

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Автоматизоване надсилання персоналізованих запитів інвесторами і компаніям через інтеграцію з Crunchbase та OpenAI  
(тема)

Виконав:  
здобувач четвертого року навчання,  
групи ІТШ-21-5

Данііл Євтухов  
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
Освітня програма Штучний інтелект  
(повна назва освітньої програми)

Керівник ас. Марія Погурська  
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ \_\_\_\_\_  
(підпис)

Олег ЗОЛОТУХІН  
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_

Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_

Освітня програма \_\_\_\_\_ Штучний інтелект \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Євтухову Даніілу Олександровичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Автоматизоване надсилення персоналізованих запитів інвесторами і компаніям через інтеграцію з Crunchbase та OpenAI \_\_\_\_\_

затверджена наказом університету від 19 травня 2025 р. № 378Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вихідні дані до роботи Інтеграція з Crunchbase для збору даних про компанії, використання OpenAI для генерації листів та обробки листування, а також розробка веб-системи, що автоматизує пошук роботи або інвесторів на основі персональних даних користувача.

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

1) Аналіз предметної галузі \_\_\_\_\_

2) Методи, технології та підходи \_\_\_\_\_

3) Розробка та реалізація серверної частини \_\_\_\_\_

4) Розробка візуальної частини \_\_\_\_\_



## РЕФЕРАТ

Пояснювальна записка: 71 с., 22 рис., 1 дод., 17 джерел.

БАЗА ДАНИХ, ПАРСЕР, ШТУЧНИЙ ІНТЕЛЕКТ, AZURE, CRUNCHBASE, FIGMA, OPENAI, PYTHON, REACTTS, TYPESCRIPT.

Об'єктом дослідження є процес автоматизації комунікації між спеціалістами/стартапами та потенційними інвесторами або роботодавцями за допомогою сучасних інформаційних технологій. Особлива увага приділяється цифровим платформам, що дозволяють використовувати бази даних, такі як Crunchbase, у поєднанні з інструментами штучного інтелекту.

Предметом дослідження виступає розробка веб-платформи на основі технологій Python, TypeScript, ReactTS та Azure, яка інтегрує штучний інтелект (OpenAI) для парсингу даних із Crunchbase, формування персоналізованих листів і моніторингу комунікації з потенційними партнерами. Додатково досліджується UX-дизайн у Figma та ефективність реалізації парсера для збору інформації.

Метою роботи є створення концепції та прототипу інноваційної AI-платформи, яка дозволяє автоматизовано шукати потенційних інвесторів, роботодавців або партнерів, надсилати їм персоналізовані запити та відслідковувати результативність комунікації на основі даних із Crunchbase з використанням сучасних мов програмування і хмарних сервісів.

У роботі використовуються методи аналізу і синтезу інформаційних технологій, проектування програмного забезпечення, а також методи UX/UI-дослідження у Figma. Технічна реалізація включає розробку парсера на Python, застосування ReactTS і TypeScript для фронтенду, інтеграцію з OpenAI API для генерації текстів, використання Azure як хмарного середовища, а також створення та управління базою даних для зберігання результатів комунікації.

## **ABSTRACT**

Bachelor's thesis contains: 71 pp., 22 fig., 1 ann., 17 references.

ARTIFICIAL INTELLIGENCE, AZURE, CRUNCHBASE, DATABASE, FIGMA, OPENAI, PARSER, PYTHON, REACTTS, TYPESCRIPT.

The object of the research is the process of automating communication between professionals/startups and potential investors or employers using modern information technologies. Particular attention is paid to digital platforms that enable the use of databases such as Crunchbase in combination with artificial intelligence tools.

The subject of the research is the development of a web platform based on Python, TypeScript, ReactTS, and Azure technologies, which integrates artificial intelligence (OpenAI) for parsing data from Crunchbase, generating personalized emails, and monitoring communication with potential partners. Additionally, UX design in Figma and the effectiveness of implementing a data parser are examined.

The goal of the project is to create a concept and prototype of an innovative AI platform that allows automated search for potential investors, employers, or partners, sends them personalized requests, and tracks communication outcomes based on data from Crunchbase using modern programming languages and cloud services.

The project applies methods of information technology analysis and synthesis, software design techniques, and UX/UI research using Figma. The technical implementation includes developing a parser in Python, using ReactTS and TypeScript for the frontend, integrating with the OpenAI API for text generation, utilizing Azure as a cloud environment, and creating and managing a database to store communication results.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	7
Вступ.....	8
1 Аналіз предметної галузі .....	10
1.1 Проблематика працевлаштування молоді .....	10
1.2 Пошук роботи за допомогою ШІ.....	12
1.3 Цифрова B2B-комунікації для інвесторів і компаній .....	13
1.4 Сучасні підходи до агрегування даних про компанії .....	15
1.5 Автоматизація персоналізованих запитів.....	18
1.6 Роль AI/LLM у генерації релевантних комунікацій .....	20
2 Методи, технології та підходи .....	23
2.1 Теоретичні аспекти обраних підходів.....	23
2.2 Визначення функціональних вимог .....	25
2.3 Визначення технічних та програмних ресурсів .....	25
3 Розробка та реалізація серверної частини .....	28
3.1 Розробка парсера для отримання даних із Crunchbase.....	28
3.2 Хмарна реалізація механізму збору та оновлення даних.....	30
3.3 Фільтрація даних та IMAP протокол .....	32
3.4 Інтеграція штучного інтелекту для генерації листів .....	36
3.5 Розробка REST API для взаємодії з фронтендом.....	40
4 Розробка візуальної частини .....	43
4.1 Розробка дизайну у середовищі Figma .....	43
4.2 Реалізація фронт-енд частини.....	52
Висновки .....	66
Перелік джерел посилання .....	68
Додаток А Відомість кваліфікаційної роботи .....	71

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

AI – Artificial Intelligence – штучний інтелект;

IT – Information Technology – інформаційні технології;

LLM – Large Language Models – великі мовні моделі;

UI – User Interface – інтерфейс для користувача;

UX – User Experience – користувацький досвід.

## ВСТУП

В сучасному світі питання працевлаштування набуває все більшого значення в нашому швидкоплинному світі, оскільки цифрові технології постійно змінюють все. Галузь інформаційних технологій, яка зазвичай виділяється як одна з найбільш яскравих і перспективних сфер, відчуває сильну конкуренцію на ринку праці. ІТ-сектор постійно потребує нових спеціалістів, але шукачі роботи стикаються зі зростаючими труднощами через насичений пул кандидатів та економічну нестабільність. Професіонали повинні приділяти багато часу пошуку вакансій і створенню резюме, підбираючи супровідні листи та проходячи численні процеси відбору.

Нестабільний ринок праці в поєднанні зі зростанням безробіття спонукає фахівців до розробки нових методів самореклами та створення професійних мереж. Спеціалісти-початківці, а також стартапи та учасники міжнародного ринку вважають ці виклики особливо важливими. Інструмент для автоматизації рутинних завдань із одночасним підвищенням ефективності спілкування з роботодавцями та інвесторами стає важливим у цьому середовищі.

Незважаючи на те, що війна та світова криза створили складні соціально-економічні умови, ринок праці в Україні демонструє сигнали відновлення. Останні дослідження показують, що рівень зайнятості зріс, що свідчить про те, як населення повільно адаптується до нових обставин. Процес відновлення супроводжується значним психологічним стресом і нестабільністю, а також гострою конкуренцією, особливо в таких затребуваних сферах, як ІТ.

І те, що ІТ-сектор допускає віддалену роботу та забезпечує гнучкість, його робоча сила стикається з проблемами, оскільки кількість спеціалістів, які шукають роботу, зростає, але наявних посад не завжди достатньо для задоволення цього попиту. Шукачі повинні самостійно шукати

роботодавців і регулярно оновлювати свої профілі, беручи участь у нетворкінгових заходах і надсилаючи численні листи, на які часто не відповідають.

У таких умовах особливо актуальною стає потреба в інструменті, який здатен автоматизувати процес комунікації з компаніями, допомогти обрати релевантні вакансії та створити ефективну систему зворотного зв'язку. Саме поєднання технологій штучного інтелекту, хмарних сервісів і великих баз даних може стати рішучим кроком до модернізації процесу працевлаштування.

Зростання кількості зайнятого населення є позитивною тенденцією, проте за цими цифрами стоїть інтенсивна боротьба за кожне робоче місце. Особливо це стосується молодих фахівців, які лише починають свій кар'єрний шлях, а також спеціалістів, які змінюють професію або прагнуть перейти на новий рівень у своїй сфері. Ринок вимагає не лише глибоких технічних знань, а й вміння себе презентувати, швидко реагувати на зміни та бути постійно включеним у процес самопромоції.

Класичні методи пошуку роботи – перегляд вакансій на сайтах, надсилання резюме, написання мотиваційних листів – дедалі частіше виявляються малоефективними. У відповідь на це зростає інтерес до інноваційних рішень, які дозволяють підвищити видимість кандидата на ринку праці, автоматизувати рутинні завдання та встановити зв'язок із потенційними роботодавцями без зайвих посередників.

Зважаючи на вищезазначене, актуальною є розробка універсального цифрового інструменту, який би надавав змогу фахівцям ефективно взаємодіяти з компаніями, скориставшись перевагами штучного інтелекту, парсингу даних з відкритих джерел (зокрема Crunchbase), а також сучасними вебтехнологіями. Такий підхід здатен суттєво знизити бар'єри при пошуку роботи, особливо у висококонкурентних середовищах, і зробити процес працевлаштування більш прозорим, цілеспрямованим та зручним.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Проблематика працевлаштування молоді

У сучасному світі, де глобальні процеси та динаміка змін торкаються всіх сфер життя, питання зайнятості молоді набуває особливої актуальності. Це питання потребує уваги на національному та міжнародному рівнях, оскільки впливає на соціально-економічний розвиток країн, інтеграцію молодих спеціалістів на ринку праці та майбутнє суспільства в цілому. Актуальність теми зумовлена стрімкими змінами у світовій економіці, викликами глобалізації, а також необхідністю адаптації законодавчих та освітніх систем до сучасних вимог ринку праці.

В Україні відбувається зростання кількості працевлаштованих громадян, попри втому і напруженість, що переважають у настроях населення (рисунок 1.1). Все більше опитаних українців мають роботу і працюють повний або частковий робочий день (69% опитаних декларують це зараз, проти 62% у грудні 2023 року) [1].

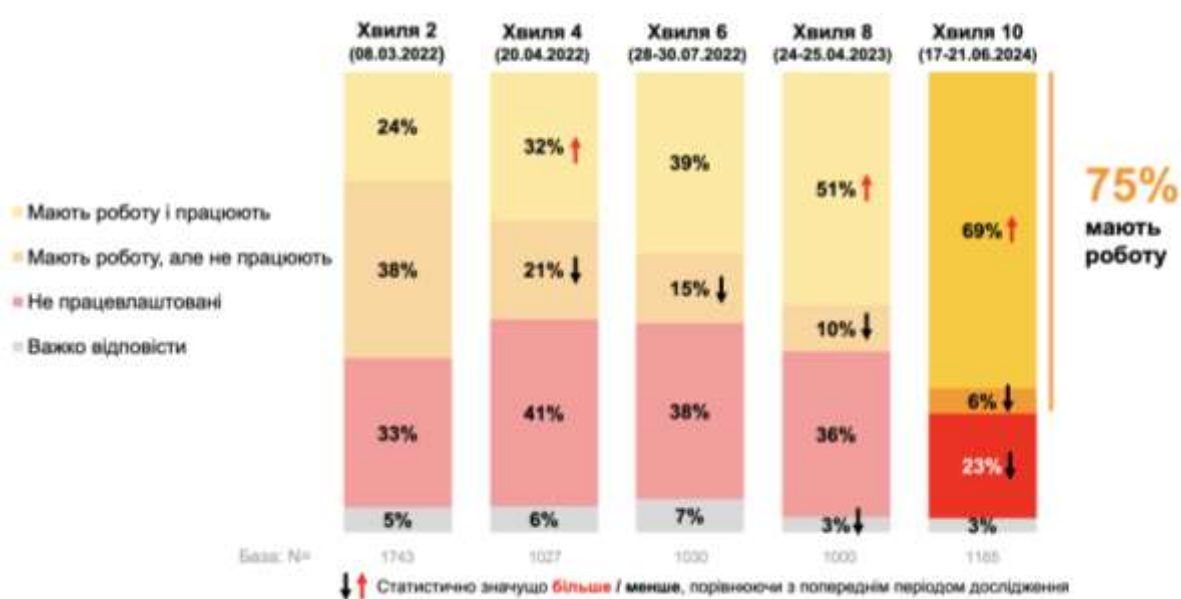


Рисунок 1.1 – Працевлаштованість під час війни

В умовах України, де суспільно-політичні та економічні процеси набувають особливої гостроти, питання працевлаштування молоді набуває ще більшої ваги, з огляду на воєнний стан та його вплив на ринок праці [2].

У сучасну цифрову епоху ІТ-галузь є найдинамічнішим і найперспективнішим сектором економіки. Він розвивається з кожним днем, створюючи нові можливості для працевлаштування для великих технологічних корпорацій, а також для стартапів, що швидко розвиваються. При цьому розвиток супроводжується сильним розвитком конкуренції серед кандидатів. Питання особливо актуальне серед молодих спеціалістів, які починають свою кар'єру, та експертів, які намагаються перейти на роботу в іноземних проектах.

Сучасні методи пошуку роботи – це ручний перегляд вакансій на професійних сайтах, складання резюме та мотиваційних листів, співбесіди та відвідування професійних заходів. Однак підхід довготривалий, нечасто ефективний і вимагає постійного оновлення інформації про себе. Крім того, роботодавці зазвичай не мають достатньо часу для ознайомлення з великою кількістю кандидатів, тому заявка просто не знаходить адресата або проходить без попередження.

На думку науковців, варто розглянути поняття працевлаштування, як у широкому, так і у вузькому розумінні. У широкому розумінні – це процес будь-якого влаштування на роботу, у тому числі й самостійного, і за допомогою служби зайнятості. У вузькому розумінні – це діяльність державних органів, органів місцевого самоврядування, громадських організацій, суб'єктів підприємницької діяльності, яка спрямована на підшукування роботи особам, що її шукають, включаючи і процес професійної підготовки та перенавчання [3].

Інструменти, за допомогою яких ви можете вивести процес працевлаштування на новий рівень, стають особливо актуальними – через автоматизацію обміну повідомленнями, визначення відповідних компаній відповідно до особливих вимог, посилення на відкриті бази даних, такі як

Crunchbase, і використання можливостей штучного інтелекту для створення ефективної текстової допомоги.

## 1.2 Пошук роботи за допомогою ШІ

Автоматизація у 21-му столітті є головною тенденцією не лише для виробництва, але навіть для комунікацій, найму та особистого виступу. Завдяки прискореному прогресу в області штучного інтелекту, особливо моделі, розробленій навколо OpenAI, люди тепер можуть писати індивідуальний і гнучкий контент, такий як супровідні листи, повідомлення, профілі опису тощо. Тепер із цього випливає, як само штучний інтелект допомагає в пошуку роботи.

По-перше, штучний інтелект допомагає автоматизувати рутину. Описати функціонал своєї позиції, створити дизайн резюме та навіть заповнити його – ШІ може зробити це буквально за кілька секунд. Залишається відредагувати отримані результати та відправити резюме в компанію.

По-друге, штучний інтелект може виділити та розписати навички, якими ви, ймовірно, володієте, на основі ваших попередніх ролей або якщо ви зазначите йому бажану позицію.

По-третє, ШІ можна використовувати, щоб відрефлексувати думки, в яких ви не впевнені. Наприклад, ви можете описати, що саме вам подобається робити, та попросити штучний інтелект підібрати вам найрелевантніші вакансії [4].

Є такі сайти, як Crunchbase, які надають відкритий доступ до інформації про стартапи, венчурні фонди та технологічні компанії. Інтерфейс із даним типом бази даних дозволяє створити цільову вибірку компаній, яка відповідає потребам користувача – галузі, географії, рівню фінансування, стадії розвитку тощо.

Технічно цю концепцію можливо реалізувати за допомогою стека технологій: Python для реалізації логіки аналізатора та обробки даних, TypeScript + ReactTS для клієнтської частини інтерфейсу, хмарної інфраструктури Azure для розміщення серверних елементів, бази даних для зберігання інформації про користувачів і компанії та Figma для розробки зручного та інтуїтивно зрозумілого UX/UI.

Таким чином, поєднання сучасних технологій з відкритими даними та штучним інтелектом закладає основу для розробки інноваційної платформи, яка не тільки спростить пошук роботи, але й зробить його більш ефективним і доступним для широкого кола користувачів.

