Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Факультет інформаційних технологій
(факультет)

Кафедра системного аналізу та управління
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

Здобувачу вищої освіти Логвиненка Олександра Олександровича
академічної групи 124-21-2
спеціальності 124 Системний аналіз
за освітньо-професійною програмою Системний аналіз
на тему: «Аналіз вимог до кандидатів на посаду системних аналітиків у вакансіях на платформах працевлаштування»

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	*к.т.н., доц. Коряшкіна Л.С.*			
розділів:				
Інформаційно-аналітичний	*к.т.н., доц. Коряшкіна Л.С.*			
Спеціальний розділ	*к.т.н., доц. Коряшкіна Л.С.*			
Експериментально-аналітичний	*к.т.н., доц. Коряшкіна Л.С.*			
Рецензент				
Нормоконтролер	*к.ф.-м.н., доц. Хом'як Т.В.*			

Дніпро
2025

ЗАТВЕРДЖЕНО:

завідувач кафедри

Системного аналізу та управління
(повна назва)

_____ *к.т.н., доц. Желдак Т.А.*
(підпис) (прізвище,ініціали)

«____» _____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня *бакалавра*

здобувачу вищої освіти *Логвиненку О. О.* **академічної групи** *124-21-2*
спеціальності: *124 Системний аналіз*
за освітньо-професійною програмою *Системний аналіз*
на тему «*Аналіз вимог до кандидатів на посаду системних аналітиків у вакансіях на платформах працевлаштування»*
затверджену наказом ректора НТУ «Дніпровська політехніка»
від *15.05.2025 р. №336-с*

Розділ	Зміст	Терміни виконання
1.Інформаційно-аналітичний розділ	*Проаналізувати структуру об'єкта дослідження. Визначити предметну область дослідження та проблему, що розв'язується. Обґрунтувати методи виконання поставлених завдань*	10.01.2025 – 01.03.2025
2.Спеціальний розділ	*Розв'язати поставлені задачі: розробити алгоритми та створити систему для автоматизації збору, форматування, обробки та аналізу блоку вимог у вакансіях на платформах працевлаштування*	01.03.2025 – 10.06.2025
3.Експериментально-аналітичний розділ	*Провести аналіз отриманих даних. Зробити висновки та на їх основі побудувати портрет ідеального кандидата на посаду системного аналітика*	01.03.2025 – 10.06.2025

Завдання видано _____ *доц. Коряшкіна Л.С.*
(підпис) (прізвище, ініціали)

Дата видачі: *06.12.2024 р.*

Дата подання до екзаменаційної комісії: 09.06.2025

Прийнято до виконання _____ Логвиненко О. О.

(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 82 с., 49 рис., 7 табл., 5 додатків, 41 джерело.

Об'єкт дослідження – процеси автоматизованого збору, обробки та аналізу текстової інформації з онлайн-ресурсів, що містять опис вакансій.

Предмет дослідження – методи і програмні засоби виокремлення, структурування, збереження та аналізу блоку вимог до кандидатів із неструктурованих описів вакансій.

Мета дослідження: розробка програмного засобу для автоматизованого збору, структуризації та обробки описів вакансій з онлайн-ресурсу robota.ua, з акцентом на виокремлення блоку вимог до кандидатів. Визначені найбільш затребуваних навичок, а також створенні на основі отриманих даних портрету ідеального кандидата на посаду системного аналітика.

В *інформаційно-аналітичному розділі* наведено аналіз об'єкта дослідження та ключових проблем на ньому. Поставлено задачі дослідження та обрано методології їх дослідження та розв'язання.

У *специфічному розділі* розроблено програми для збору, обробки, збереження та аналізу вимог до кандидатів на посаду системного аналітика у вакансіях на сайтах працевлаштування.

Практична цінність отриманих результатів полягає в визначенні актуальних вимог до кандидатів на посаду системного аналітика, які можуть бути використані претендентами, навчальними закладами та роботодавцями.

Ключові слова: СИСТЕМНИЙ АНАЛІТИК, ВАКАНСІЇ, ВИМОГИ, PYTHON, API, ПАРСИНГ, АНАЛІЗ ДАНИХ, POWER BI, ВІЗУАЛІЗАЦІЯ.

ABSTRACT

Explanatory note: 82 p., 49 fig., 7 tables, 5 appendices, 41 sources.

The object of research is the processes of automated collection, processing, and analysis of text information from online resources containing job descriptions.

The subject of the study is methods and software tools for isolating, structuring, storing, and analyzing a block of candidate requirements from unstructured job descriptions.

The purpose of the study: development of a software tool for automated collection, structuring and processing of job descriptions from the online resource robota.ua, with an emphasis on isolating a block of requirements for candidates. The most sought-after skills were identified, as well as creating a portrait of an ideal candidate for the position of systems analyst based on the data obtained.

The *information and analytical section* provides an analysis of the research object and key problems in it. Research tasks are set and methodologies for their research and solution are selected.

A *specific section* has developed programs for collecting, processing, storing, and analyzing requirements for candidates for the position of systems analyst in vacancies on employment sites.

The practical value of the results obtained lies in determining the current requirements for candidates for the position of systems analyst, which can be used by applicants, educational institutions, and employers.

Keywords: SYSTEM ANALYST, VACANCIES, REQUIREMENTS, PYTHON, API, PARSE, DATA ANALYSIS, POWER BI, VISUALIZATION.

ЗМІСТ

ВСТУП

Актуальність теми. У сучасних умовах цифровізації економіки та зростання попиту на IT-спеціалістів онлайн-платформи з працевлаштування стали головним джерелом інформації про відкриті вакансії. Проте більшість таких описів подається у неструктурованому вигляді, що істотно ускладнює їх автоматизовану обробку, порівняння та аналіз. Відсутність єдиного формату подачі даних, різноманітність формулювань і нестабільність структури створюють бар'єри для машинного аналізу та порівняння вимог до кандидатів. У зв'язку з цим виникає необхідність розробки програмного рішення, здатного здійснювати автоматизований збір та обробку вакансій, із подальшим виокремленням блоку вимог до кандидатів. Використання мови програмування Python та її бібліотек у поєднанні з відкритим API створює технічні передумови для ефективної реалізації такого підходу як в HR-сфері, так і в контексті розвитку технології обробки природної мови та аналітики текстових даних.

Метою та основним завданням роботи є розробка програмного засобу для автоматизованого збору, структуризації та обробки описів вакансій з онлайн-ресурсу robota.ua, з акцентом на виокремлення блоку вимог до кандидатів. Реалізоване рішення повинно забезпечувати виявлення ключових текстових маркерів, фільтрацію та нормалізацію неструктурованих даних, а також збереження результатів у форматі, придатному для подальшої аналітики та статистичної обробки. Визначені найбільш затребуваних навичок, а також створенні на основі отриманих даних портрету ідеального кандидата на посаду системного аналітика.

