        post_descriptions
    );

    let body = json!({
        "model": "llama3",
        "prompt": prompt
    });

    match client.post("http://localhost:11434/api/generate")
        .json(&body)
        .send()
        .await {
        Ok(response) => match response.text().await {
            Ok(text) => Ok(text),
            Err(err) => Err(format!("Error reading response text: {}", err)),
        },
        Err(err) => Err(format!("Error sending request: {}", err)),
    }
}
```

```
    }  
}  
  
fn main() {  
    tauri::Builder::default()  
        .invoke_handler(tauri::generate_handler![fetch_image, generate_description])  
        .run(tauri::generate_context!())  
        .expect("error while running tauri application");  
}
```