### 1.3 Цифрова B2B-комунікації для інвесторів і компаній

У сучасних умовах розвитку технологій цифрова B2B-комунікація перетворилася на стратегічно важливий елемент інноваційного бізнесу. Саме взаємодія між компаніями та інвесторами все більше переходить у площину автоматизованих, даних-орієнтованих систем, де критичне значення має не лише швидкість, а й якість інформаційного обміну .

Традиційні способи налагодження контактів, такі як ручний пошук партнерів, розсилка типових листів або нетворкінг на профільних заходах, поступово поступаються місцем інтегрованим цифровим платформам. У світі, де темп технологічних і бізнес-змін постійно зростає, ефективність первинного контакту визначає не лише долю окремої угоди, а й конкурентоздатність усієї компанії чи стартапу на глобальному ринку.

Проблема полягає в тому, що обсяг відкритих даних про компанії й фонди, динаміка оновлення профілів та збільшення кількості гравців роблять класичні комунікаційні інструменти недостатньо ефективними. У результаті учасники ринку стикаються з парадоксом: технологічні рішення для пошуку партнерів існують, але більшість інструментів або не

враховують сучасну специфіку B2B-взаємодії, або недостатньо масштабовані для автоматизації тисяч індивідуальних запитів.

Цифрові B2B-комунікації сьогодні, це вже не просто email або CRM, а багаторівневі екосистеми, які поєднують аналітику, машинне навчання, автоматизований парсинг даних, динамічну генерацію контенту, інтеграцію з соціальними мережами й хмарними сервісами. Найуспішніші рішення побудовані на принципах Data-Driven Outreach, коли кожна точка контакту базується не на гіпотезах чи ручній роботі, а на ретельно оброблених даних про поточний стан ринку, потреби адресата, історію його взаємодії та навіть зміни в його профілі, зафіксовані через API або веб-парсинг [5].

В контексті венчурних інвестицій і стартап-екосистеми цифрова B2B-комунікація дозволяє інвесторам швидко отримувати релевантну інформацію про потенційні об'єкти інвестування, відслідковувати тренди, формувати цільові вибірки і запускати масштабовані outreach-кампанії з індивідуальними сценаріями для кожного учасника. З іншого боку, стартапи та компанії, що шукають партнерів чи клієнтів, мають можливість не тільки швидко презентувати себе тисячам релевантних контактів, а й отримувати зворотний зв'язок, аналізувати поведінку аудиторії, будувати динамічні воронки комунікацій та управляти подальшими кроками в режимі реального часу.

Унікальність сучасної цифрової B2B-комунікації полягає в її комплексності. Вона охоплює всі етапи: від автоматизованого пошуку, валідації та enrichment даних про компанії, включаючи фінансові метрики, інвестиційні раунди, галузі, географію, динаміку розвитку, до персоналізованої генерації листів, багатоканального відправлення, аналізу відповідей, сегментації аудиторії та повторної оптимізації наступних контактів. Усе це відбувається на тлі постійної конкуренції за увагу адресатів, де важливу роль відіграє навіть дрібна деталь у стилі комунікації, часовий слот розсилки, тон повідомлення чи релевантність пропозиції [6].

З'являється новий клас рішень, який базується на штучному інтелекті, великих мовних моделях і аналітиці великих даних. Такі системи здатні самостійно аналізувати профілі компаній, пропонувати гіперперсоналізовані сценарії взаємодії, генерувати cold та warm emails із гнучким налаштуванням тону та стилю, аналізувати відповіді та навіть виявляти потенційні «болі» чи можливості для партнерства на основі відкритих або приватних даних.

Покажемо є те, що цифрові B2B-комунікації, зокрема для інвесторів і компаній, не обмежуються лише автоматизацією — вони вимагають побудови довіри, формування стійких зв'язків і миттєвої адаптації до динаміки ринку. Саме тому найефективніші рішення об'єднують гнучкість інтерфейсів, аналітику поведінки, можливості інтеграції зі сторонніми платформами (Crunchbase, LinkedIn, AngelList, PitchBook тощо) та багатоканальний підхід до комунікації, включаючи email, соціальні мережі, корпоративні месенджери й навіть автоматизовані дзвінки.

Така еволюція цифрових B2B-комунікацій змінює правила гри для всіх учасників ринку. У фокусі опиняється швидкість реакції, персоналізація, автоматичне навчання на даних та мінімізація людського фактору в процесах, які ще кілька років тому вважалися ручними і невіддатливими для повної автоматизації. Ринок все чіткіше рухається у бік розумних платформ, які не тільки забезпечують ефективний зв'язок між інвесторами та компаніями, а й дозволяють масштабувати бізнес без збільшення операційних витрат, підвищувати якість перших контактів і автоматично відслідковувати ефективність кожної кампанії.

#### 1.4 Сучасні підходи до агрегування даних про компанії

В основі сучасних підходів до агрегування лежить ідея створення централізованого профілю компанії, своєрідного цифрового дос'є, що охоплює фінансові метрики, кадровий склад, історію інвестицій,

технологічний стек, медіаактивність, згадки у новинах і навіть реакції на зовнішні ринкові зміни. Зростання обсягів і розмаїття даних робить неможливим ручний моніторинг, тому на перший план виходять автоматизовані системи збору та обробки інформації.

Головним джерелом структурованих даних про стартапи й технологічні компанії виступає Crunchbase – платформа, яка стала галузевим стандартом завдяки масштабованості, гнучкості API та підтримці регулярного оновлення інформації. Її відкриті та платні дані охоплюють понад мільйон компаній, інвесторів і фондів, а можливість фільтрації за галузями, розміром, географією й стадією розвитку дозволяє гнучко формувати цільові списки для досліджень, продажу чи інвестицій.

Водночас, самих сирих даних уже недостатньо: сучасні проекти використовують багаторівневий enrichment профілів, інтегруючи Crunchbase із LinkedIn, AngelList, PitchBook, CB Insights та власними корпоративними БД. Автоматизовані парсери й ETL-процеси синхронізують, нормалізують і звіряють інформацію, мінімізуючи дублювання й помилки, що неминучі при одноразових ручних завантаженнях. Найбільш прогресивні компанії поєднують використання Crunchbase API, web scraping для збору специфічної інформації, автоматичну валідацію контактів через email-верифікатори й AI-based enrichment для відновлення або уточнення даних.

Підвищення вимог до якості даних породило тренд на багатоступеневу валідацію: первинний збір, це лише початок. Далі дані проходять автоматичну перевірку на коректність (використання регулярних виразів, пошук аномалій), enrichment з відкритих джерел (зокрема, автоматичне визначення галузі, знаходження контактів керівників, відстеження змін у соціальних мережах компанії), а також позначення свіжості інформації через timestamp-метки. Лише після цього формується фінальний профіль, придатний для AI-аналізу чи генерації персоналізованих запитів [7].

Суттєву роль у сучасних підходах до агрегування відіграють так звані no-code та low-code платформи, які дозволяють організувати складні багатоступеневі pipeline-и без глибоких знань програмування. Завдяки інтеграції з Crunchbase, Zapier, Integromat (Make) та кастомним API, навіть невеликі команди можуть будувати власні системи збору, enrichment і розсилки, автоматизуючи весь цикл від збору до першого контакту й аналітики відповіді (рисунок 1.2).

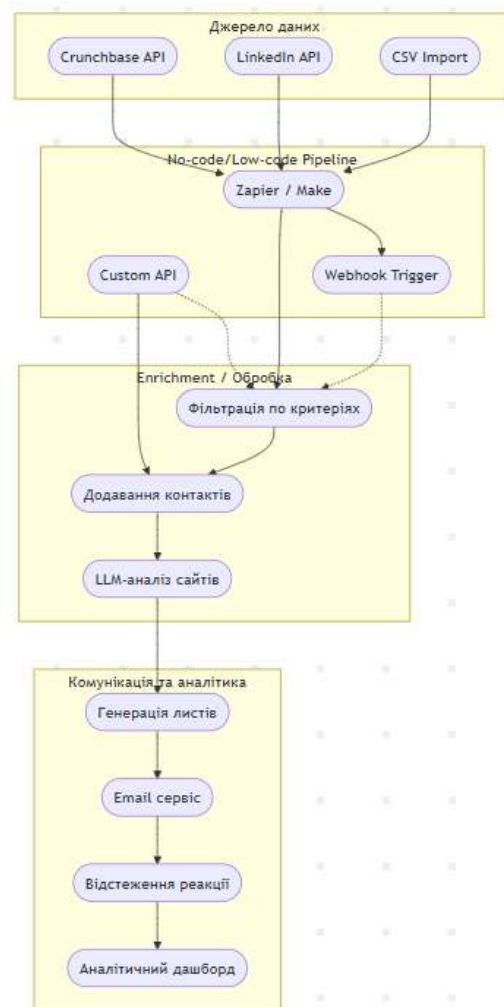


Рисунок 1.2 – Автоматизований pipeline агрегування, enrichment і комунікації на основі no-code/low-code платформ

Із появою LLM з'явилася можливість ще глибшого enrichment даних: автоматичне генерування резюме компанії, класифікація за новими

ознаками, генерація запитань до потенційних партнерів або навіть прогнозування шансів на успішну угоду на основі історичних патернів. LLM також здатні автоматично знаходити релевантні додаткові джерела інформації для поглиблення профілю.

Інноваційним стає й підхід до агрегування контактних даних: автоматизований пошук email-адрес керівників або відділів, валідація доменів, інтеграція з корпоративними скриньками через IMAP або SMTP, автоматичне вилучення контактів із підписів, парсинг LinkedIn та інших соцмереж. Усе це дозволяє формувати максимально “живу” і актуальну базу для подальших персоналізованих комунікацій.

Актуальним викликом залишається також дотримання вимог до приватності та безпеки даних: GDPR, CCPA, локальні закони про персональні дані вимагають вбудованих механізмів анонімізації, контролю доступу, логування змін і обмеження часу зберігання. Тому сучасні системи агрегування все частіше використовують хмарні сховища з granular access control, криптографічний захист інформації, а також автоматичний аудит доступу до даних.

### 1.5 Автоматизація персоналізованих запитів

Персоналізація у сучасній B2B-комунікації вже давно перетворилася на реальний стандарт ефективної взаємодії між компаніями. Саме якісний індивідуальний підхід визначає результативність контактів, рівень довіри, ймовірність відповіді і подальшу співпрацю. Проте в умовах динамічного ринку і зростання обсягів даних ручна персоналізація стає неможливою навіть для невеликих команд, не кажучи вже про великі компанії, які щоденно контактують із сотнями або тисячами потенційних партнерів.

Автоматизація цього процесу стає основною рушійною силою підвищення конверсії у B2B-продажах, венчурних інвестиціях, рекрутингу та партнерському пошуку. Завдяки впровадженню сучасних технологій

з'явилася можливість поєднувати масштабність із справжньою індивідуальністю кожного звернення. Класичні інструменти масових розсилок більше не справляються із вимогами ринку, оскільки часто пропонують лише базову підстановку імені, компанії або галузі, але не враховують реальних інтересів і контексту адресата.

Інтеграція великих мовних моделей, API сучасних платформ і автоматичних систем enrichment дозволяє генерувати унікальні звернення для кожного контакту на основі десятків параметрів. До уваги береться не лише профіль компанії чи її поточна діяльність, а й історія фінансування, особливості ринку, останні публікації у ЗМІ, поведінка у соціальних мережах і навіть стиль внутрішньої комунікації. Такі системи автоматично формують релевантний зміст повідомлення, адаптують тон, структуру, підбирають оптимальний момент для відправлення та відстежують реакцію з боку адресата [8].

Автоматизовані платформи вже на етапі формування списку контактів використовують алгоритми, які виявляють найбільш перспективні цілі, ранжують їх за ймовірністю позитивної відповіді, а також збагачують профілі додатковою інформацією. Далі персоналізовані запити генеруються не за шаблоном, а у вигляді оригінального тексту, що враховує специфіку ситуації, і одразу ж відправляються через вибраний канал: email, месенджер або інтегрований корпоративний чат. На цьому цикл не закінчується: після отримання відповіді система автоматично аналізує зміст листування, пропонує оптимальні наступні кроки та коригує сценарій подальшої взаємодії.

Такий підхід дає змогу одночасно спілкуватися з тисячами адресатів, при цьому кожен із них відчуває, що звернення дійсно написано саме для нього. Саме інтелектуальні моделі і багаторівневий enrichment створюють якісно новий рівень персоналізації, що підвищує шанси на успішний контакт і дозволяє вибудовувати довгострокові партнерські відносини. Водночас автоматизація персоналізованих запитів дозволяє мінімізувати

людський фактор, уникати помилок, оптимізувати ресурси і витратити час лише на ті контакти, які справді мають значення.

### 1.6 Роль AI/LLM у генерації релевантних комунікацій

Основний прорив, який приносять LLM, полягає у здатності розуміти контекст і генерувати тексти, які не відрізняються від людських, за змістом, стилем і навіть емоційною насиченістю. Якщо раніше автоматизовані системи обмежувалися простим підбором шаблонів і автозаміною базових даних, сьогодні LLM можуть працювати із складними сценаріями, де враховуються десятки факторів, професійна біографія адресата, історія попередніх комунікацій, новини ринку, інвестиційні тенденції, особливості бізнесу компанії, цілі і болі цільової аудиторії.

LLM здатні аналізувати вхідні дані про компанію, підбирати найбільш релевантний стиль комунікації та адаптувати зміст звернення до очікувань одержувача. Для цього використовуються складні prompt-сценарії — тобто спеціально структуровані інструкції, які задають логіку, тон і бажану глибину тексту. Система може автоматично згенерувати короткий cold email для першого контакту з інвестором, докладний аналітичний лист із запрошенням на зустріч, або ж шаблон follow-up для повторного звернення. При цьому в кожному випадку враховується історія контактів, тип компанії, обсяг фінансування, актуальні новини чи навіть стиль листування, властивий конкретному decision-maker [9].

Завдяки цьому процес створення релевантних комунікацій стає динамічним і гнучким. Система може автоматично реагувати на зміни в даних: якщо в профілі компанії оновилися інформація про фінансування чи з'явилася новина у ЗМІ, це миттєво відображається в контексті майбутніх звернень. Штучний інтелект адаптує тематику листа, пропозицію, навіть можливі сценарії подальшої співпраці. Такі можливості вкрай важливі у

венчурних і технологічних сферах, де оперативність і точність комунікації прямо впливають на шанси укладання угоди.

LLM також значно підвищують якість персоналізації. Якщо класичні інструменти підстановки обмежувалися змінними на кшталт імені, посади чи галузі, сучасні AI-системи працюють із комплексною багатовимірною моделлю профілю адресата. Наприклад, якщо компанія тільки що отримала новий раунд інвестицій, штучний інтелект може сформулювати привітання з урахуванням цієї події, підкреслити релевантність власної пропозиції для поточного етапу розвитку або наголосити на спільних інтересах. Це дозволяє створювати справді живу і вчасну комунікацію, яка сприймається як унікальна й небанальна.

Окрім генерації листів, LLM застосовуються для аналізу вхідної кореспонденції, автоматичного розпізнавання тональності відповіді, витягу релевантних даних та класифікації запитів. Система може самостійно аналізувати, чи варто продовжувати комунікацію, запропонувати оптимальний час для наступного звернення, згенерувати відповідь на стандартні запитання або навіть автоматично оновити CRM-дані за результатами листування.

Важливою перевагою таких моделей є їхня здатність навчатися на нових даних і адаптуватися під специфіку конкретної компанії або індустрії. Власні тренування на масивах бізнес-листування, документації, угод чи соціальних мереж дозволяють LLM краще розуміти професійну лексику, вловлювати неочевидні зв'язки, навіть виявляти ризики та пропонувати альтернативні сценарії взаємодії [10].

На рівні архітектури, сучасні системи автоматизованої комунікації реалізують повний цикл: від збору даних із відкритих джерел і enrichment профілю, через генерацію та аналіз контенту, до аналітики реакцій і прогнозування подальших кроків. Усе це працює в режимі реального часу, часто із застосуванням хмарних сервісів, що дає змогу масштабувати рішення під будь-який обсяг контактів.

Роль AI/LLM у сучасних платформах полягає не лише в генерації тексту, а й у створенні цінності для бізнесу. Підвищується конверсія відповідей, скорочується час на підготовку звернень, оптимізується ресурс команди, зменшуються ризики втрати критично важливих контактів. Особливо важливо, що рівень персоналізації і динамічної адаптації значно підвищує довіру до компанії-відправника, дозволяє формувати стійкі партнерські зв'язки і відрізнятись на тлі конкурентів.

У найближчій перспективі LLM поступово переходять від простого генератора тексту до справжнього цифрового асистента, який супроводжує всі етапи взаємодії: від дослідження ринку до укладання угоди. Уже сьогодні такі системи підтримують автоматичну обробку сотень тисяч запитів щодня, допомагають масштабувати бізнес із мінімальними витратами та забезпечують аналітику, яка раніше була доступна лише в ручному режимі.