Об'єкт дослідження – процеси автоматизованого збору, обробки та аналізу текстової інформації з онлайн-ресурсів, що містять опис вакансій.

Предмет дослідження – методи і програмні засоби виокремлення, структурування, збереження та аналізу блоку вимог до кандидатів із неструктурованих описів вакансій.

Методи дослідження. У роботі використано методи аналізу й синтезу для вивчення структури вакансій та принципів роботи з API, методи моделювання для формалізації підходу до виокремлення вимог, а також інструментальні методи розробки програмного забезпечення. Програмна реалізація здійснювалася із використанням мови Python та бібліотек requests, BeautifulSoup та pandas, що забезпечили доступ до веб-ресурсів, обробку HTML-контенту та структурування даних. Візуалізація отриманих даних за допомогою функціоналу програми Microsoft Power BI.

Наукова новизна отриманих результатів. Запропоновано новий підхід до автоматизованої обробки описів вакансій та аналізу й визначення найбільш затребуваних навичок, що базується на використанні текстових маркерів (ключових фраз) для точного виокремлення фрагментів, що містять вимоги до кандидатів. На відміну від більшості існуючих рішень, які передбачають жорстку залежність від конкретної структури або шаблонів сайту, запропонований підхід є гнучким і стійким до варіативності форматів подання вакансій, що особливо важливо для роботи з великими масивами даних із відкритих джерел.

Практичне значення отриманих результатів. Полягає у розробці універсального програмного засобу для автоматизованого збору, обробки та структурування текстової інформації з описів вакансій, що публікуються на онлайн-ресурсах. Створений інструмент дозволяє суттєво зменшити часові та трудові витрати, пов’язані з ручною обробкою неструктурованих даних, та забезпечити зручне представлення результатів у вигляді таблиць, придатних для подальшого аналізу або імпорту в аналітичні системи. А також виокремлені найбільш релевантних навичок для отримання посади системного аналітика.

РОЗДІЛ 1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ

1.1 Постановка проблеми

У процесі цифрової трансформації ринку праці дедалі більше компаній використовують онлайн-ресурси для публікацій вакансій. Попри широке розповсюдження спеціалізованих платформ таких як, robota.ua, work.ua, Jooble тощо. Структура описів вакансій залишається переважно неформалізованою. Це створює низку проблем як для шукачів роботи, так і для роботодавці, аналітиків та розробників інформаційних систем.

Опис кожної вакансії формується у довільній формі, а ключові елементи (вимоги до кандидата, обов’язки, умови праці) можуть мати різні назви, порядок розташування, стилі подачі або бути зовсім відсутніми. Внаслідок цього:

- ускладнюється автоматизоване структурування та фільтрація вакансій;
- неможливо проводити точний аналіз ринку праці за компетенціями;
- значна частина цінної інформації залишається невикористаною або втраченою внаслідок відсутності єдиного формату.

Наявні рішення з обробки вакансій мають надмірно вузьку адаптацію під окремі сайти. Також спостерігається брак відкритих універсальних інструментів, які дозволяли б ефективно виділяти фрагменти тексту з вимогами до кандидатів із великої кількості вакансій.

Таким чином, проблема полягає у відсутності ефективного механізму автоматичного виокремлення блоку вимог до кандидатів з описів вакансій, що розміщуються на онлайн-ресурсах у неструктурованому вигляді. Її вирішення потребує розробки відповідного програмного засобу, здатного здійснювати:

- автоматизований збір вакансій із відкритих джерел;

- ідентифікацію маркерів початку й завершення блоку вимог;
- збереження структурованих результатів у придатному для подальшого аналізу форматі.

1.2 Перспективи використання обробленої інформації вакансій у практичних задачах

Уніфіковані та очищенні дані про вакансії можуть бути використані для вирішення низки практичних завдань:

- Аналіз ринку праці: можливість визначення найпоширеніших вимог до певних посад, середнього рівня зарплат, географічного розподілу тощо.
- Інтеграція з системами кар’єрного консультування: формування списку навичок, які необхідно здобути для досягнення певної посади.
- Розробка рекомендаційної системи вакансій: на основі подібності текстів описів, ключових навичок та минулого досвіду кандидата.

Таким чином, автоматизований аналіз текстів вакансій не лише покращує процес пошуку роботи та підбору персоналу, а й створює передумови для застосування сучасних технологій аналізу тексту, машинного навчання та бізнес-аналізу в HR-сфері.

1.3 Значення висновків отриманих в результаті аналізу вимог до кандидатів на посаду системного аналітика

У процесі стрімкого розвитку IT-ринку, роль системного аналітика стає дедалі важливішою для забезпечення ефективної взаємодії між бізнесом та технічними командами. У зв’язку з цим, формування чітких та структурованих вимог до кандидатів у вакансіях набуває особливого значення як для роботодавців, так і для здобувачів вищої освіти.

Аналіз вимог, представлений у вакансіях на сайтах працевлаштування, дозволяє:

- Виділити ядро компетенцій, які очікуються від фахівців. Це як технічні навички (hard skills), так і особистісні (soft skills).
- Виявити актуальні методології, які використовуються в компаніях, що дозволяє підготувати більш релевантні навчальні програми та тренінги.
- Оцінити тенденції ринку праці – наприклад, наскільки часто вимагається знання англійської мови, досвід роботи з ERP/CRM системами, або роботи з BI-інструментами.

Для здобувачів така інформація дасть змогу об’єктивно оцінити власну відповідність очікувань ринку, окреслити вектори професійного розвитку, а також оптимізувати подання себе у резюме чи на співбесіді. Для освітніх закладів результати подібного аналізу можуть використані для оновлення програм підготовки фахівців, адаптації навчальних курсів до реальних потреб галузі.

Таким чином системний аналіз змісту вакансій виступає ефективним інструментом зворотного зв’язку між ринком праці та всіма учасниками професійної екосистеми – роботодавцями, кандидатами та закладами освіти.

1.4 Огляд літератури та загальних відомостей

1.4.1 HTTP

HTTP – протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від Hypertext Transfer Protocol, протокол передачі гіпертекстових документів [2]. Основним призначенням протоколу HTTP є передача веб-сторінок (текстових файлів з розміткою HTML, зображень та застосунків), проте за його допомогою успішно передаються й інші файли (в цьому плані HTTP складає конкуренцію складнішому FTP) [3].

До основних властивостей протоколу HTTP відносяться:

- Клієнт-серверна модель: протокол HTTP функціонує відповідно до концепції взаємодії клієнта і сервера, де клієнтська сторона (наприклад, веб-браузер) ініціює запит, а серверна сторона обробляє його та повертає відповідь.
- Безстанова природа: HTTP не зберігає інформацію про попередні запити. Кожна HTTP-транзакція є незалежною від попередніх, що спрощує її реалізаціє, але потребує додаткових механізмів для реалізації сесійної взаємодії.
- Гнучкість та розширюваність: HTTP підтримує низку стандартних методів взаємодії (таких як GET, POST, PUT, DELETE тощо) і забезпечує можливість додавання нових функціональних можливостей через заголовки, що робить його придатним для широкого спектра задач у веб-середовищі.