## 2 МЕТОДИ, ТЕХНОЛОГІЇ ТА ПІДХОДИ

### 2.1 Теоретичні аспекти обраних підходів

Велика мовна модель (Large Language Model, LLM) – це сучасний підхід до розробки та застосування мовних моделей на основі штучного інтелекту, який використовує глибоке навчання. Великі мовні моделі зазвичай побудовані на основі рекурентних або трансформерних архітектур і тренуються на величезних наборах текстових даних.

Однією з найбільш відомих та впливових великих мовних моделей є GPT (Generative Pre-trained Transformer), розроблена компанією OpenAI. GPT використовує трансформерну архітектуру та тренується на широкому спектрі інтернет-даних, що дозволяє йому генерувати текст та виконувати завдання, пов'язані з мовою, з вражаючою ефективністю та різноманітністю.

Великі мовні моделі мають потужний потенціал у таких областях, як машинне навчання, обробка природної мови, автоматичне перекладання, генерація тексту та розуміння контексту. Проте вони також можуть породжувати етичні питання, пов'язані з використанням, а також вимагати великих обчислювальних ресурсів для навчання та використання [6].

В основі роботи LLM лежить прогнозування наступного слова (токена) на основі заданого контексту. Це досягається завдяки попередньому навчання моделі на величезних корпусах тексту з різних джерел (веб-сайти, книги, наукові статті тощо).

Однією з ключових переваг LLM є здатність до генерації змістовних, граматично коректних і стилістично адаптованих текстів. Для досягнення високої якості тексту велике значення має техніка prompt engineering – формулювання запиту до моделі у такий спосіб, щоб максимізувати релевантність відповіді.

У межах цього проєкту LLM використовується для автоматичного створення персоналізованих повідомлень (cold email і warm email), адаптованих до сфери діяльності отримувача та профілю користувача.

Парсинг (scraping) – це процес автоматизованого збору структурованих даних із веб-сторінок. Для ефективного витягу даних із платформ, таких як Crunchbase, застосовуються такі підходи:

- HTML-парсинг з використанням бібліотек, які дозволяють витягувати DOM-елементи на основі CSS-селекторів (наприклад, BeautifulSoup);
- кероване веб-скрапінг середовище з використанням Selenium або Playwright для взаємодії з JavaScript-динамічним контентом;
- доступ до API, якщо платформа його надає, для надійного доступу до структурованих даних без прямої роботи з HTML.

Для уникнення блокувань, парсери мають реалізовувати механізми чергування User-Agent, затримок між запитами, обробки cookies та помилок.

Зібрані дані проходять попередню очистку, нормалізацію та фільтрацію за заздалегідь визначеними критеріями (індустрія, географія, стадія компанії тощо).

Після збору даних наступним кроком є їх структуризація та підготовка до використання в системі. У межах проєкту застосовуються:

- ORM-підходи для взаємодії з базою даних (наприклад, SQLAlchemy);
- фільтрація та агрегація за ключовими ознаками для формування релевантних вибірок;
- оптимізація запитів до БД для підвищення продуктивності та масштабованості системи.

Також важливою є відповідність зібраних і збережених даних правовим нормам (наприклад, GDPR), з чим пов'язані механізми анонімізації або обмеження зберігання персональної інформації.

## 2.2 Визначення функціональних вимог

Мета проекту – створити веб-платформу, яка автоматично синхронізує процес пошуку контактів між стартапами та інвестиційними компаніями, а також надає користувачам можливість зв'язуватися з ними через штучний інтелект. Система повинна забезпечувати простий і швидкий спосіб пошуку потенційних роботодавців або партнерів шляхом аналізу бази даних Crunchbase, створення персоналізованих електронних листів і відстеження їх статусу в режимі реального часу.

Crunchbase – онлайн-платформа з інформацією про стартапи, що належить TechCrunch, технологічному виданню з США. Зараз на сторінці компанії в Crunchbase можна побачити її назву, галузь, статус IPO, номер телефону, сайт, акаунти в соцмережах і короткий огляд бізнесу [11].

Мінімальні операційні потреби платформи:

- дані зібрані з Crunchbase. Платформа повинна використовувати внутрішньо-платформенний парсер, який збирає звичайні структуровані дані про компанії, стартапи та венчурні фонди з Crunchbase. Відфільтровані дані з'являються на платформі за попередньо визначеними критеріями (індустрія, локація, стадія розвитку компанії тощо);

- профіль користувача. Кожен користувач може створити обліковий запис, подати дані для профілю, додати опис про себе та про проекти;

- генерація листів. Система генерує два типи повідомлень електронної пошти за допомогою OpenAI API: перше – коротке вступне повідомлення, cold email, для привернення уваги, друге – довший лист, warm email, із презентацією користувача та його здібностей.

## 2.3 Визначення технічних та програмних ресурсів

Для реалізації платформи необхідно було обрати сучасний технологічний стек, який би забезпечував ефективну взаємодію між базою

даних, системою генерації листів та інтерфейсом. Розробка використовує такі ресурси:

- Python – це мова, яка буде використовуватися як основний інструмент для побудови парсера даних з Crunchbase. Python дозволяє працювати з API, обробляти велику кількість структурованих даних та взаємодіяти з базами даних через бібліотеки ORM. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ [12];

- OpenAI API – це сервіс генерації тексту на основі штучного інтелекту. Він буде використовуватися для автоматичного створення електронних листів (холодних та теплих) з метою значного підвищення релевантності комунікації з потенційними роботодавцями або інвесторами. Для отримання максимальної якості тексту слід налаштувати відповідні запити API;

- Firebase або інша хмарна база даних для безпечного зберігання даних користувачів, даних компанії, журналів відправлення електронних листів та статусу відповідей. Хмарне рішення забезпечить високу доступність даних, швидку масштабованість та захист інформації відповідно до політик GDPR;

- React з TypeScript для розробки динамічного інтерфейсу користувача наступного покоління. React дозволяє створити адаптивний фронтенд зі швидким оновленням даних у режимі реального часу, а TypeScript забезпечує перевірку типів та зменшить кількість помилок під час процесу розробки;

- Azure для розміщення бекенду, керування додатком та масштабування інфраструктури при додаванні більшої кількості

користувачів. Хмари також дозволяють інтегрувати системи моніторингу продуктивності, системи сповіщень та автоматичне розгортання оновлень;

- Figma для розробки та проектування макетів інтерактивних платформ. Попередній перегляд інтерфейсу на предмет зручності використання за допомогою Figma дозволяє заощадити час на перевірці пізніше, скоординувати дизайн між членами команди та уникнути великої кількості помилок у подальшому впровадженні.

Окрім основних інструментів, також знадобилося:

- Netlify для швидкого розгортання фронтенду;
- Node.js (якщо створюється окремий бекенд) для проектування API, який забезпечить безпечну передачу даних між сервером і клієнтом.

Таким чином, правильний вибір та поєднання інструментів забезпечив масштабованість, безпеку та продуктивність під час обробки великих обсягів інформації, а також створив сучасний продукт, який відповідає високим очікуванням користувачів.

## 3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ

### 3.1 Розробка парсера для отримання даних із Crunchbase

На початковому етапі розробки бекенд-частини платформи було створено парсер для обробки та вилучення структурованої інформації про стартапи та компанії з відкритих джерел Crunchbase.

Основною метою даного модуля є автоматизація збору галузевих даних (industry), а також додаткових параметрів, що характеризують компанії-профілі.

Архітектура реалізована на базі Python з використанням таких технологій:

- FastAPI – як основний фреймворк для створення REST API;
- Selenium – для автоматизованого скрапінгу інформації зі сторінок Crunchbase;
- PostgreSQL – як СУБД;
- SQLAlchemy – для ORM-інтеграції з базою даних;
- CSV/JSON – формати імпорту/експорту та взаємодії з фронтендом;
- Docker – для контейнеризації та розгортання парсера;
- dotenv – для безпечного керування конфігураціями.

Вхідними даними для парсингу є CSV-файл, що містить унікальний ідентифікатор компанії та її URL у системі Crunchbase. Обробка здійснюється за допомогою сервісу WebParserSelenium, який працює у безголовому режимі Chrome через Selenium Grid.

Парсер очікує завантаження елементів DOM, після чого витягує теги з класом .chip-text, що містять інформацію про індустрії, функціонал якої можна побачити у лістингу 3.1. Він очікує наявності необхідного контенту в DOM, що забезпечує стабільну роботу навіть при повільному завантаженні сторінки.

Лістинг 3.1 – Програмний код функціоналу збору індустріальної інформації з кількома спробами.

```
def fetch_data(self, url: str) -> list:
    attempt = 0
    max_attempts = 3
    industries = []
    while attempt < max_attempts:
        ...
        WebDriverWait(driver, 30).until(
            EC.presence_of_element_located((By.CSS_SELECTOR, ".chip-
            text")))
        )
        ...
        chip_texts = chips_container.find_all("div",
        class_="chip-text")
        industries = [chip.text.strip() for chip in
        chip_texts]
```

Оброблені дані фільтруються та експортуються у формат CSV. Результати зберігаються у базі даних, де передбачено зв'язок між таблицями WebsiteInfo та CompanyIndustry за допомогою проміжної таблиці file\_loader\_websiteinfo\_industry. Крім того, реалізована валідація UUID та автоматичне додавання нових галузей у відповідну таблицю, якщо такі ще не існують.

Для організації доступу до парсера створено набір ендпоінтів, про які буде описуватись в наступних розділах, зокрема:

- /get\_file\_purpose\_from\_db/ – отримання типів файлів;
- /get\_file\_upload\_from\_db/ – історія завантажень;
- /parse\_industry/ – основна точка обробки CSV-файлу та запуск Selenium-парсера.

База даних регулярно оновлюється з періодичністю один раз на місяць. Такий підхід забезпечує підтримку актуального переліку індустрій і дозволяє формувати точні та персоналізовані запити до потенційних партнерів на основі найсвіжішої інформації з Crunchbase.

### 3.2 Хмарна реалізація механізму збору та оновлення даних

У межах реалізації серверної частини платформи особливу увагу було приділено розгортанню інфраструктури в хмарному середовищі Microsoft Azure. Хмарні служби Azure – це приклад платформи як послуги (PaaS). Як і служба додатків Azure, ця технологія розроблена для підтримки масштабованих, надійних та недорогих в експлуатації програм. Так само, як служба додатків розміщується на віртуальних машинах (VM), так само і хмарні служби Azure. Однак є можливість більше контролю над VM. Можна встановлювати власне програмне забезпечення на VM, які використовують хмарні служби Azure, і отримувати до них віддалений доступ.

Застосунок хмарних служб Azure зазвичай стає доступним для користувачів через двоетапний процес. Розробник спочатку завантажує програму в область проміжної версії платформи. Коли розробник готовий запустити програму, він використовує портал Azure для заміни проміжної версії на робочу. Це перемикання між проміжною та робочою версією можна виконати без простоїв, що дозволяє оновити запущену програму до нової версії, не турбуючи її користувачів.

Хмарні служби Azure також забезпечують моніторинг. Як і віртуальні машини, вони виявляють несправний фізичний сервер і перезапускають VM, які працювали на цьому сервері, на новій машині. Але хмарні служби Azure також виявляють несправні VM і програми, а не лише апаратні збої. На відміну від віртуальних машин, він має агента всередині кожної веб-ролі та робочої ролі, тому він може запускати нові віртуальні машини та екземпляри програм у разі виникнення збоїв [9].

Особливий акцент було зроблено на використанні Azure AI, зокрема, для генерації унікальних холодних email-повідомлень (cold emails). Використовуючи сервіс Azure OpenAI, для кожної компанії на основі її опису, географії, галузі діяльності та інших даних з Crunchbase формувалося персоналізоване повідомлення, яке можна використовувати в email-кампаніях. Це дозволяло автоматизувати масштабне залучення потенційних клієнтів або інвесторів з мінімальним ручним втручанням.

Azure AI забезпечив необхідний рівень масштабованості, дозволяючи одночасно генерувати сотні унікальних email-шаблонів. Вбудована підтримка API, хороша документація та інтеграція з іншими хмарними сервісами Azure зробили його ідеальним вибором для задач, де важлива швидка обробка текстових даних, генерація природного тексту та відповідність безпековим вимогам корпоративного середовища.

Основною базою даних проєкту виступає PostgreSQL, яка була розгорнута як керована служба в Azure за допомогою сервісу Azure Database for PostgreSQL. Це дозволило уникнути ручної конфігурації БД, а також отримати вбудовану підтримку резервного копіювання, моніторингу, автоматичного масштабування та шифрування даних.

Для зберігання конфіденційних змінних середовища, таких як ключі доступу до бази даних, токени OpenAI, шляхи до ресурсів і параметри підключення до зовнішніх API, використовувався Azure Key Vault. Цей сервіс гарантує безпечне зберігання чутливої інформації та спрощує її використання у процесі розгортання. Параметри середовища, які не потребують особливого захисту, задавалися безпосередньо в конфігурації обраного хостингу (наприклад, у змінних середовища App Service).

Кожне оновлення коду запускало відповідний pipeline, який збирає Docker-образи, пушить їх у ACR, після чого виконує деплой за допомогою CLI або ARM/Вісер-шаблонів. Це забезпечує безперервну інтеграцію та доставку змін до робочого середовища без участі розробника в рутинних операціях.

Ще одним важливим етапом у побудові інфраструктури стало забезпечення регулярного оновлення бази компаній із Crunchbase. Для цього було розроблено автоматичний механізм оновлення, що працює на основі таймерів – зокрема, через Azure Functions або Azure Logic Apps. Один раз на місяць ці сервіси викликають спеціальний ендпоінт у FastAPI-додатку, який ініціює повторний запуск парсера, оновлює дані про компанії та індустрії, і записує результат у базу даних.

Таким чином, інтеграція з Azure надала змогу побудувати гнучку, масштабовану та безпечну серверну інфраструктуру для обробки великих обсягів даних, що періодично оновлюються, з подальшою інтеграцією в бізнес-логіку та фронтенд-платформу.

### 3.3 Фільтрація даних та IMAP протокол

Після первинного збору даних із Crunchbase постало завдання їх подальшої обробки: фільтрації за роллю компаній, очищення, а також розширення на основі зовнішніх джерел. Основна мета – зробити сирі дані більш придатними для бізнес-аналітики, email-маркетингу та персоналізованої взаємодії з потенційними клієнтами або партнерами.

Першим етапом було виділення релевантних компаній із великого CSV-файлу `organizations.csv` (рисунок 3.1).

Простий фільтр на основі значення у колонці `primary_role` дозволив сформувати підвибірку, наприклад, тільки інвесторів – їх було збережено у файл `investors.csv`. Надалі саме з цим сегментом і проводилася поглиблена обробка.

Наступним кроком стало розширення даних за допомогою автоматичного збору контактних email-адрес із сайтів компаній. Для цього було реалізовано два підходи: через зовнішній інструмент `email_extractor`, що запускався за допомогою Python-модуля `subprocess`, та через кастомний багатопотоковий/багатопроецесний фреймворк `WebsiteProcessor`,

побудований на бібліотеці `extract_emails`. Функціонал парсингу імейлів можна побачити у лістингу 3.2.

### Лістинг 3.2 – Програмний код функціоналу парсеру імейлів

```
class WebsiteProcessor(BaseFastProcessor):
    def __init__(self, websites, **kwargs):
        super().__init__(websites, **kwargs)
        self.browser = kwargs.get("browser",
RequestsBrowser)()

    def worker_function(self, website):
        factory = Factory(
            website_url=website,
            browser=self.browser,
            depth=5,
            max_links_from_page=1
        )
        worker = DefaultWorker(factory)
        return worker.get_data()
```

uid	name	type	primary_role	cb_url	domain	homepage_url	logo_url	facebook_url	twitter_url	linkedin_url	combined_stock_symbols	city	region	country_code	short_description
3454	Hutchison	organs	investor	https://hutchco	https://www.hutch	https://www.hutch	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/hutchison	https://twitter.com/hutchison	https://www.linkedin.com/company/hutchison		Hong Kong	HK	HKG	Hutchison Whampoa Li
56904	Cheung Kc	organs	investor	https://www.ckh.com	https://www.ckh.com	https://www.ckh.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/ckh	https://twitter.com/ckh	https://www.linkedin.com/company/ckh		Hong Kong	HK	HKG	Cheung Kong Holdings
20263	Goldman S	organs	investor	https://www.goldman.com	https://www.goldman.com	https://www.goldman.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/goldmansachs	https://twitter.com/goldmansachs	https://www.linkedin.com/company/goldmansachs		New York	NY	USA	Goldman Sachs is a mu
40394	Tapuz	organs	investor	https://www.tapuz.co.il	https://www.tapuz.co.il	https://www.tapuz.co.il	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/tapuz	https://twitter.com/tapuz	https://www.linkedin.com/company/tapuz		Herzliya	Aviv	ISR	Tapuz is a financial orga
40471	Legg Mason	organs	investor	https://www.leggmason.com	https://www.leggmason.com	https://www.leggmason.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/leggmason	https://twitter.com/leggmason	https://www.linkedin.com/company/leggmason		Baltimore	Maryland	USA	Legg Mason is an Amer
30604	3i Group	organs	investor	https://www.3i.com	https://www.3i.com	https://www.3i.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/3igroup	https://twitter.com/3igroup	https://www.linkedin.com/company/3igroup		London	England	GBR	3i is an international inv
11364	Allen & Cox	organs	investor	https://www.allenandcox.com	https://www.allenandcox.com	https://www.allenandcox.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/allenandcox	https://twitter.com/allenandcox	https://www.linkedin.com/company/allenandcox		Lakeland	Florida	USA	Allen & Company focus
50224	Garage Tech	organs	investor	https://www.garagetech.com	https://www.garagetech.com	https://www.garagetech.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/garagetech	https://twitter.com/garagetech	https://www.linkedin.com/company/garagetech		Palo Alto	California	USA	Garage Technology Ven
46884	Citi	organs	investor	https://www.citi.com	https://www.citi.com	https://www.citi.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/citi	https://twitter.com/citi	https://www.linkedin.com/company/citi		New York	NY	USA	Citigroup is a financial s
40254	SoftBank C	organs	investor	https://www.softbank.com	https://www.softbank.com	https://www.softbank.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/softbank	https://twitter.com/softbank	https://www.linkedin.com/company/softbank		Tokyo	Tokyo	JPN	SoftBank Group is a hol
40784	GE Comm	organs	investor	https://www.ge.com	https://www.ge.com	https://www.ge.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/ge	https://twitter.com/ge	https://www.linkedin.com/company/ge		Norfolk	Connecticut	USA	GE Commercial Finance
50464	Shea Vent	organs	investor	https://www.sheaventures.com	https://www.sheaventures.com	https://www.sheaventures.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/sheaventures	https://twitter.com/sheaventures	https://www.linkedin.com/company/sheaventures		Walton	California	USA	Shea Ventures provide
30444	Global Cat	organs	investor	https://www.globalcatalyst.com	https://www.globalcatalyst.com	https://www.globalcatalyst.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/globalcatalyst	https://twitter.com/globalcatalyst	https://www.linkedin.com/company/globalcatalyst		Redwood City	California	USA	Global Catalyst Partne
40404	Anchor	organs	investor	https://www.anchor.com	https://www.anchor.com	https://www.anchor.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/anchor	https://twitter.com/anchor	https://www.linkedin.com/company/anchor		Denver	Colorado	USA	You might say that Den
40404	Herald Ven	organs	investor	https://www.heraldventures.com	https://www.heraldventures.com	https://www.heraldventures.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/heraldventures	https://twitter.com/heraldventures	https://www.linkedin.com/company/heraldventures		London	England	GBR	Herald specialise in sch
30154	Outcome C	organs	investor	https://www.outcomecapital.com	https://www.outcomecapital.com	https://www.outcomecapital.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/outcomecapital	https://twitter.com/outcomecapital	https://www.linkedin.com/company/outcomecapital		Reston	Virginia	USA	Outcome Capital is an i
30094	Seed&Spark	organs	investor	https://www.seedandspark.com	https://www.seedandspark.com	https://www.seedandspark.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/seedandspark	https://twitter.com/seedandspark	https://www.linkedin.com/company/seedandspark		London	England	GBR	Seed&Spark is a Europ
40404	SCP World	organs	investor	https://www.scpworld.com	https://www.scpworld.com	https://www.scpworld.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/scpworld	https://twitter.com/scpworld	https://www.linkedin.com/company/scpworld		New York	NY	USA	SCP Worldwide is a gro
41471	Elron Ven	organs	investor	https://www.elronventures.com	https://www.elronventures.com	https://www.elronventures.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/elronventures	https://twitter.com/elronventures	https://www.linkedin.com/company/elronventures		Tel Aviv	Aviv	ISR	Elron Ventures is an ear
40171	The Kraft G	organs	investor	https://www.kraftgroup.com	https://www.kraftgroup.com	https://www.kraftgroup.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/kraftgroup	https://twitter.com/kraftgroup	https://www.linkedin.com/company/kraftgroup		Foxboro	Massachusetts	USA	The Kraft Group is a pri
50404	eorCapital	organs	investor	https://www.eorcapital.com	https://www.eorcapital.com	https://www.eorcapital.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/eorcapital	https://twitter.com/eorcapital	https://www.linkedin.com/company/eorcapital		Centennial	Colorado	USA	eorCapital is a venture
50104	Francisco P	organs	investor	https://www.franciscopartners.com	https://www.franciscopartners.com	https://www.franciscopartners.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/franciscopartners	https://twitter.com/franciscopartners	https://www.linkedin.com/company/franciscopartners		San Francisco	California	USA	Francisco Partners is a
40794	Ben Frankl	organs	investor	https://www.benfranklin.com	https://www.benfranklin.com	https://www.benfranklin.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/benfranklin	https://twitter.com/benfranklin	https://www.linkedin.com/company/benfranklin		Bethesda	Pennsylvania	USA	The Ben Franklin Techn
24004	Indiana's 2	organs	investor	https://www.indianastatecapital.com	https://www.indianastatecapital.com	https://www.indianastatecapital.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/indianastatecapital	https://twitter.com/indianastatecapital	https://www.linkedin.com/company/indianastatecapital		Indianapolis	Indiana	USA	The Indiana 21st Centur
40404	Deutsche B	organs	investor	https://www.deutschebank.com	https://www.deutschebank.com	https://www.deutschebank.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/deutschebank	https://twitter.com/deutschebank	https://www.linkedin.com/company/deutschebank		Frankfurt	Hessen	DEU	Deutsche Bank, a Frank
17004	Flagship P	organs	investor	https://www.flagshippartners.com	https://www.flagshippartners.com	https://www.flagshippartners.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/flagshippartners	https://twitter.com/flagshippartners	https://www.linkedin.com/company/flagshippartners		Cambridge	Massachusetts	USA	Flagship Pioneering is a
40404	Novesta	organs	investor	https://www.novesta.com	https://www.novesta.com	https://www.novesta.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/novesta	https://twitter.com/novesta	https://www.linkedin.com/company/novesta		Stockholm	Sweden	SWE	Novesta is an independ
40404	The Mussa	organs	investor	https://www.mussa.com	https://www.mussa.com	https://www.mussa.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/mussa	https://twitter.com/mussa	https://www.linkedin.com/company/mussa		Wayne	Pennsylvania	USA	The Mussa Group was
40404	T. Rowe Pr	organs	investor	https://www.troweprice.com	https://www.troweprice.com	https://www.troweprice.com	https://images.crunchbase.com/images/upload/ctb-default-original/1387176475/7	https://www.facebook.com/troweprice	https://twitter.com/troweprice	https://www.linkedin.com/company/troweprice		Baltimore	Maryland	USA	T. Rowe Price is an inv

Рисунок 3.1 – Таблиця organizations.csv

Клас `WebsiteProcessor` розширює базовий фреймворк `BaseFastProcessor`, реалізуючи логіку збору email-адрес з сайтів за допомогою бібліотеки `extract_emails`. Обхід виконується в кілька потоків і процесів, що дозволяє ефективно обробляти великі об'єми сайтів. Функціональний код цієї реалізації можна побачити у лістингу 3.3.

### Лістинг 3.3 – Програмний код функціоналу екстрактору імейлів

```
def extract_emails_from_url(self, homepage_url: str) -> list:
    try:
        subprocess.run(
            [
                "email_extractor",
                f"-out={self.output_file}",
                f"-url={homepage_url}",
            ],
            check=True,
            stdout=subprocess.DEVNULL,
            stderr=subprocess.DEVNULL,
        )
        with open(self.output_file, mode="r", encoding="utf-8") as f:
            emails = f.read().strip().split("\n")
            return [email for email in emails if email]
    except subprocess.CalledProcessError as e:
        print(f"Error processing {homepage_url}: {e}")
        return []
```

Метод `extract_emails_from_url` викликає зовнішню утиліту `email_extractor` через `subprocess.run`. Після завершення роботи результат читається з CSV-файлу. Такий підхід дозволяє інтегрувати сторонні засоби витягування даних без глибокої залежності від їх API або структури.

Обидва методи дозволяли масово сканувати сайти, витягувати звідти електронні адреси (з глибиною обходу до 5 рівнів) і пов'язувати їх із UUID

компаній. Обробка виконувалась паралельно, що значно пришвидшувало процес.

Окрім пошуку email-адрес на сайтах, було також реалізовано інтеграцію з поштовими скриньками за допомогою протоколу IMAP. Це дозволило парсити вхідні email-листи безпосередньо зі скриньок – наприклад, для автоматичного вилучення контактних адрес із листування або перевірки актуальності знайдених контактів. Таке доповнення значно підвищило точність і повноту зібраної контактної бази.

Після збору всі дані проходили етап валідації та очищення. Використовувався клас EmailValidator, який перевіряв правильність форматування email-адрес за допомогою регулярних виразів. Далі, модуль RemoveDuplicatesEmails видаляв дублікати як усередині окремої компанії, так і загалом по всій вибірці. Це дозволяло зберегти лише унікальні, валідні, ймовірно робочі адреси для подальших маркетингових дій.

Загальна архітектура побудови парсерів була спроектована з урахуванням принципів масштабованості: були винесені універсальні міксини для роботи з потоками та процесами, спільна логіка обробки розділена на абстрактні базові класи, що спрощувало модульність і розширення.

Таким чином, фільтрація за галузями, автоматичне збирання контактів через веб та IMAP, а також подальша очистка даних дозволили значно збагачувати первинну інформацію з Crunchbase, перетворюючи її на цінний аналітичний і маркетинговий інструмент.

Після успішного парсингу, було отримані csv-файли, поля таблиць який допоможе при подальшій візуалізації даних та парсингу мейлів, маючи Facebook компанії, які парсяться звідти (рисунок 3.2).

uuid	name	type	primary_role	cb_url	domain	homepage_url	logo_url	facebook	twitter	linkedin	combined_stock_symbols	city	region	country	short_description	
3d9e	Hutchis	organizatio	investor	<a href="https://www.cr">https://www.cr</a>	hutchison	<a href="http://www.hutchi">http://www.hutchi</a>	<a href="https://images.crunchbase.com/image">https://images.crunchbase.com/image</a>	hkg-0013				Hong	Hong	Ki	HKG	Hutchison Whamp

Рисунок 3.2 – Таблиця зі спаршеними даними

### 3.4 Інтеграція штучного інтелекту для генерації листів

Одним із ключових напрямків автоматизації в рамках розробки бекенд-платформи стала генерація холодних електронних листів (cold emails) із використанням технологій штучного інтелекту. Основна мета – забезпечити створення персоналізованих, релевантних і водночас ефективних комунікацій із представниками компаній, інформацію про які було зібрано за допомогою попередніх парсерів.

Для реалізації цього функціоналу було обрано хмарну платформу Azure AI, зокрема – сервіс Azure OpenAI, який надає доступ до сучасних мовних моделей на кшталт GPT-4. Це дозволило безпечно та масштабовано інтегрувати генерацію тексту у серверну частину системи, гарантуючи стабільність, відповідність нормативам безпеки, а також хорошу інтеграцію з іншими хмарними сервісами (наприклад, Azure Functions чи Azure Blob Storage).

Процес створення cold email-листів побудовано навколо шаблонного запиту до великої мовної моделі. В якості вхідних даних використовуються наступні параметри:

- назва компанії;
- галузь, у якій вона працює;
- короткий опис із Crunchbase;
- географічне розташування;
- основна роль (наприклад, стартап, інвестор тощо);
- знайдені email-адреси;
- потенційний мотив звернення (наприклад, інвестиційна пропозиція, партнерство, запрошення на подію тощо).

Ці параметри формують системний prompt, який задає загальний тон повідомлення: неформальний, але професійний стиль, короткий вступ, чітка пропозиція цінності й заклик до дії (наприклад, «давайте поговоримо» або «відповідайте, якщо цікаво»). Модель генерує унікальний текст на основі

цих даних, який потім проходить постобробку: видаляються непотрібні звернення, додається підпис компанії-ініціатора, валідується граматика та структура.

Для генерації релевантного тексту повідомлення штучний інтелект потребує попереднього контексту, в якому окреслюється роль користувача, мета та особливості проєкту. Цей контекст формується за допомогою `system prompt`, функціональний код якого можна побачити у лістингу 3.4.

#### Лістинг 3.4 – Програмний код формування системного промπτу

```
result["system_prompt"] = (
    f"You are a strategically-minded assistant. "
    f"You work for the company {sender.sender_company}. "
    f"Your company specializes in:
{sender.sender_company_profile}. "
    f"Your name is {sender.sender_name}. "
    f"You serve as {sender.sender_position}. "
    f"You present a project:
{sender.sender_project.project_name}. "
    f"Your project description:
{sender.sender_project.project_description}. "
    f"Project key advantages:
{sender.sender_project.key_advantages}. "
    f"Project ROI: {sender.sender_project.projected_roi}. "
    f"Project growth strategy:
{sender.sender_project.growth_strategy}. "
    f"project market potential:
{sender.sender_project.market_potential}. "
    "Your goal is to engage potential investors by providing
concise, compelling information about the project and its
growth potential."
)
```

Цей фрагмент задає роль помічника як представника компанії, який знайомий з основною інформацією про проєкт: його назвою, описом,

перевагами, фінансовим потенціалом та ринковими перспективами. Це допомагає моделі сформуванню узгоджене й контекстуально коректне повідомлення.

Для першого контакту з потенційним партнером використовується так званий `user prompt`, який задає вимоги до стилю листа, його тону та основних акцентів, функціональний код якого можна побачити у лістингу 3.5.

### Лістинг 3.5 – Програмний код формування промпту для cold email

```
result['user_prompt'] = (
    f"Write a professional email for the first contact with a
    {company.name} company from the {company.industry} industry, "
    f"briefly introducing your company services with a mention
    of the capabilities of your project. "
    f"Take in account company mission:
    {company.short_description} "
    f"The goal of the email is to spark interest in our offer
    and invite further discussions on potential collaboration. "
    f"The email should sound natural and professional.
    Highlight the benefits of using
    {sender.sender_project.project_name or 'our solution'}. "
    f"The recipient's name and position may be unknown to you,
    so you will craft the emails based on the available
    information. "
    f"The emails should maintain a professional, but not
    overly formal tone, avoiding grandiose language. "
    f"Use the following data to form the bottom email
    signature without any changes:\n"
    f"{sender.sender_name}\n"
    f"{sender.sender_position} of {sender.sender_company}\n"
    f"Phone: {sender.sender_phone}\n"
    f"Website: {sender.sender_company_website}\n"
)
```

Цей запит орієнтує модель на створення листа, який викликає інтерес і відкриває шлях до подальшої комунікації. Він містить базову інформацію про компанію-отримувача, а також інструкції щодо стилю, структури та підпису.

У разі отримання позитивної відповіді від потенційного партнера система формує warm email, який має за мету перевести комунікацію до конкретних дій – зокрема, обговорення умов співпраці, функціональний код якого можна побачити у лістингу 3.6.

**Лістинг 3.6 – Програмний код формування промπτу warm email на основі відповіді клієнта**

```
self.prompt.user_prompt = (
    "As a professional and tactful assistant tasked with
    advancing the negotiations to the contract stage with a client
    who has expressed interest in collaboration. "
    "Use a polite but confident tone. The goal is to discuss
    the details of the partnership and move towards signing the
    contract. "
    f"Based on the company's response:
    {self.response_email.body} with subject:
    {self.response_email.subject} "
    f"and previously sent cold email text:
    {self.cold_email.body}, generate warm email. "
    "Express sincere gratitude for their interest and offer to
    discuss the specifics of the collaboration. "
    "Politely inquire when it would be convenient to have a
    call or meeting to go over the details and finalize the
    contract. "
    "Include link to the company website.")
```

Тут модель отримує інформацію про попередню розмову – текст холодного листа та відповідь компанії – і на цій основі формує виважене, вдячне і конструктивне повідомлення з пропозицією зустрічі або дзвінка.

Генерація листів реалізована як окремий сервіс у хмарі, що працює асинхронно: після завершення парсингу контактних даних система автоматично формує запит до Azure OpenAI для кожної релевантної компанії. Результати зберігаються у відповідні JSON або CSV-файли, які потім можуть бути використані для масової email-розсилки або ручної перевірки.

Також було реалізовано базову перевірку семантичної унікальності текстів: наприклад, при генерації сотень листів до інвесторів з однієї галузі, система перевіряє, наскільки сильно відрізняються результати, аби уникнути повторів або спаму. За потреби можна вмикати режим «temperature boosting» – більш креативний стиль письма з варіативнішими формулюваннями.