Також протокол HTTP має захищене розширення HTTPS. HTTPS (HyperText Transfer Protocol) – це розширення HTTP, яке забезпечує шифрування даних між клієнтом і сервером за допомогою протоколів TLS або SSL. Це гарантує конфіденційність та цілісність переданих даних, захищаючи їх від перехоплення та змін [4].

1.4.2 Swagger

Swagger – це набір інструментів для розробки, документування та тестування RESTful API, який базується на специфікації OpenAPI. Він дозволяє описувати структуру API у форматах YAML або JSON, що робить її зрозумілою як для людей так і для машин. [5]

Використання Swagger має багато переваг для кінцевих користувачів – зокрема для тестувальників, бізнес-аналітиків, інтеграторів та зовнішніх клієнтів, які споживають API. Він має зрозумілий інтерфейс для взаємодії з API. Swagger UI забезпечує зручну веб-форму, де кожна кінцева точка API

(endpoint) подається у вигляді документа з чітко зазначеними параметрами, типами даних, очікуваними відповідями та можливими кодами помилок. Така подача полегшує ознайомлення з функціональністю сервісу навіть для осіб без глибоких знань у галузі програмування [6].

Також однією з переваг є інтерактивне тестування яке дозволяє користувачам взаємодіяти з API безпосередньо через браузер, надсилаючи запити й отримуючи відповіді в реальному часі. Це усуває потребу у використанні додаткових інструментів та прискорює процес перевірки функціональності API.

Оскільки кожен параметр, формат відповіді та можливі сценарії помилок чітко описані в документації, користувачі мають змогу коректо реалізувати взаємодію з API, не покладаючись на припущення. Це підвищує точність інтеграції і зменшує потребу в технічній підтримці.

Swagger виступає як самодокументована платформа – нові члени команди або зовнішні партнери можуть швидко ознайомитись з можливостями API без потреби у супровідній технічній документації або консультації зі сторони розробників. Навіть до фактичної інтеграції користувачі можуть оцінити функціональні можливості API, протестувати приклади запитів, переглянути відповіді та зрозуміти структуру даних, що істотно прокрашує планування розробки клієнтської частини.

1.4.3 Python

Python є високорівневою, інтерпретованою мовою програмування загального призначення, яка здобула широку популярність у різних сферах розробки програмного забезпечення, зокрема в аналізі даних, машинному навчанні, веб-розробці та автоматизації [7-9]. Її синтаксис наближений до природної мови, що спрощує написання, читання та супровід коду. Окрім цього Python підтримує кілька парадигм програмування, включаючи процедурну, об’єкто-орієнтовану (ООП), та функціональну, що робить її

універсальним інструментом у реалізації складних програмних рішень. Має широкі можливості для розв'язання як прикладних, так і науково-технічних завдань [10-12]. Мова Python має велику екосистему бібліотек та фреймворків, що є однією з визначальних переваг. [13]

Для вирішення розглянутих завдань особливе значення мають такі модулі, як requests, BeautifulSoup, time та pandas, кожен із яких виконує специфічну функцію в процесі збору, обробки й аналізу інформації з відкритих джерел [14].

1.4.4 Використання бібліотеки requests

Бібліотека requests – це потужний інструмент для надсилання HTTP запитів. Вона забезпечує простий та інтуїтивно зрозумілий інтерфейс для відправки запитів до API та отримання відповідей у форматах JSON або HTML, дозволяючи розробникам зосередитися на логіці програми, а не на деталях реалізації HTTP-протоколу. Бібліотека підтримує всі основні HTTP-методи (GET, POST, PUT, DELETE, HEAD, OPTIONS та PATCH). Зокрема у межах даної роботи за допомогою requests реалізовано отримання структурованих даних про вакансії через REST API robota.ua [15].

Передача параметрів у запиті може здійснюватися через аргумент params, який формує відповідні ключі й значення в URL-рядку. Для надсилання даних у тілі запиту використовуються аргументи data або json, що забезпечує підтримку як традиційного формату веб-форм, так і сучасного формату обміну даними JSON. [16] Крім того, бібліотека надає можливості роботи з HTTP-заголовками за допомогою параметра headers, що дозволяє налаштовувати ідентифікацію, формат прийнятих даних та інші характеристики з'єднання. [17] Для забезпечення контролю над станом виконаного запиту та обробки помилок використовуються відповідні механізми: атрибут status_code дозволяє оцінити результат запиту, а метод

raise_for_status() – перехопити виняткові ситуації у разі виникнення помилок з боку сервера або клієнта [18].

Бібліотека requests здобула широку популярність серед розробників завдяки ряду переваг. Насамперед, вона вирізняється простим та логічним синтаксисом, що значно спрощує реалізацію HTTP-запитів. Окрім цього, requests забезпечує високу гнучкість, дозволяючи легко налаштовувати параметри запитів та обробляти відповіді сервера [19]. Ця бібліотека має розвинену спільноту користувачів і добре задокументована, що сприяє швидкому вирішенню можливих проблем під час її використання. Вона сумісна з багатьма версіями Python, а також добре інтегрована з іншими популярними інструментами, зокрема BeautifulSoup, що робить її ефективним рішенням для автоматизованого збору даних із веб-ресурсів [20].

1.4.5 BeautifulSoup

У межах даної кваліфікаційної роботи для обробки HTML-даних, отриманих у результаті запитів до API платформи пошуку роботи, використовується бібліотека BeautifulSoup, яка є одним із найбільш поширених засобів синтаксичного аналізу (парсингу) HTML та XML-документів у Python. Основною метою її застосування є трансформація неструктурованого HTML-контенту в логічно організовану структуру, придатну для подальшої автоматизованої обробки, аналізу й виділення семантично значущих фрагментів [21].

Метод get_text у бібліотеці BeautifulSoup використовується для отримання текстового HTML-документа, із повним ігноруванням тегів розмітки. Його застосуванням дозволяє отримати «чистий» текстовий контент, що міститься між відкриваючими і закриваючими тегами, включаючи текст усіх вкладених дочірніх елементів [22].

У практиці обробки веб-даних цей метод є надзвичайно корисним, зокрема для задач попередньої обробки текстів, коли HTML-структура не несе

додаткового смислового навантаження. Метод get_text() підтримує параметри налаштування форматування виводу, наприклад, separator, який дозволяє вставляти заданий символ (зазвичай пробіл або перенесення рядка) між частинами тексту, що витягуються з різних елементів [23].

У межах даної кваліфікаційної роботи метод get_text(separator=" ") застосовується з метою лінійного об’єднання всіх текстових вузлів документа в один суцільний фрагмент тексту. Це забезпечує можливість подальшого уніфікованого аналізу вмісту вакансій, зокрема пошук ключових слів та виокремлення фрагментів, що містять вимоги до кандидатів.

1.4.6 Модуль time

Модуль time є частиною бібліотеки Python і забезпечує інтерфейс для доступу до функцій, пов’язаних із часом. Цей модуль дозволяє виконувати різноманітні операції, такі як вимірювання часу виконання програми, форматування часу, затримка у виконанні коду та інші [24]. Для використання функціональності модуля time необхідно його імпортувати. Функція time.sleep() призначена для призупинення виконання програми на заданий проміжок часу який виражений у секундах [25]. Її застосування є доцільним у випадках, коли наявна потреба у контролі частоти звернень до зовнішніх сервісів або імітувати паузи у виконанні певних дій.

Завдяки своїй універсальності та простоті використання, модуль time широко використовується в автоматизованих скриптах, логуванні подій, тестуванні та багатьох інших сферах, де необхідна робота з часовими інтервалами.

В даній роботі модуль time використовується для регулювання частоти звернень до API-сервера robota.ua з метою уникнення блокування за надмірну активність. Зокрема, функція time.sleep() дозволяє реалізувати паузи між послідовними HTTP-запитами, забезпечуючи стабільність роботи крипта та дотримання етичних стандартів веб-взаємодії.

1.4.7 Бібліотека pandas

Бібліотека pandas є інструментом з відкритим вихідним кодом, який розроблений для обробки та аналізу даних у середовищі програмування Python. Вона забезпечує зручні у використанні та продуктивні структури даних, зокрема Series (одновимірні масив) та DataFrame (двовимірні таблиці) [26]. В роботі програми використовуються саме DataFrame, що дозволяє ефективно працювати з табличною інформацією за аналогією до баз даних SQL або електронних таблиць Excel та Google Sheets [27-28]. Її інструментарій охоплює багато різних можливостей, але для нашої програми головною є можливість інтеграцію з різними форматами даних, що забезпечує зчитування та збереження інформації у форматах CSV, Excel, JSON, SQL тощо [29].

1.4.8 Бібліотека re

У наукових дослідженнях, пов’язаних із обробкою текстових даних, модуль re мови програмування Python є ключовим інструментом для реалізації операцій з регулярними виразами. Цей модуль забезпечує потужні засоби для пошуку, зіставлення та маніпуляції рядками на основі заданих шаблонів [30].

Модуль re дозволяє компілювати регулярні вирази в об’єкти шаблонів, які потім можна використовувати для пошуку відповідностей у рядках, розділення рядків за шаблоном або заміни частини рядка [31].

У контексті аналізу вакансій, модуль re є незамінним для виділення ключових частин тексту, таких як вимоги до кандидатів або обов’язки. Наприклад, можна використовувати регулярні вирази для пошуку фраз, що починається з «вимоги», «необхідні навички» або «обов’язки», і виділення відповідних блоків тексту. Це дозволяє автоматизувати процес аналізу великої кількості вакансій та структурувати інформацію для подальшого аналізу [32].

1.4.9 Бібліотека googletrans

Бібліотека googletrans є неофіційним Python-інтерфейсом до веб-сервісу Google Translate, який забезпечує можливість здійснення автоматизованого перекладу текстової інформації між понад 100 мовами без необхідності використання API-ключа або аутентифікації. Вона базується на зворотному інжинірингу клієнтських запитів до Google Translate та функціонує через HTTP-запити з подальшим розбором відповідей [33].

Серед основних функціональних можливостей бібліотеки варто відзначити:

- автоматичне визначення мови джерела за допомогою вбудованого механізму ідентифікації;
- переклад текстових рядків будь-якої довжини або списків рядків на задану мову;
- підтримка пакетної обробки запитів;

В рамках обробки вакансій бібліотека googletrans використовується для автоматичного перекладу описів посадових вимог з російської, англійської або інших мов на українську. Такий підхід забезпечує уніфікацію мовного середовища в структурованих даних та підвищує ефективність подальшого лінгвістичного аналізу. У поєднанні з модулем langdetect, бібліотека дозволяє виконувати переклад лише в тих випадках, коли мова оригіналу відрізняється від української, що значно оптимізує обчислювальні ресурси при обробці великих обсягів текстових даних [34].

1.4.10 Бібліотека langdetect

Бібліотека langdetect є Python-портом Java-бібліотеки для визначення мови, розробленої Шуйо Накадамі (Shuyo Nakatani) в рамках проєкту Google. Вона реалізує метод наївного байєсівського класифікатора, що ґрунтується на аналізі n-грам (n-грамою називається послідовність з n послідовних символів

у тексті), для ідентифікації мови тексту. Бібліотека підтримує 55 мов, включаючи українську, англійську, російську, французьку та інші [35].

Дана бібліотека широко використовується в задачах обробки природної мови (NLP), зокрема для:

- Попередньої обробки текстових даних (автоматичне визначення мови тексту перед подальшою обробкою або аналізом)
- Фільтрація контенту (відбір текстів певною мовою з багатомовних джерел).
- Інтеграція з іншими інструментами.

Функціонал який надає бібліотека langdetect є ефективним інструментом для автоматичного визначення мови тексту в Python. Її простота використання та підтримки широкого спектра мов роблять її корисною в багатьох задачах обробки текстових даних.

1.4.11 Power BI

Power BI – це потужна система бізнес-аналітики від компанії Microsoft, що забезпечує повний цикл роботи з даними: від їхнього імпорту та попередньої обробки до візуалізації та спільного доступу до звітів. Інструмент орієнтований на потреби користувачів різного рівня технічної підготовки – від аналітиків до управлінців, що дозволяє створювати динамічні інформаційні панелі (дашборди) для моніторингу ключових показників ефективності (KPI) та виявлення закономірностей у даних [36].

Серед основних функцій Power BI варто виділити:

- Інтеграція з численними джерелами даних, включаючи бази даних (SQL Server, MySQL, Oracle), хмарні сервіси (Azure, Salesforce, Google Analytics), текстові та табличні формати (CSV, Excel).
- Очищення та трансформація даних за допомогою Power Query – графічного редактора ETL-процесів.

- Побудову аналітичних моделей з використанням мови DAX для створення обчислюваних стовпців, міри і KPI.
- Створення інтерактивних звітів з великою кількістю візуальних елементів (графіки, карти, таблиці, фільтри тощо).
- Публікацію та спільний доступ до звітів через Power BI Service або вбудовування в сторонні застосунки та веб-сайти.
- Підтримка оновлення даних у реальному часі та планових автоматичних оновлень [37].

Power BI активно застосовується в аналітичній діяльності підприємств різних галузей – фінансовому аналізі, контролінгу, маркетингових досліджень, прогнозуванні продажів, цифрових трансформацій бізнесу тощо. Його використання сприяє підвищенню прозорості процесів, обґрунтованості прийнятих рішень та формуванню аналітичної культури в організації [38].