Інтеграція штучного інтелекту дозволила повністю автоматизувати процес створення контенту для email-комунікації, що раніше вимагав значних людських ресурсів. В результаті система здатна генерувати сотні персоналізованих повідомлень на день, адаптованих до профілю компанії, цілей звернення та специфіки ринку.

### 3.5 Розробка REST API для взаємодії з фронтендом

Для забезпечення повноцінної інтеграції бекенд-системи з користувацьким інтерфейсом було розроблено REST API, який реалізовано за допомогою фреймворку FastAPI. Такий підхід дозволив створити гнучкий, масштабований та добре документований API, який відповідає сучасним вимогам до взаємодії фронтенду та серверної частини вебзастосунку.

REST API виступає центральною точкою взаємодії між клієнтом і всією логікою платформи. Було реалізовано низку основних функціональних ендпоінтів, які забезпечують базові сценарії користувача.

Модуль API:

- POST /api/token/ – створення токена доступу (авторизація);
- POST /api/token/refresh/ – оновлення токена доступу.

Модуль file\_loader:

- GET /file\_loader/api/v1/websiteinfo/ – список компаній (отриманих із Crunchbase);
- POST /file\_loader/api/v1/websiteinfo/ – додати компанію;
- GET /file\_loader/api/v1/websiteinfo/{uuid}/ – перегляд інформації про одну компанію;
- GET /file\_loader/api/v1/websiteinfo/cities/, /industries/, /company\_names/ – отримання фільтрів (міста, індустрії, назви);
- GET/POST /file\_loader/api/v1/emails/ – робота з холодними листами, які стосуються компаній (чернетки, історія).

Модуль mailhub:

- POST /mailhub/api/v1/emails/{id}/generate-warm-email/ – згенерувати теплий лист (warm email);
- POST /mailhub/api/v1/emails/{id}/send\_email/ – відправити лист;
- GET/POST /mailhub/api/v1/projects/ – створення «проектів», кампаній або груп листів;
- GET/POST /mailhub/api/v1/prompts/ – промпти, які використовуються для генерації контенту (можуть бути шаблонами).

Модуль mailhub:

- GET/POST /mailhub/api/v1/sender\_profiles/ – управління профілями відправника (ім'я, email, підпис тощо).

Модуль users:

- POST /users/v1/user/login/ – логін користувача;
- GET /users/v1/user/verify-email/{token}/ – верифікація email;
- GET/POST /users/api/v1/user\_infos/ – розширена інформація профілю користувача (портфоліо, біо тощо);

- GET/POST /users/api/v1/bookmarks/ – збереження компаній/листів у закладки;
- POST /users/api/v1/bookmarks/{id}/generate\_prompt/ – створити промпт (AI-запит) для збереженого контакту;
- GET/POST /users/api/v1/notifications/ – повідомлення про відповіді або статус листів;
- POST /users/api/v1/notifications/read\_all/, /delete\_read/ тощо – керування прочитаними/непрочитаними сповіщеннями.

Також, на цьому етапі був задеплойований Swagger (рисунок 3.3) [11].

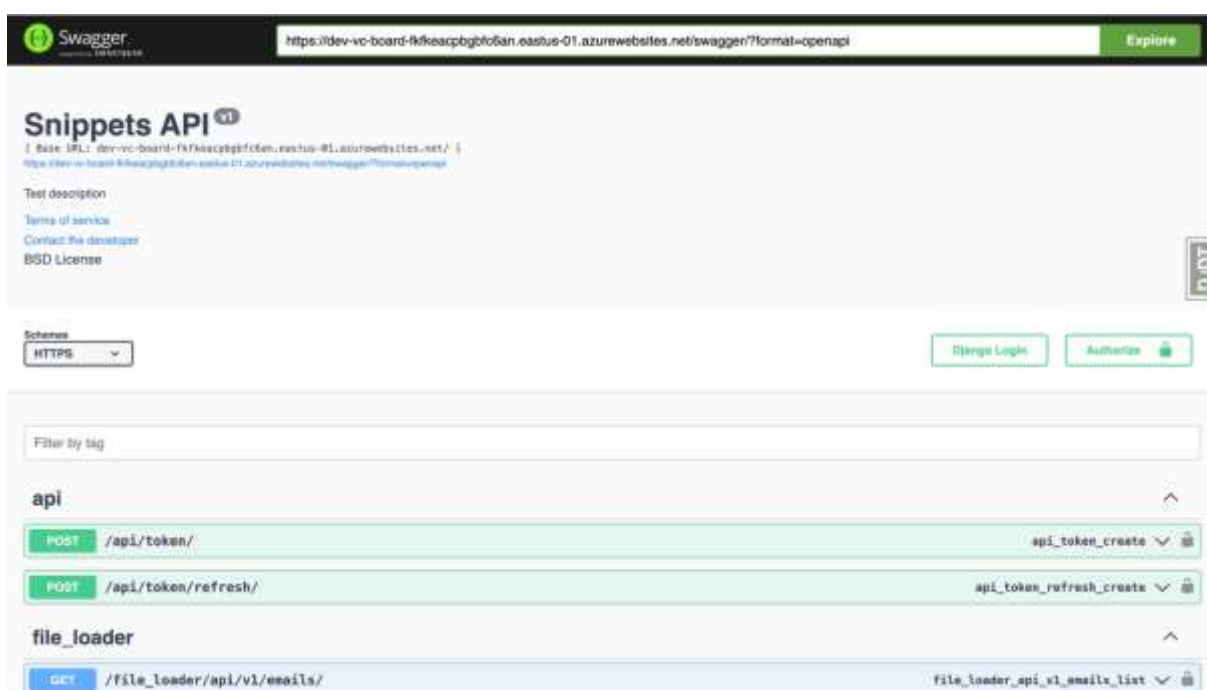


Рисунок 3.3 – Опублікований swagger для Front-end

Уся система побудована з урахуванням вимог безпеки, масштабованості та адаптації до змін у зовнішніх джерелах даних, що створює надійну основу для подальшого розширення функціональності платформи.

## 4 РОЗРОБКА ВІЗУАЛЬНОЇ ЧАСТИНИ

### 4.1 Розробка дизайну у середовищі Figma

Дизайн майбутнього інтерфейсу був реалізований за допомогою графічного редактора Figma. Figma – це застосунок для роботи із векторною графікою, що дозволяє створювати різні макети, елементи інтерфейсу, зображення, ілюстрації, проектування і багато інших креативів.

Цей інструмент популярний серед веб- та UI/UX-дизайнерів, SMM-менеджерів, stories-мейкерів та інших digital-експертів. Бо він зручний для використання, доступний і тут можливо працювати над проектом віддалено разом з іншими учасниками [13].

Робота велась одразу в двох напрямках: для десктопних і мобільних пристроїв, що забезпечує повну адаптивність і зручність користування на різних платформах.

На початку був створений UI Kit – набір повторно використовуваних компонентів, кольорів, кнопок та текстових стилів, що дозволяє підтримувати єдину стилістику та швидко компоувати нові елементи.

Основні характеристики:

- основним кольором інтерфейсу є темний фон (#1E1E2F), з акцентними бірюзовими (#00E1D1) для активних елементів, зеленими та червоними – для повідомлень про успіх чи помилки. Така палітра є приємною для очей і комфортною для роботи в темному режимі;

- кнопки (звичайні, успішні, небезпечні, неактивні), інпут-поля, фільтри, пагінація, таби, аватари, значки дій, скелетони;

- завдяки UI Kit однакові елементи повторно використовуються на всіх сторінках (наприклад, кнопки, поля введення, фільтри), що зменшує кількість унікального коду та прискорює розробку. Нижче наведено рисунок 4.1 основних компонентів.



Рисунок 4.1 – UI Kit, основні компоненти сайту scrap-lab

Після успішного складання UI Kit, було розставлено пріоритетність і визначення в цілому, які сторінки мають бути на дизайні. Дизайн розроблявся як під десктопну версію, так і під мобільну, тож є дизайн однієї сторінки двома розширеннями, як можна бачити на рисунках 4.2 та 4.3.



Рисунок 4.2 – Середовище розробки сторінок для десктопу

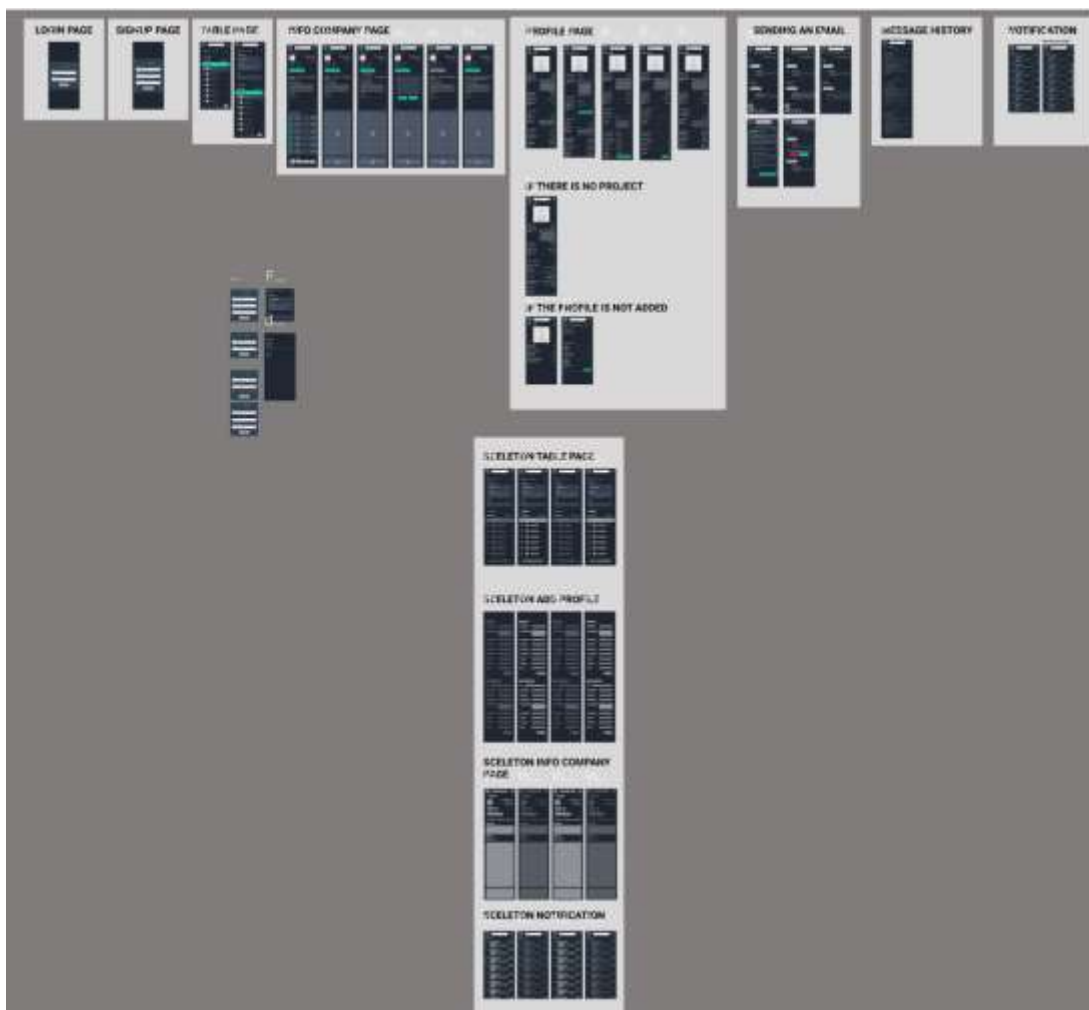


Рисунок 4.3 – Середовище розробки сторінок з респонсивом

Нижче наведено порядок сторінок, які розроблялися.

Sign Up / Log In Page сторінка є першою точкою взаємодії користувача із застосунком. Вона має два режими: авторизація та реєстрація. При реєстрації користувач вводить електронну пошту, пароль та підтвердження пароля.

При першому вході в систему, користувачеві прийде лист на електронну пошту з кодом, який потрібно буде ввести в полі, яке з'явиться після натискання на кнопку «Sign up». При логіні – лише електронну пошту і пароль. Передбачено можливість перегляду пароля за допомогою іконки «ока» (рисунок 4.4), що підвищує зручність введення даних.

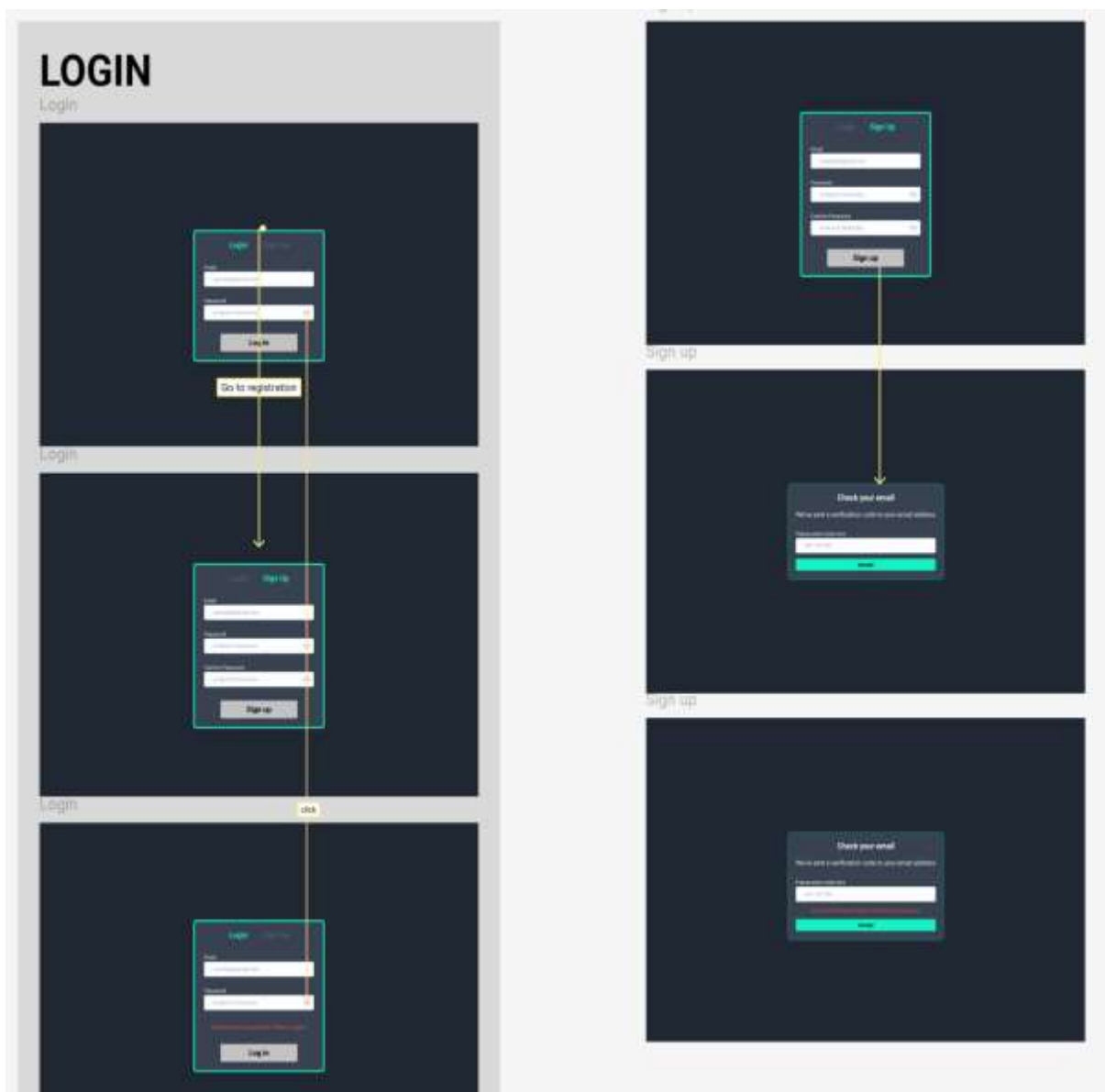


Рисунок 4.4 – Дизайн сторінки Sign up/ Log in

Table page відображається після входу до системи (рисунок 4.5), яка візуалізує зібрані компанії у вигляді таблиці. Вона є центральним елементом застосунку. На цій сторінці користувач має змогу переглянути компанії, за допомогою пагінації, може відображати більше результатів.

Також, користувачеві доступне сортування за алфавітним порядком та Search панель, яка дозволяє шукати компанії за їх назвою, індустрією та локацією. При завантаженні даних, було реалізовано Skeleton (рисунок 4.6) для підвищення взаємодії користувача із сайтом.