Завдяки поєднанню широкого функціоналу, інтуїтивно зрозумілого інтерфейсу та гнучких можливостей розгортання Power BI залишається одним із найпопулярніших інструментів у сфері бізнес-аналітики та систем підтримки прийняття рішень [39].

1.4.12 Power Query

Power Query – це сучасний інструмент для побудови, трансформацій та завантаження даних, який інтегровано до середовища Microsoft Power BI та Microsoft Excel. Основною перевагою Power Query є можливість обробки неструктурованими та слабо структурованих даних без необхідності володіння мовами програмування. [40]

Завдяки підтримці великої кількості джерел даних (локальні файли, бази даних, веб-ресурси, хмарні сервіси тощо) та широкому набору функцій для трансформації, Power Query дозволяє здійснювати попереднє очищення, фільтрацію, нормалізацію, об’єднання та агрегування даних. Усі операції

здійснюються послідовно у вигляді запитів, що робить процес повністю автоматизованим та відтворюваним. [41]

У рамках виконання даного дослідження Power Query було використано для завантаження та попередньої обробки інформації з Excel-файлів «system_analyst_data.xlsx» та «keyword_frequency_results.xlsx». За допомогою цього інструменту здійснювалось очищення від зайвої інформації, а також зміна типу даних.

1.5 Методологія обробки та структура тексту вакансій

Під час аналізу вакансій ключовим завданням стало виявлення блоку, що містить вимоги до кандидата. Для цього було використано підхід на основі ключових маркерів – характерних фраз, які найчастіше вказують на початок або кінець релевантного фрагмента опису. Серед прикладів таких маркерів – «вимоги», «необхідні навички», «qualifications» «must have», а також «ми пропонуємо», «умови», які позначають межу завершення цього блоку.

Однією з головних труднощів при автоматизації обробки інформації про вакансії є відсутність єдиної стандартної структури їх опису. Тексти вакансій формуються вручну HR-фахівцями або рекрутерами, що призводить до значної варіативності у форматах подачі інформації, формулюваннях та навіть мовних конструкціях.

У більшості випадків вакансії містять такі логічні блоки:

- загальна інформація про компанію;
- обов’язки на посаді;
- вимоги до кандидата;
- умови праці;

- мотиваційні блоки (наприклад, «чому варто обрати саме нас»)

Однак порядок, назви розділів, а подекуди й наявність самих блоків – варіюються. Зокрема, блок вимоги може позначатися як «Вимоги», «Необхідні навички», «Required», «Skills», або ж бути вбудованими у загальний опис без окремого заголовка.

Для вирішення цієї проблеми у роботі застосовано підхід, що базується на виділенні маркерів початку та завершення змістовних блоків. Такий підхід дозволяє сформувати уніфіковану структуру для подальшої роботи з даними.

1.6 Автоматизоване збирання та збереження даних

Збір даних у даній роботі реалізовується шляхом надсилання запитів до відкритого API сайту rabota.ua. За допомогою requests.post() виконується пошук вакансій за заданими параметрами (наприклад, ключове слово «system analyst»), після чого для кожної вакансії виконується додатковий запит requests.get(), який дозволяє отримати повний HTML-опис.

Після вилучення й обробки описів інформація зберігається у структурованому форматі в Excel-файлі з кількома аркушами:

- список вакансій;
- список міст;
- список компаній.

Цей підхід дозволяє не лише проводити подальший аналіз, а й легко інтегрувати дані у бази даних чи BI-системи (наприклад, Power BI).

Висновки до першого розділу

У першому розділі кваліфікаційної роботи було розглянуто теоретичні основи та методологічні підходи до задачі автоматизованої обробки текстової інформації з онлайн-платформ з працевлаштування. Встановлено, що опис

вакансій на таких платформах, як robota.ua, work.ua, Jooble, часто не має уніфікованої структури, що істотно ускладнює їх автоматичну обробку, структуризацію та подальший аналіз.

Проведено аналіз наукових і технічних джерел щодо принципів функціонування HTTP-протоколу, інструментів взаємодії з API (Swagger), а також розглянуто сучасні програмні засоби на базі мови Python, зокрема бібліотеки requests, BeautifulSoup, time, pandas, що є ключовими у побудові системи збору та фільтрації вакансій. Особливу увагу було приділено можливостям кожної бібліотеки у контексті вирішення завдань веб-скрейпінгу, попередньої обробки HTTP-документів, вилучення релевантних фрагментів тексту та збереження результатів у структурованому вигляді.

У ході роботи було обґрунтовано використані підходи, що базуються на ключових маркерах – фразах, які дозволяють виявити межі блоку з вимогами до кандидатів. Такий підхід забезпечує можливість формування уніфікованої таблиці вакансій, придатних для подальшого аналізу та побудови статистичних моделей.

Також у межах розділу описано етапи автоматичного збору даних через API, логіку обробки HTML-вмісту вакансій, а також формат збереження даних у Excel-файлі для подальшої аналітики. Розглянуто перспективні напрями застосування оброблених даних – зокрема, в автоматизованому створенні профілів кандидатів, моніторингу ринку праці, інтеграції з системами кар’єрного консультування та побудови інтелектуальних сервісів для пошуку вакансій. Також було окреслено практичну користь для роботодавців, здобувачів освіти та навчальних закладів.

Таким чином, результати, викладені у першому розділі, підтверджують актуальність обраної теми та створюють необхідне теоретико-практичне підґрунтя для реалізації прикладного програмного рішення, що буде розглянуто у наступних розділах роботи.

РОЗДІЛ 2 СПЕЦІАЛЬНИЙ

2.1 Отримання даних для подальшої аналітики

Для аналізу ринку вакансій та вимог роботодавців для претендентів на посади системних аналітиків необхідно отримати актуальні дані із вакансій з сайтів для працевлаштування.

Було обрано сайт https://robota.ua оскільки він має найбільшу базу вакансій на посаду системного аналітика. Також цей сайт дає можливість у вільному доступі отримувати інформацію з бази даних та обробляти її згідно власних бажань та потреб.

Необхідно отримати таку інформацію: вакансії системних аналітиків в Україні. Вакансія має містити: унікальний ключ id, назву позиції, опис вакансії (відомості: про компанію, вимоги до кандидатів, умови праці та додаткові переваги).

Роблячи певні запити да бази даних https://robota.ua можна отримати дані які необхідні для проведення аналізу. Використовуємо лише ресурс robota.ua, оскільки майже всі вакансії також дублюються роботодавцями на інших платформах. І тому для уникнення повторів зосередимо свою увагу на одному з найбільш популярних сайтів для пошуку роботи.

Усю інформацію (Рис. 2.1) для створення запитів до бази даних https://robota.ua можна отримати за посиланням: https://api.robota.ua. Можна отримати різні дані від бази даних сайту robota.ua, в залежності від створених HTTP запитів. HTTP запити можуть мати різні методи для передачі інформації між сервером та клієнтом: GET, POST, PUT, DELETE, OPTIONS, CONNECT, TRACE та інші. Також HTTP запит підтримує передачу для вибірки деяких параметрів лише потрібну інформації з бази даних. Здебільшого параметри, які використовуються у запитах, можна знайти у документації Swagger. В наслідок чого для отримання вакансій, інформації та вимог до кандидатів що міститься в них було необхідно дослідити можливості Swagger та HTTP методів і параметрів, які можна передавати при запиті в базу даних.

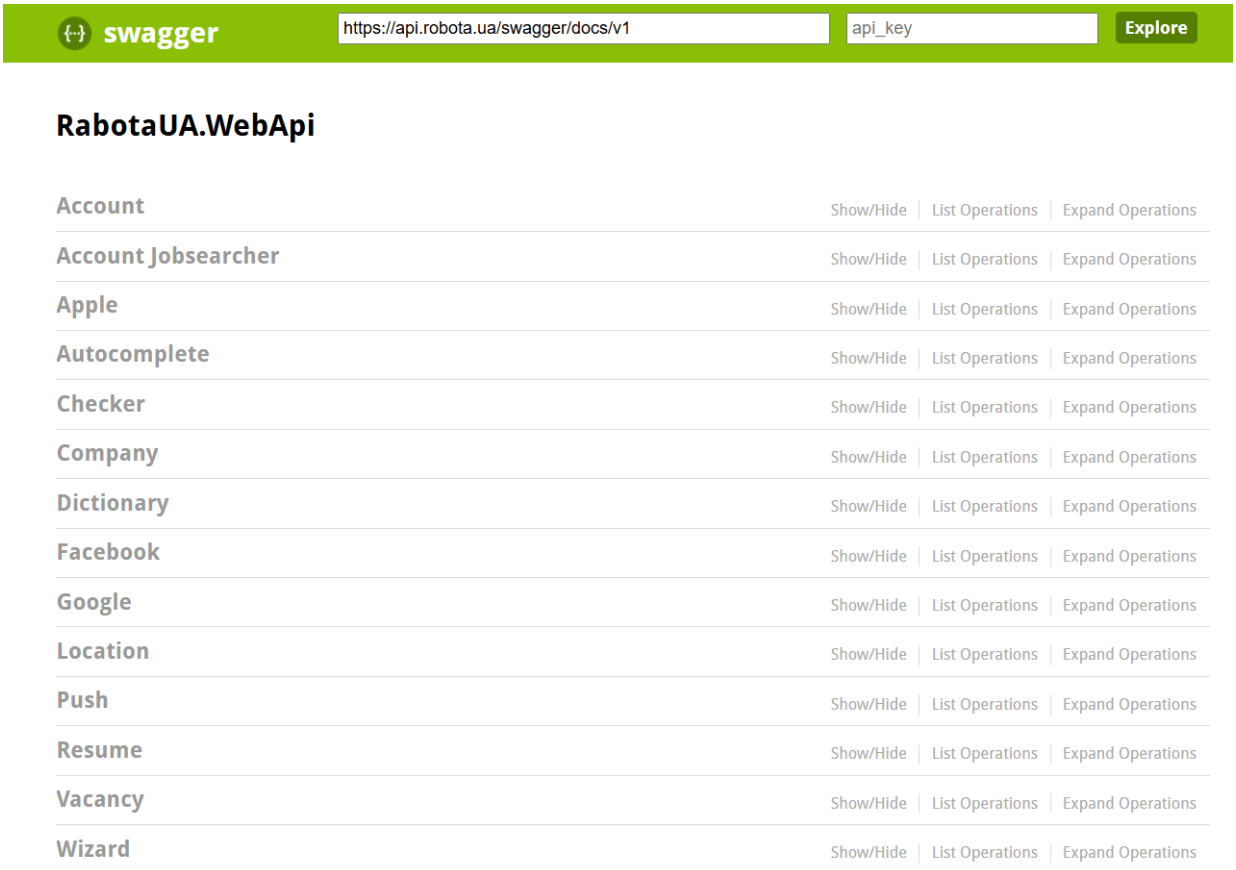


Рисунок 2.1 – Можливі запити на сервер https://robota.ua для отримання інформації

2.2 Автоматизоване отримання даних про вакансії

Для отримання вакансій на посаду системного аналітика в Україні здійснимо пошук за ключовою фразою «system analyst» створивши запит на api.robota.ua/vacancy/search (Рис. 2.2, 2.3). Вказавши необхідні параметри запиту.

При виконанні запиту на сервер зображеного на рисунках 2.2 та 2.3 повертається масив розміром 20 вакансій с коротким їх описом. У відповіді від серверу також міститься інформація про кількість всіх вакансій про працевлаштування в Україні за ключовим словом «system analyst» у полі «total» (Рис. 2.4).

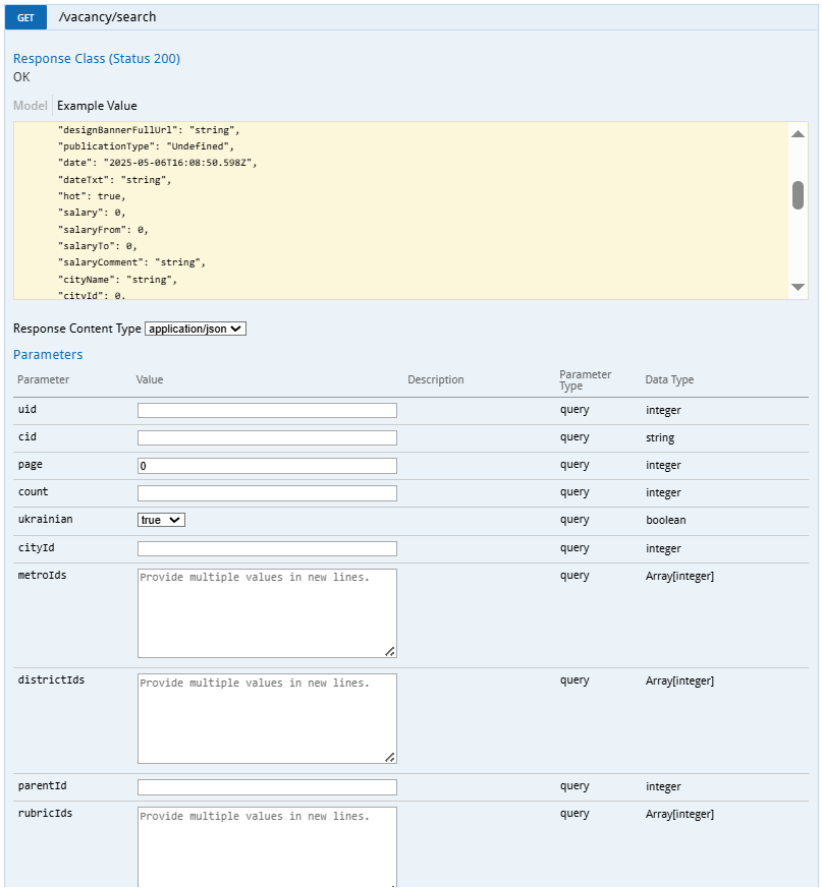


Рисунок 2.2 – Запит на отримання вакансій в Україні за ключовою фразою «system analyst»