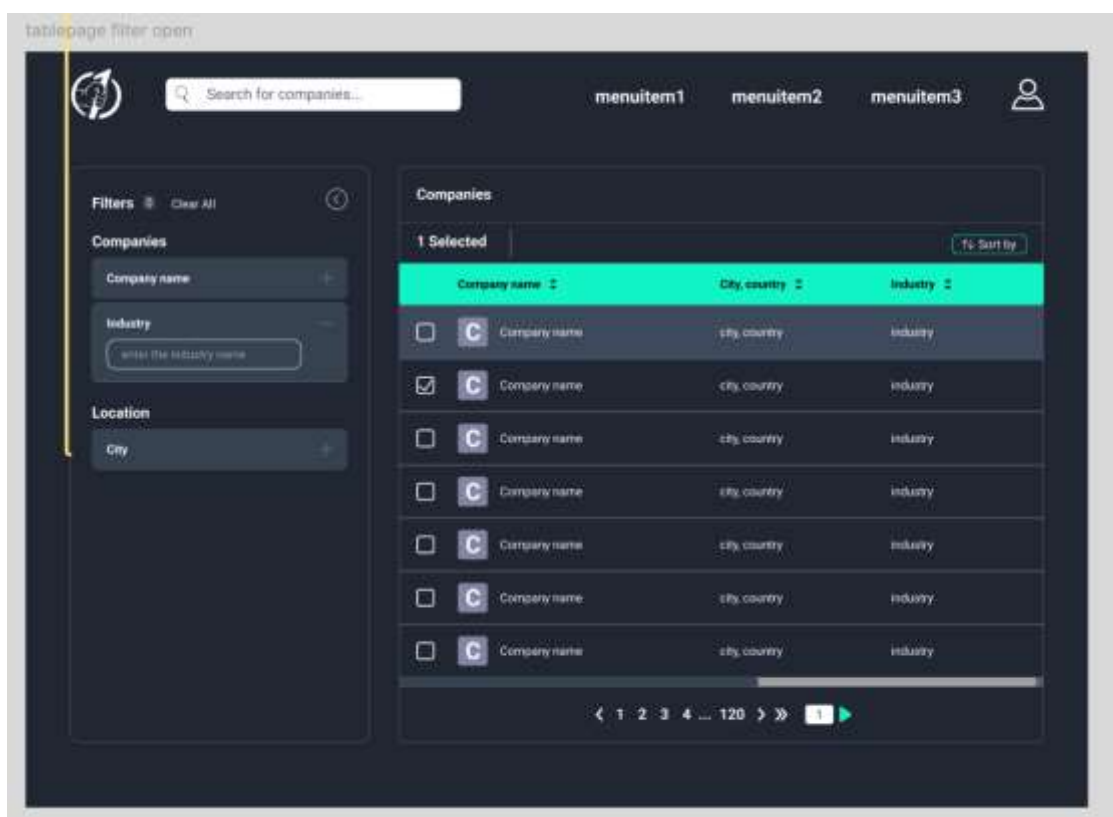


Рисунок 4.5 – Дизайн сторінки Table Page

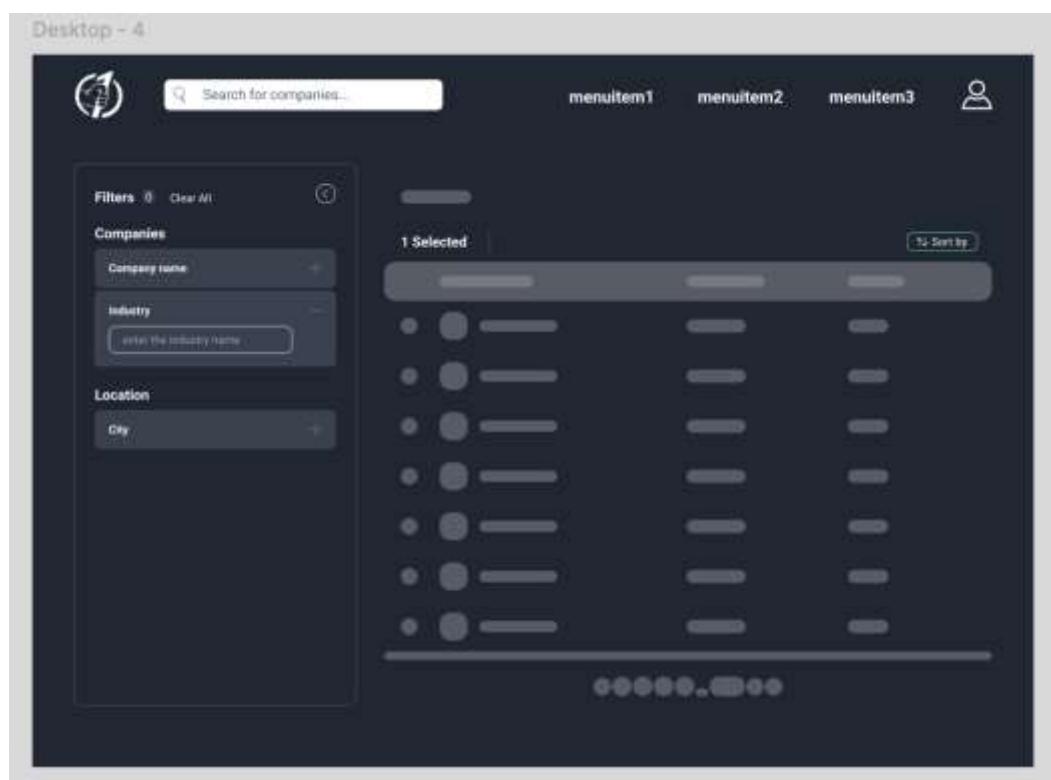


Рисунок 4.6 – Skeleton Table Page

Profile page – одна з найважливіших сторінок у застосунку, де відбувається початкова ідентифікація користувача та надається можливість презентувати власний проект. Саме звідси пишуться індивідуальні листи до компаній. Тут можна заповнити свою особисту інформацію: ім'я, посаду, компанію, телефон, веб-сайт. Заповнити профіль проекту: назва, опис, переваги, стратегія зростання, рентабельність інвестицій, ринковий потенціал. Після заповнення форми всі поля зберігаються в базі даних і використовуються для створення холодних/теплих електронних листів.

Побачити дизайн можна на рисунку 4.7.

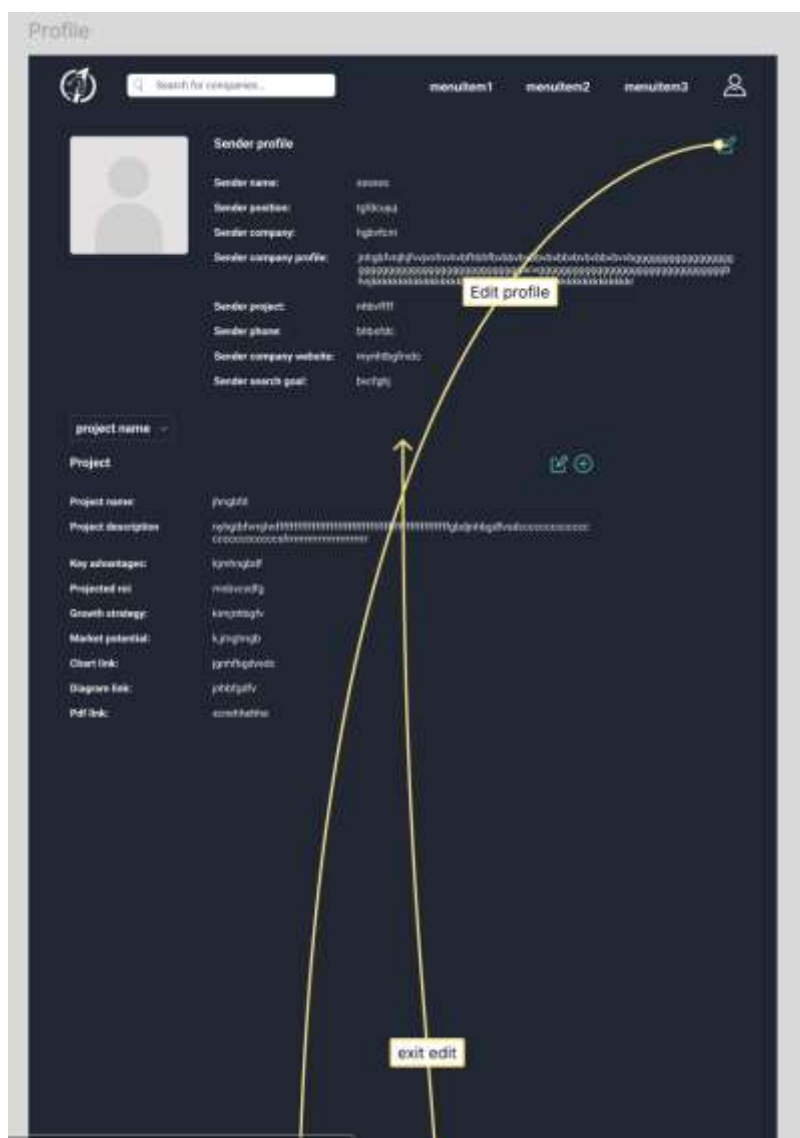


Рисунок 4.7 – Дизайн Profile Page

Info Company Page зображена на рисунку 4.8. Вона надає користувачеві можливість ознайомитися з детальною інформацією про потенційного партнера – компанію, якій буде адресовано лист. Основна мета – вказати назву компанії, місію, галузь, місцезнаходження, відображення контактних даних, наприклад, ділових електронних листів, якщо вони були вилучені через парсинг або обробку даних, а також перехід на сторінку створення електронного листа безпосередньо з цієї сторінки.

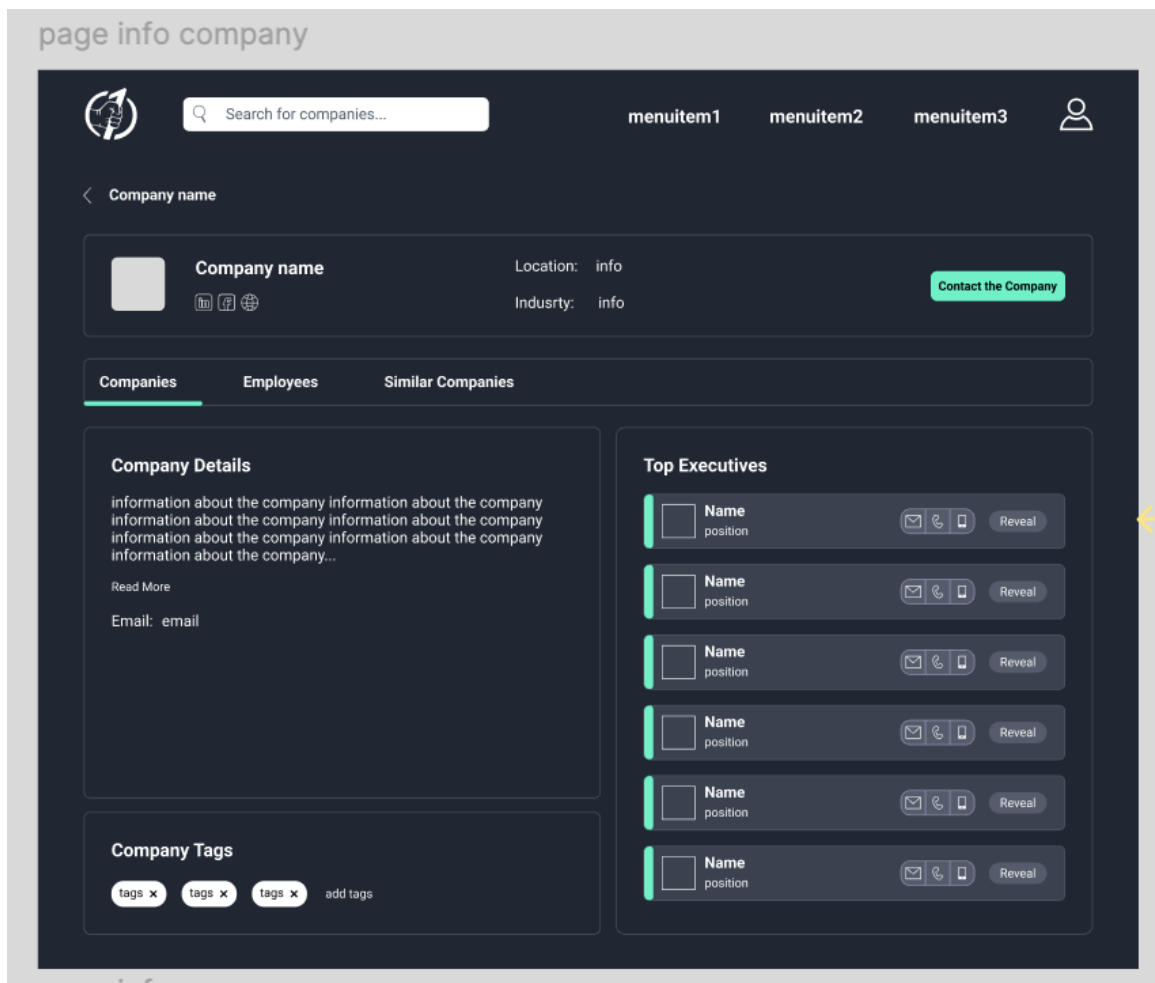


Рисунок 4.8 – Дизайн Info Company Page

Sending Emails дуже важлива сторінка, яка дозволяє писати, редагувати та надсилати теплий або холодний лист вибраній компанії. Головною функцією є редагування чернетки листа (штучний інтелект

пропонує автоматизовану чернетку, яку користувач може редагувати вручну), перемикається між холодним листом (перший контакт) та теплим листом (другий контакт після виявлення зацікавленості), відображення попереднього перегляду остаточної версії листа та його надсилання. На рисунку 4.9 можна побачити її дизайн.

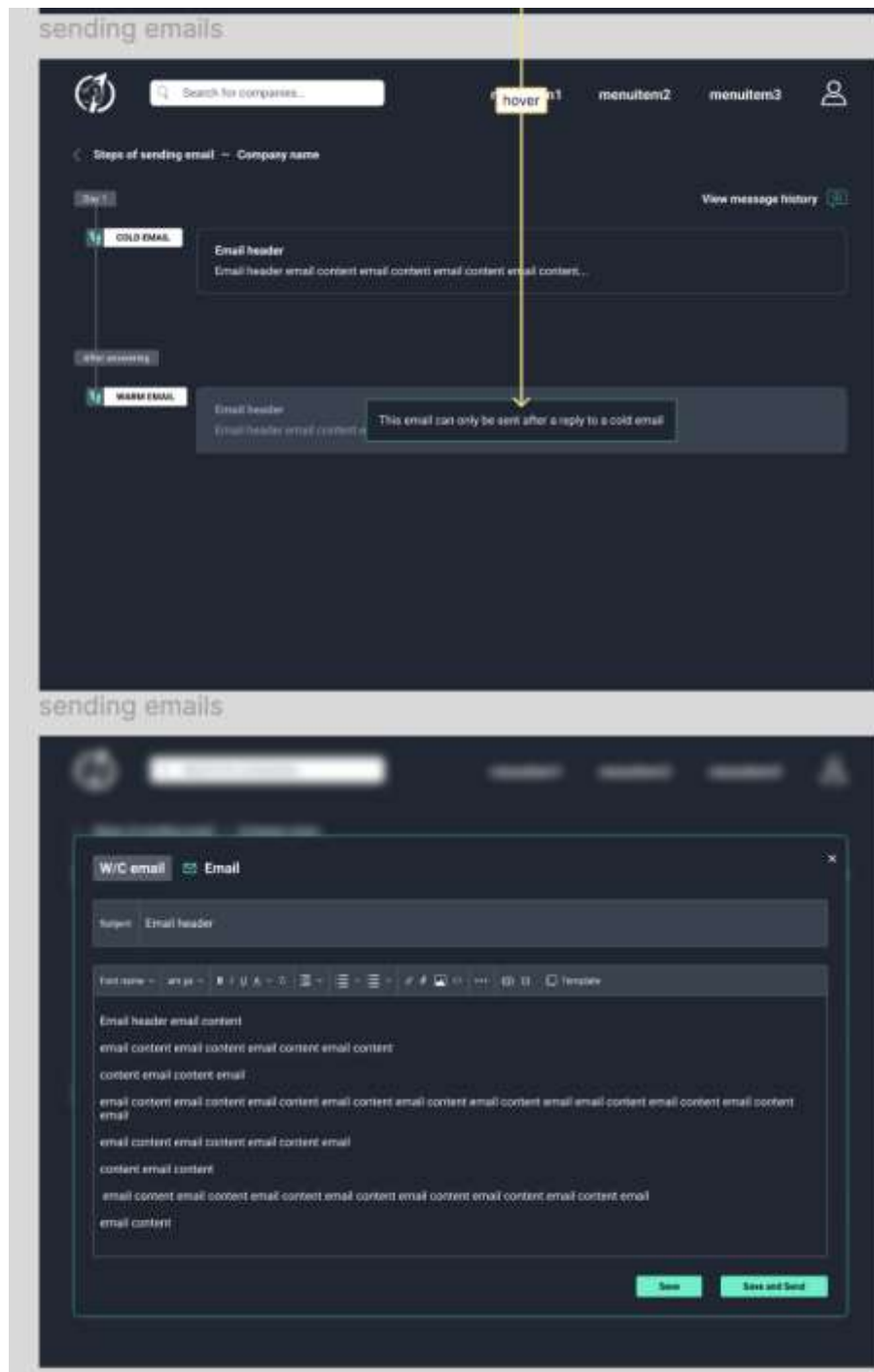


Рисунок 4.9 – Дизайн сторінки Sending Emails

Message History сторінка дозволяє відстежувати історію спілкування з компаніями. Це необхідно для уникнення кількох запитів та ефективнішої обробки процесу продажу/співпраці. Тут можна переглянути список усіх ваших вихідних та вхідних електронних листів для кожної компанії, фільтрувати за датою, компанією чи статусом (надіслано/отримано/очікувано відповідь) та переглянути вміст кожного електронного листа.