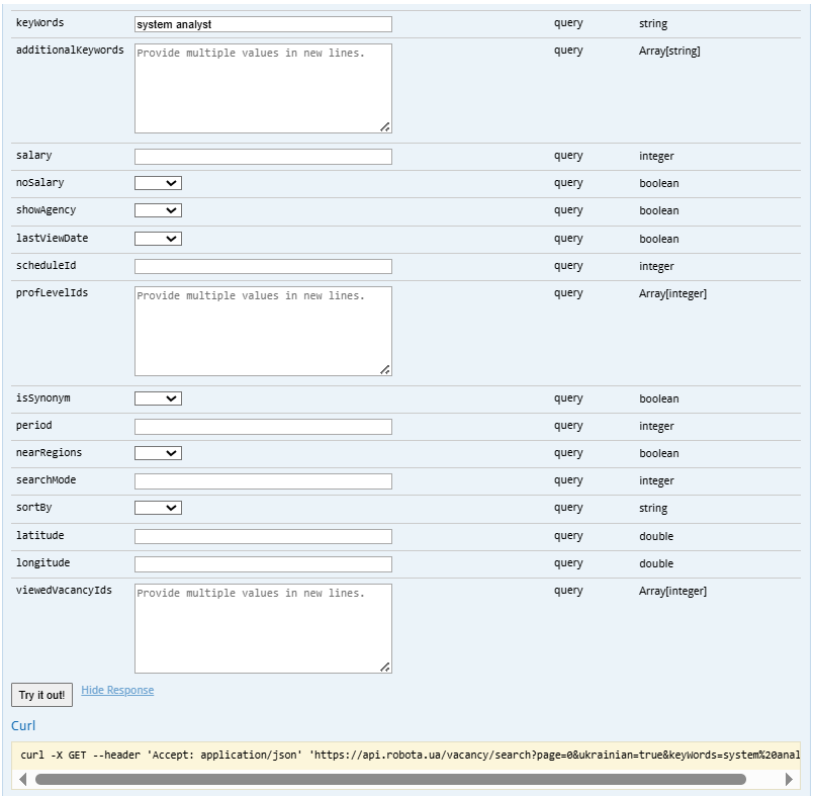


Рисунок 2.3 – Запит на отримання вакансій в Україні за ключовою фразою «system analyst»

```
{
  "took": 116,
  "start": 0,
  "count": 20,
  "total": 387,
  "hasGeoCount": 376,
  "errorMessage": null,
  "documents": [
    {
      "id": 10517374,
      "name": "System Analyst (Data Management)",
      "logo": "3235793_20250108124834.png",
      "designBannerUrl": "",
      "designBannerFullUrl": "",
      "publicationType": 2,
      "date": "2025-05-06T17:01:00.24",
      "dateTxt": "",
      "hot": true,
      "salary": 0,
      "salaryFrom": 0,
      "salaryTo": 0,
      "salaryComment": "",
      "cityName": "Київ",
      "cityId": 1,
      "metroName": "Дорогожичі",
      "metroId": 49,
      "districtName": "Шевченківський",
      "districtId": 10,
      "notebookId": 3235793,
      "companyName": "Ajax Systems",
      "formApplyCustomUrl": "https://jobs.lever.co/ajax/3c4160df-e478-4ab9-9723-
f00c2ff5a69d",
      "isLiked": false,
      "isApply": false,
      "applyDate": null,
      "lastViewDate": null,
      "distanceText": "",
      "latitude": 50.470078,
      "longitude": 30.4633,
      "shortDescription": "\t\t\t\t\t\t\t\t\t\tAjax Systems is a global technology company
and the leading developer and manufacturer of Ajax security systems with smart home
capabilities in Europe. It encompasses a comprehensive ecosystem featuring 180 devices,
mobile and desktop appli…",
      "badges": [
        {
          "id": 4047517,
          "name": "Офіс з генератором"
        },
        {
          "id": 4047518,
          "name": "Офіс з безперебійним інтернетом"
        },
        {
          "id": 4047519,
          "name": "Медичне страхування"
        },
        {
          "id": 4047520,
          "name": "Курси іноземної мови"
        }
      ]
```


Рисунок 2.4 – Частина відповіді з сервера на запит пошуку вакансій за ключовою фразою «system analyst»

Request URL з інформацією яку ми отримали у відповідь на наш запит до серверу: https://api.robota.ua/vacancy/search?page=0&ukrainian=true&keyWords=system%20analyst.

У межах цієї роботи мова Python була обрана як основний інструмент розробки завдяки своїй універсальності, зручності у роботі з API, структурованими даними, а також доступності бібліотек, що спрощують автоматизацію обробки інформації з веб-джерел.

Ми отримаємо по 20 вакансій з кожної сторінки, отже ми маємо передбачити автоматизований збір інформації з усіх сторінок при написанні програмного коду (Рис. 2.5).

```
# We get 20 vacancies from each page. The number of pages is determined dynamically
search_url = "https://api.robota.ua/vacancy/search?page=0&ukrainian=true&keyWords=system%20analyst"
vacancy_url = "https://api.robota.ua/vacancy"
headers = {"Accept": "application/json"}
payload = {
    "keyWords": "system analyst",
    "page": 0,
    "count": 20  # Number of vacancies on the page
}
```

Рисунок 2.5 – Скрипт для надсилання запиту на https://api.rabota.ua/vacancy/search з ключовою фразою «system analyst»

З поля «total» ми знаємо, що наразі наявні 387 вакансій, тому 387/20 = 19.35, тобто 20 сторінок має переглянути и зберегти програмний алгоритм наразі, але також слід не забувати, що ринок вакансій динамічний та постійно змінюється. В наслідок чого слід передбачити автоматичний перегляд усіх наявних сторінок з вакансіями в момент спрацювання алгоритму (Рис. 2.6).