Дизайн цієї сторінки можна побачити на рисунку 4.10.



Рисунок 4.10 – Дизайн сторінки message History

Notifications Page підтримки, яка додає рівень обізнаності для відображення нових подій у вигляді повідомлень, сповіщень про нову відповідь від компанії. Дизайн цієї сторінки можна побачити на рисунку 4.11.

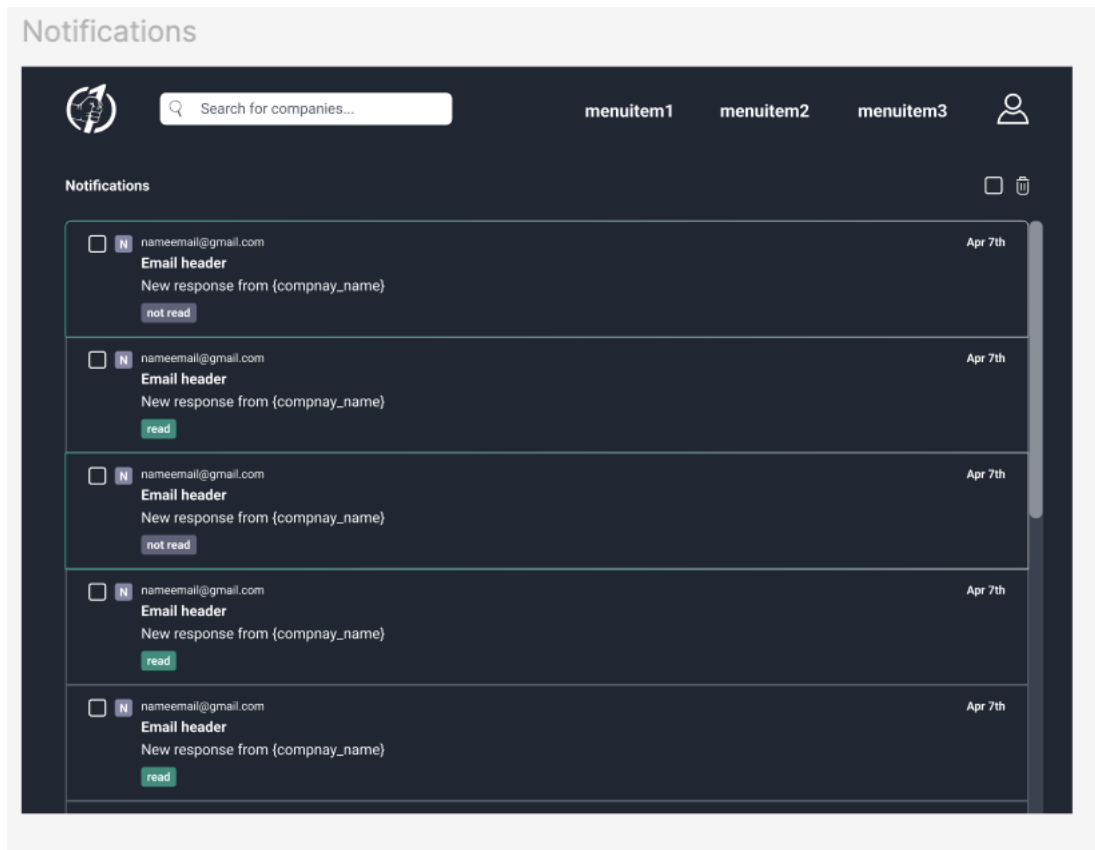


Рисунок 4.11 – Дизайн Notification Page

## 4.2 Реалізація фронт-енд частини

Було вирішено впровадити компонентно-орієнтований підхід (Component-Based Architecture), який підтримує повторне використання частин інтерфейсу, спрощує тестування та підтримку.

Організація проекту окремо наголошує на:

- сторінках (pages) з маршрутами;
- компонентах інтерфейсу (components);

- сервісах для обробки API (services);
- помічниках (utils);
- стилях (styles);
- глобальних файлах конфігурації.

Стек технологій фронт-енду був такий:

- React + TypeScript – для безпечної розробки за типами;
- Vite – як сучасний пакет з високою швидкістю збірки;
- React Router DOM – для реалізації маршрутизації клієнтів;
- Axios – для виклику API;
- TailwindCSS – для швидкої стилізації з уніфікованим виглядом.

Також було розглянуто Redux, але під час роботи виявилось, що глобальний стан не потрібно керувати складним чином. Усі основні випадки були реалізовані на основі хуків (використовуючи useState, useEffect, useContext), тому Redux практично не використовувався. Хуки – це програмні механізми, які надають розробникам можливість вбудовувати свій код у системні процеси або змінювати поведінку програми на глибшому рівні. Основна роль хуків полягає в тому, щоб надавати точки розширення або зміни стандартного потоку виконання програми, даючи змогу в такий спосіб гнучкіше керувати її поведінкою. React надає набір вбудованих хуків, які спрощують управління станом і життєвим циклом компонентів. Наприклад:

- useState дозволяє функціональним компонентам додавати локальний стан. Він повертає пару значень: поточне значення стану і функцію для його оновлення;

- useEffect використовується для виконання сайд-ефектів у функціональних компонентах. Наприклад, він може бути використаний для виконання коду після завершення рендеру, оформлення підписки на зовнішні дані та інших сценаріїв;

- useContext надає доступ до значення контексту в компоненті, дозволяючи йому використовувати це значення.

Переваги використання хуків:

- спрощення коду: наприклад, під час використання `useState` у React код, який раніше потребував класових компонентів, тепер можна виразити більш компактним і зрозумілим функціональним кодом;
- підвищення читабельності коду: відсутність необхідності в конструкціях класів, біндингах методів і конструкторах знижує надмірність коду, роблячи його більш читабельним для розробників;
- перевикористовуваність коду: хуки легко перевикористовуються в різних компонентах, що підвищує ступінь абстракції. Це прискорює розробку і дає змогу створювати більш гнучкі та масштабовані архітектури [14].

На початковому етапі розробки фронтенду було обговорено та остаточно визначено основні інструменти та технології, які будуть використовуватися в проєкті. Базовий стек вже був визначений і включав сучасні веб-технології, зокрема React. Предметом обговорення в команді були методи стилізації інтерфейсу: було розглянуто різні бібліотеки та системи стилізації, виходячи з вимог до гнучкості, масштабованості та можливості підтримувати один стиль у всьому застосунку. На рисунку 4.12 є файли загальних компонентів.

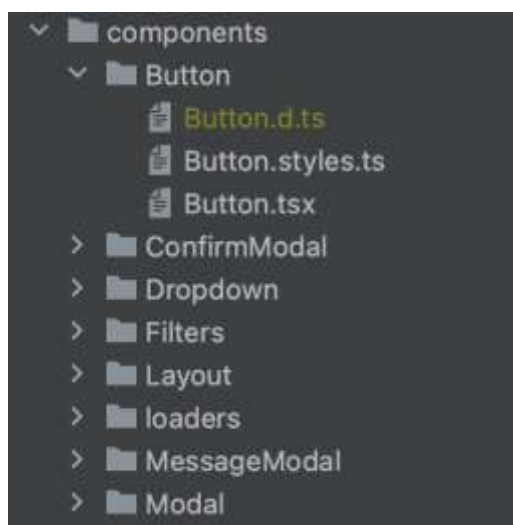


Рисунок 4.12 – Загальні компоненти веб-сайту scrap-lab

Було розроблено модуль `services`, який обгортає логіку запитів до бекенду через `Axios`. Особлива увага була приділена:

- обробці токенів доступу/оновлення: якщо позначено `access_token`, сервіс автоматично оновлює токен через перехоплювачі;
- централізованій обробці помилок (401, 403 тощо);
- простому підключенню до API різних модулів (профіль, компанія, електронна пошта).

Для забезпечення централізованої взаємодії з API та керування токенами авторизації було додано новий модуль `userApi.ts`, що використовує `Axios` (лістинг 4.1).

Лістинг 4.1 – Програмний код з базовою URL-адресою та заголовками

```
const api = axios.create({
  baseURL: API_URL,
  headers: {
    'Content-Type': 'application/json',
  },
});
```

Усім запитам автоматично надається токен доступу з `localStorage` через перехоплювач (лістинг 4.2).

Лістинг 4.2 – Програмний код з наданням доступу токenu

```
api.interceptors.request.use((config) => {
  const accessToken = localStorage.getItem('accessToken');
  if (accessToken) {
    config.headers.Authorization = `Bearer ${accessToken}`;
  }
  return config;
});
```

Якщо термін дії токена доступу закінчився, він автоматично поповнюється за допомогою токена оновлення. Якщо це не вдається, користувача буде перенаправлено на сторінку входу (лістинг 4.3).

Лістинг 4.3 – Програмний код з оновленням токена та обробкою помилок

```
api.interceptors.response.use(
  (response) => response,
  async (error) => {
    const originalRequest = error.config;
    if (error.response?.status === 401 &&
!originalRequest._retry) {
      // логіка оновлення токенів...
    }
    return Promise.reject(error);
  }
);
```

Після невдалої авторизації викликається процес очищення даних, це наведено у лістингу 4.4.

Лістинг 4.4 – Програмний код з очищенням даних

```
export const clearAuthData = () => {
  localStorage.removeItem('accessToken');
  localStorage.removeItem('refreshToken');
  window.location.replace(ROUTES.LOGIN);
};
```

Також є форма входу, яка надсилає дані до бекенду та зберігає токени в localStorage. Далі йде реалізація заголовка – він забезпечує навігацію між сторінками, індикатор авторизації, меню профілю.

Крім того, на сторінці зі списком компаній також встановлено блок фільтрації. Блок фільтрації дозволяє сортувати компанії одразу за галуззю, місцем розташування, стадією розробки тощо. Для цього реалізовано функцію вибору, введення та видалення рядків запиту. На цій сторінці користувач бачить логотип компанії, назву, соціальні мережі, сайт та галузь. Крім того, реалізовано процес підтвердження того, чи профіль користувача заповнений: лише після створення профілю, який вважає цікавим, є можливість зв'язатися з компанією.

Кнопка «Зв'язатися з компанією» активується лише за наявності у користувача заповненого профілю, а компанія має більше однієї адреси електронної пошти. У деяких випадках користувачеві пропонується вибрати одну з них, і відкривається сторінка списку розсилки. Якщо умови не виконуються, відображається відповідне повідомлення.

Для покращення UX реалізовано автоматичне закриття модального повідомлення через 3 секунди. Також реалізовано обробку кліків за межами модального вікна для закриття вікна, функціональний код якої можна побачити у лістингу 4.5.

#### Лістинг 4.5 – Програмний код з очищенням даних

```
const handleClick = () => {
  if (!projectsExist || !hasEmail) {
    setShowModalMessage(true);
    if (timeoutRef.current) clearTimeout(timeoutRef.current);
    timeoutRef.current = setTimeout(() => {
      setShowModalMessage(false);
      timeoutRef.current = null;
    }, 3000);
  } else {
    if (hasEmail) {
      setShowModalEmails(true);
    } else {
      if (company?.emails?.length === 1) {
```

### Продовження лістингу 4.5

```

        handleClickEmail(company.emails[0].id);
    }
}
}
};

```

Після логіки авторизації і паралельно з фільтрацією, розроблялася сторінка таблиці компаній. Вона включає:

- відображення списку компаній з пагінацією;
- інтерактивні елементи (посилання на сторінку інформації про компанію, кнопка «Надіслати електронний лист»);
- підтримка фільтрації та пошуку;
- завантажувачі, порожні стани та обробка помилок.

Таблиця побудована на користувацькому компоненті: Tailwind + TypeScript. CSS-фреймворк – це заздалегідь підготовлений набір CSS-файлів. Ідея в тому, щоб при створенні кожного нового проєкту використовувати готовий набір файлів, у яких для вас уже створений макет, задане шрифтове оформлення та написаний весь основний код, загальний для більшості сайтів.

Одним із найпопулярніших CSS-фреймворків являється Tailwind. Він дозволяє швидше писати та підтримувати код програми. Використовуючи цю платформу, яка орієнтована на утиліти, не потрібно писати власний CSS для оформлення програми. Натомість необхідно використовувати службові класи [15]. Приклад стилізації можна побачити на лістингу 4.6.

### Лістинг 4.6 – Програмний код з прикладом стилізації

```

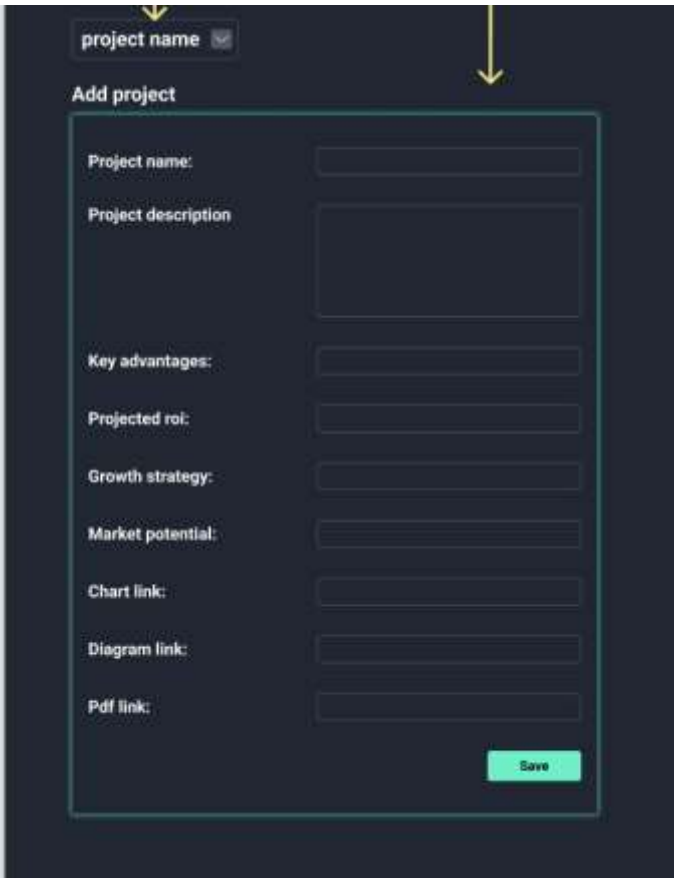
export const TdContainer = styled.div`
display: flex;
gap: 1rem;
justify-content: start;
align-items: center;

```

### Продовження лістингу 4.6

```
padding: 0 4px;  
cursor: pointer;  
&:has(svg) {  
  justify-content: center;  
}  
`;  
`;
```

Після цього розпочалася розробка сторінки профілю. Сторінка реалізована як багатоетапна форма з валідацією. Користувач отримує доступ до функціональності електронної пошти після збереження профілю із заповненими полями про інформацією про проект (рисунок 4.13), а також редагувати свої дані, такі як: фото, ім'я, посаду, компанію, профіль (якщо є), телефон, свій веб-сайт і мету реєстрації (рисунок 4.14).



The image shows a dark-themed user interface for adding a project. At the top, there is a dropdown menu labeled 'project name'. Below it is the title 'Add project'. The main form area contains several input fields, each with a label: 'Project name:', 'Project description', 'Key advantages:', 'Projected rol:', 'Growth strategy:', 'Market potential:', 'Chart link:', 'Diagram link:', and 'Pdf link:'. A green 'Save' button is positioned at the bottom right of the form. Two yellow arrows are overlaid on the image: one points to the 'project name' dropdown, and the other points to the 'Save' button.

Рисунок 4.13 – Заповнення форми проекту на сторінці акаунту



Останнім кроком була реалізація сторінки відправлення електронної пошти, а також сторінка, яка показує історію переписки з компанією. Сторінка «Sending Emails» Вона має два етапи:

- редагування згенерованих холодних та теплих електронних листів;
- підтвердження та відправлення (рисунок 4.15).

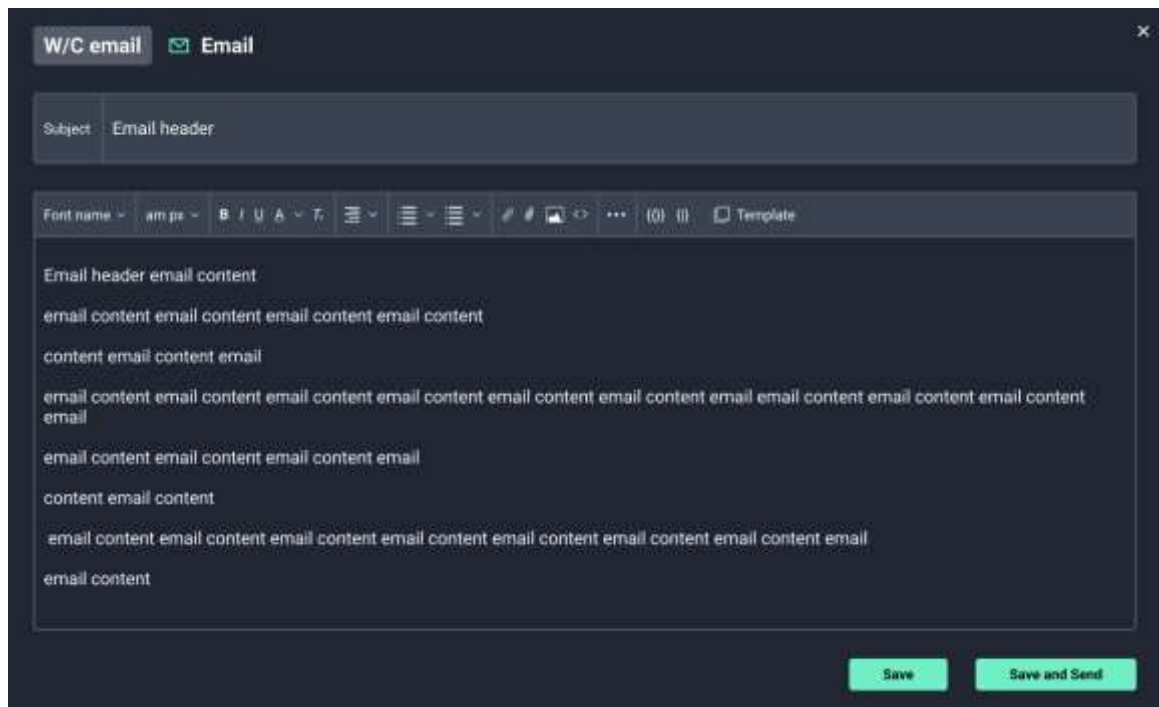


Рисунок 4.15 – Редагування та відправка cold/warm email

На етапі генерації електронної пошти, бекенд викликається для створення текстів через OpenAI API. Електронний лист генерується з даних, які раніше ввів користувач або система. Цей процес включає кілька кроків та здійснює виклики до бекенду через відповідні кінцеві точки REST, які були раніше визначені та протестовані на етапі розробки API за допомогою Swagger. Першим кроком є отримання відповідного запиту, який використовується як запит для генерації тексту за допомогою API OpenAI. Цей рівень використав функцію `getPrompt` (лістинг 4.7) з сервісного рівня для реалізації цього.

### Лістинг 4.7 – Програмний код із запитом на prompt

```
const getPrompt = async (body: PromptCreationProps):
Promise<Prompt> => {
  const allUserPromptsResponse:
AxiosResponse<ResponseAllPrompts> = await
api.get('/mailhub/api/v1/prompts/');
  const promptByCompanyEmailId =
allUserPromptsResponse.data.results.find(
  (prompts) => prompts.company_email === body.company_email,
  );
  return promptByCompanyEmailId || (await
getCreatePrompt(body)).data;
};
```

Цей код надсилає запит до `/prompts/`, щоб перевірити, чи існує потрібний запит, і, якщо його ще немає, його потрібно створити за допомогою `POST /prompts/`.

Після отримання запиту система намагається знайти холодний електронний лист, який був згенерований раніше. Реалізація функціоналу можна побачити на лістингу 4.8.

### Лістинг 4.8 – Програмний код із логікою cold email

```
const getColdEmail = async (promptId: number):
Promise<Email> => {
  const allColdEmailsResponse: AxiosResponse<Email[]> =
await api.get('/mailhub/api/v1/emails/');
  const coldEmailByPromptId =
allColdEmailsResponse.data.find(
  (coldEmails) => coldEmails.prompt === promptId,
  );
  return coldEmailByPromptId || (await
getGenerateColdEmailTxt(promptId)).data;
};
```

Якщо такого електронного листа не існує, він створюється за допомогою POST до /emails/, що призводить до виклику API OpenAI для створення контенту.

Після того, як відповідний холодний електронний лист створено або вибрано, його природним наступним кроком є перетворення на теплий електронний лист, якщо компанія відповіла. Функціонал можна побачити на лістингу 4.9.

#### Лістинг 4.9 – Програмний код із логікою warm email

```
const getGenerateWarmEmailTxt = async (coldEmailTxtID:
number): Promise<AxiosResponse<Email>> => {
    return await
    api.post(`/mailhub/api/v1/emails/${coldEmailTxtID}/generate-
warm-email/`);
};
```

Ця кінцева точка також викликає API OpenAI, використовуючи попередньо створений текст холодного електронного листа як контекст для створення більш персоналізованого теплого листа. Саме тут стали в нагоді раніше розроблені та задокументовані кінцеві точки Swagger. Їх наявність значно полегшила інтеграцію, зробила поведінку API передбачуваною та спростила тестування.

Після створення обох електронних листів їх можна редагувати через модальне вікно. У компоненті EmailModal розроблено той самий інтерфейс, який містить:

- поле редагування теми;
- поле редагування тексту листа;
- кнопки «Зберегти» та «Зберегти й надіслати».

Функція handleSaveAndSend має логіку для надсилання вмісту електронного листа, який можна побачити на лістингу 4.10.

#### Лістинг 4.10 – Програмний код із логікою збереження та відправки імейлів

```
const handleSaveAndSend = async (needSent?: boolean) => {
  if (emailSubject && emailBody && email) {
    try {
      const saveEmailResponse = await
emailServices.updateEmailText({
      ...email,
      subject: emailSubject,
      body: emailBody,
    });

```

Воно надсилає запит PUT до PUT /emails/{id}/, передаючи новий оновлений вміст електронного листа.

У випадку, якщо користувач намагається закрити вікно з очікуваними змінами, ConfirmModal запитує підтвердження. Це захищає від випадкової втрати відредагованого матеріалу (рисунок 4.16).

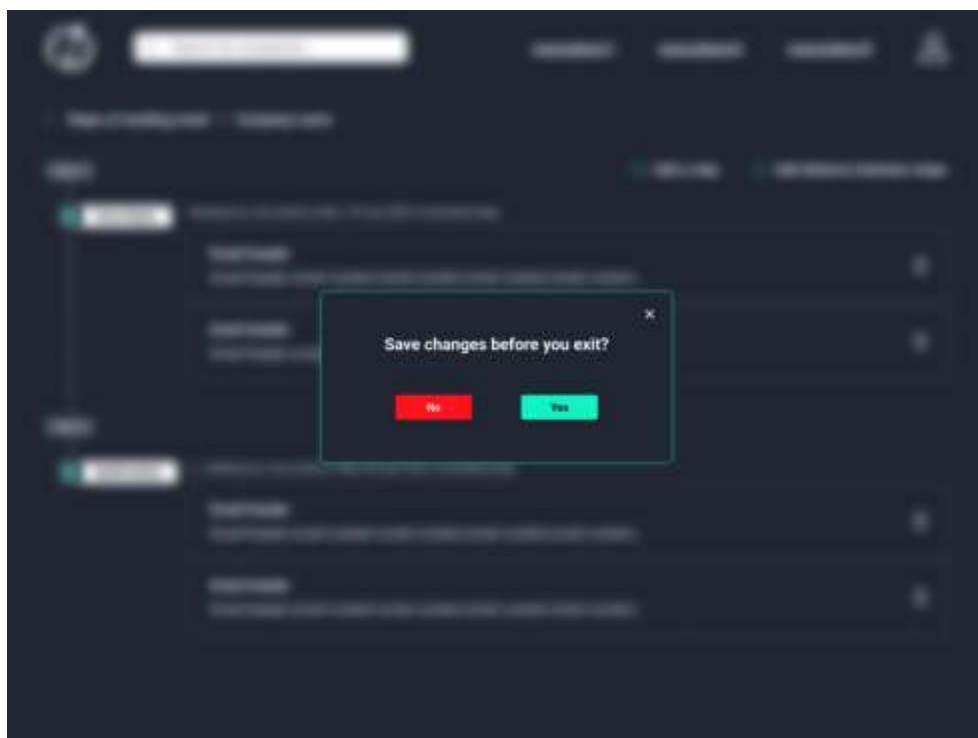


Рисунок 4.16 – Вікно від випадкової втрати контенту ConfirmModal

На завершальному було проведено інтеграційне тестування всіх сторінок. Були виявлені та усунені незначні проблеми інтерфейсу користувача, покращена адаптивність, а також рефакторинг коду деяких компонентів. Додаток було протестовано на стабільну взаємодію з API та правильну обробку помилок авторизації.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було реалізовано повноцінний веб-додаток, орієнтований на автоматизацію створення та надсилання персоналізованих електронних листів, щоб знайти першу роботу. Головною ідеєю було надати користувачеві зручний інтерфейс для генерації комунікацій з потенційними клієнтами або партнерами, використовуючи сучасні інструменти штучного інтелекту. Для цього було побудовано архітектуру взаємодії клієнт-сервер, де фронт-енд керує інтерфейсною частиною, а бек-енд має бізнес-логіку, доступ до бази даних та інтеграцію з OpenAI API для генерації тексту.

Кваліфікаційна робота охоплює всі суттєві етапи розробки програмного продукту – від аналізу вимог та проектування структури до кодування, планування взаємодії компонентів, тестування та налагодження. У ході аналізу галузі дослідження було виявлено актуальні потреби в оптимізації процесу проектування високоякісних електронних комунікацій для компаній, які встановлюють перший контакт з цільовою аудиторією. На цій основі було визначено системні вимоги, встановлено функціональні та нефункціональні характеристики, серед яких вирішальним були стабільність, безпека, розширюваність та зручність використання.

Було обрано сучасний та збалансований технологічний стек, що складається з FastAPI, PostgreSQL, SQLAlchemy, Docker на сервері та React, Axios, Tailwind CSS на стороні клієнта. Цей вибір забезпечив достатній рівень гнучкості для підтримки всіх необхідних функцій, а також можливість підтримки масштабованості проекту в майбутньому. Увага була приділена архітектурним рішенням: чітка ізоляція логіки на фронт-енді та бек-енді, обробка помилок, логування, авторизація за допомогою токенів, структура кінцевих точок REST, і все це відповідало вимогам сучасних веб-додатків.

На завершальному етапі проекту було проведено інтеграційне тестування, яке охопило всі функціональні модулі системи. Зокрема, було перевірено коректність взаємодії з API, поведінку системи у випадках помилок авторизації, втрати інтернет-з'єднання, порожніх або некоректних форм. Додатково було протестовано адаптивність інтерфейсу на різних пристроях, проведено рефакторинг окремих компонентів та виправлено виявлені візуальні дефекти. Завдяки цьому, додаток набув завершеного вигляду та стабільно добре працює в повсякденних сценаріях і відповідає всім очікуванням кінцевого користувача.

Паралельно можна спостерігати інтеграцію з OpenAI API як центральну інноваційну частину системи. Це дозволило забезпечити високу персоналізацію та контент листів, що є значною конкурентною перевагою порівняно із поширеними шаблонними рішеннями. Формат генерації та редагування листів, поряд із зручним представленням результатів, дає змогу користувачеві визначити зміст та тон повідомлення перед відправкою.

Таким чином, результати роботи підтверджують актуальність теми та демонструють практичну реалізацію ефективного програмного рішення з використанням сучасних веб-технологій та штучного інтелекту. Запропонований додаток має потенціал для подальшого розвитку, зокрема шляхом додавання аналітики відкриття листів, інтеграції з CRM-системами, розширення сценаріїв користувача та покращення персоналізації. Результати, отримані під час розробки, можуть бути використані як основа для реальних комерційних проектів або досліджень у сфері автоматизації бізнес-комунікацій.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Як змінилося працевлаштування, доходи та настрої українців від початку війни. URL: [https://uk.wikipedia.org/wiki/%D0%9E%D0%BD%D1%82%D0%BE%D0%B%D0%BE%D0%B3%D1%96%D1%8F\\_\(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0\)](https://uk.wikipedia.org/wiki/%D0%9E%D0%BD%D1%82%D0%BE%D0%B%D0%BE%D0%B3%D1%96%D1%8F_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) (дата звернення: 26.04.2025).
2. Пололян С.В., Загальні проблеми працевлаштування молоді в умовах сьогодення. КЗВО «Барський гуманітарно-педагогічний коледж імені Михайла Грушевського», 2021. 3с.
3. 6 корисних ШІ-інструментів, які допоможуть знайти й отримати роботу мрії. URL: <https://happymonday.ua/shi-dlia-poshuku-roboty-korysni-instrumenty> (дата звернення: 26.04.2025).
4. Профіль компанії на Crunchbase – як створити та в чому користь. URL: <https://uaspectr.com/2020/05/13/profil-kompaniyi-na-crunchbase/> (дата звернення: 26.04.2025).
5. Grossberg K. A. New ways B2B marketers use technology to enhance customer engagement. *Strategy & leadership*. 2015. Vol. 43, no. 1. URL: <https://doi.org/10.1108/sl-11-2014-0084> (date of access: 26.04.2025).
6. Fehrenbach D., Herrando C., Österle B. Artificial intelligence applications in the B2B sales funnel. *Journal of Business-to-Business Marketing*. 2025. P. 1–24. URL: <https://doi.org/10.1080/1051712x.2025.2481374> (date of access: 26.04.2025).
7. Why is data enrichment important? CXO level. *Corporate Database Development, Event Database, Data Cleansing, Market Research & Campaign, Worldwide Database Provider Services – CXO Level*. URL: <https://cxolevel.com/Why-is-Data-Enrichment-Important-in-2023.aspx/> (date of access: 26.04.2025).

8. Personalized sales: best practices, tips for B2B pipeline acceleration. *AI for Sales & GTM Teams / Docket*.

URL: <https://www.docket.io/blog/personalized-sales-best-practices> (date of access: 26.04.2025).

9. How large language models (LLMs) benefit businesses. URL: <https://chisw.com/blog/benefits-of-large-language-models-for-business/> (date of access: 26.04.2025).

10. Shrestha, Y. R., Ben-Menahem, S. M., & von Krogh, G. (2019). Organizational Decision-Making Structures in the Age of Artificial Intelligence. *California Management Review*, 61(4), 66-83. URL: <https://doi.org/10.1177/0008125619862257> (date of access: 26.04.2025).

11. Що таке велика мовна модель (Large Language Model, LLM)? URL: <https://thetransmitted.com/adlucem/shho-take-velyka-movna-model-large-language-model-llm/> (дата звернення: 30.04.2025).

12. Підручник з Python. URL: <https://docs.python.org/uk/3/tutorial/index.html> (дата звернення: 30.04.2025).

13. Overview of Azure Cloud Services (classic). URL: <https://learn.microsoft.com/bs-latn-ba/previous-versions/azure/cloud-services/cloud-services-choose-me> (дата звернення: 18.05.2025).

14. Що таке Figma та для кого вона потрібна. URL: [https://cases.media/en/article/sho-take-figma-ta-dlya-kogo-vona-potribna?srsltid=AfmBOooQY7jXwyk\\_Q94yhdkj-ujv1Q\\_qtpwQoMUieqamK5SQcwkDCUci](https://cases.media/en/article/sho-take-figma-ta-dlya-kogo-vona-potribna?srsltid=AfmBOooQY7jXwyk_Q94yhdkj-ujv1Q_qtpwQoMUieqamK5SQcwkDCUci) (дата звернення: 31.05.2025).

15. Що означає hook у розробці ПЗ? URL: <https://foxminded.ua/hukytse/> (дата звернення: 31.05.2025).

16. Що таке Tailwind і навіщо він потрібен? URL: <https://career.cleveroad.com/blog-posts/shcho-take-tailwind-i-navishcho-vin-potriben> (дата звернення: 03.06.2025).

17. Snippets      API. *Swagger*.      URL: <https://dev-vc-board-fkfkeacpbgfcb6an.eastus-01.azurewebsites.net/swagger/>. (дата звернення: 03.06.2025).