Для цього нам слід імпортувати бібліотеку requests у середовище Python. Request не входить до стандартної бібліотеки Python, тому перед використанням її необхідно встановити за допомогою менеджера пакетів pip (Рис. 2.7). Після встановлення слід імпортувати бібліотеку в скрипт програми (Рис. 2.8).

```python
# The algorithm looks at all available pages
while True:
    payload["page"] = page
    response = requests.post(search_url, json=payload, headers=headers)

    # If there is an error, stop collecting
    if response.status_code != 200:
        print(f"Error on page {page}")
        break

    data = response.json()

    # If there are no vacancies, complete the cycle
    docs = data.get("documents", [])
    if not docs:
        break
```

Рисунок 2.6 – Скрипт для перегляду усіх наявних сторінок

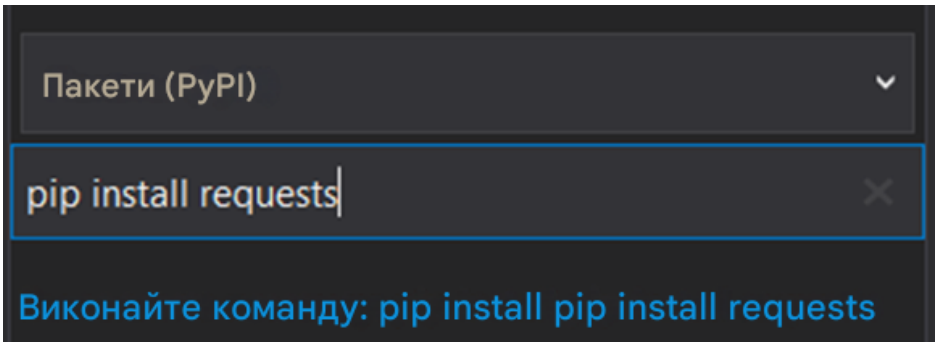


Рисунок 2.7 – Встановлення бібліотеки requests за допомогою менеджера пакетів pip

```python
import requests
```

Рисунок 2.8 – Імпорт бібліотеки requests в середовище програми

На Рис. 2.4 можна побачити, що в полі «description» повертається лише частина опису вакансії. Для отримання повного опису кожної вакансії Для повного опису потрібен окремий GET-запит на {vacancy_url}?id={vac_id} (Рис. 2.9).

```python
# Processing each job from the current page
for vac in docs:
    vac_id = vac.get("id")

    # The “description” field contains only a short description. For a full description, a separate GET request is required.
    full_response = requests.get(f"{vacancy_url}?id={vac_id}", headers=headers)
    full_description = full_response.json().get("description", "") if full_response.status_code == 200 else ""

    # Get the city and company name
    city = vac.get("cityName", "").strip()
    company = vac.get("companyName", "").strip()

    # Assign a unique city_id
    if city not in cities:
        cities[city] = city_counter
        city_counter += 1
    city_id = cities[city]

    # Assign a unique company_id
    if company not in companies:
        companies[company] = company_counter
        company_counter += 1
    company_id = companies[company]

    # Add the vacancy to the general list
    vacancies.append({
        "vacancy_id": vac_id,
        "title": vac.get("name"),
        "salaryFrom": vac.get("salaryFrom"),
        "salaryTo": vac.get("salaryTo"),
        "salaryComment": vac.get("salaryComment"),
        "date": vac.get("date"),
        "url": vac.get("formApplyCustomUrl"),
        "shortDescription": vac.get("shortDescription"),
        "fullDescription": full_description.replace("\n", " ").replace("\r", " "),
        "company_id": company_id,
        "city_id": city_id,
        "badges": ", ".join([b["name"] for b in vac.get("badges", [])])
    })

    # Small delay between requests to avoid API blocking
    time.sleep(0.2)
```

Рисунок 2.9 – Алгоритм перегляду та збереження кожної вакансії на сторінки.

За для уникнення блокування з боку API реалізовуємо штучну затримку за допомогою функції time.sleep(). Затримка тривалістю 0,2 секунди виконується після кожного запиту до окремої вакансії, що дозволяє знизити навантаження на сервер та уникнути блокування. Додаткова затримка у 0,5 секунди реалізована після завершення обробки всієї сторінки, що також сприяє контролю частоти запитів та дотримання етичних принципів взаємодії з веб-ресурсам. Виведення повідомлення про успішну обробку (print(…)) дозволяє відслідковувати поточний прогрес виконання скрипту (Рис. 2.10).

```python
    print(f"Page {page + 1} loaded")
    page += 1
    time.sleep(0.5)   # Pause between pages
```

Рисунок 2.10 – Алгоритм для відслідковування прогресу виконання програми та затримка між сторінками

Отриману інформацію будемо зберігати в форматі .xlsx для легкого та зручного перегляду в програмі Excel або Google Sheets та можливості подальшого імпортування до інших програм (Рис. 2.11). Для цього нам знадобиться функціонал бібліотеки pandas яку ми імпортуємо до нашого середовища на самому початку коду програми під скороченою для швидшого написання та покращення читабельності коду назвою pd, яка є загально прийнятим псевдонімом бібліотеки.

```
# Saving to a single .xlsx file for convenient viewing in Excel or importing into a database or Power BI
with pd.ExcelWriter("system_analyst_data.xlsx", engine="openpyxl") as writer:
    df_vacancies.to_excel(writer, sheet_name="Vacancies", index=False)
    df_cities.to_excel(writer, sheet_name="Cities", index=False)
    df_companies.to_excel(writer, sheet_name="Companies", index=False)

print("The file system_analyst_data.xlsx was successfully created!")
```

Рисунок 2.11 – Скрипт збереження в файл .xlsx для зручного перегляду в Excel або імпорту в базу даних чи Power BI

Оскільки передбачується можливість подальшого імпортування до баз даних, Microsoft Power BI слід під час зберігання привести дані до третьої нормальної форми. Тому передбачено створення окремих таблиці міст та компанія вигляду унікальний id ключ – значення (Рис. 2.12 та Рис. 2.13).

```
vacancies = []          # List of all vacancies
cities = {}             # Dictionary of unique cities
companies = {}          # Dictionary of unique companies
page = 0
city_counter = 1
company_counter = 1
```

Рисунок 2.12 – Створюємо додаткові словники для міст та компаній

```
# Convert to dataframes
df_vacancies = pd.DataFrame(vacancies)
df_cities = pd.DataFrame([{"city_id": cid, "city_name": name} for name, cid in cities.items()])
df_companies = pd.DataFrame([{"company_id": cid, "company_name": name} for name, cid in companies.items()])
```

Рисунок 2.13 – Перетворюємо на dataframes для подальшого зберігання в файл .xlsx

Також передбачено перевірку на дублювання вакансій (Рис 2.14, 2.15). Оскільки API може випадково повернути двічі вакансію з одним тим

самим id кодом, і в такому випадку буде порушено третю нормальну форму. А також зайвий допис у таблиці, який хоч і не суттєво, але вплине на загальну статистику.

```
seen_ids = set()      # Setting up tracking for already viewed job IDs
```

Рисунок 2.14 – Створюємо список з id вже переглянутих вакансій

```
# Skip duplicates
if vac_id in seen_ids:
    continue
seen_ids.add(vac_id)
```

Рисунок 2.15 – Скрипт який перевіряє за id чи не було до цього зчитано вакансію

Після цього ми отримаємо файл system_analyst_data.xlsx з трьома сторінками «Companies» (Таблиця 2.1), «Cities» (Таблиця 2.2) та «Vacancies» (Таблиця 2.3). Який зручно відкрити за допомогою програми Microsoft Excel.

Таблиця 2.1

Фрагмент переліку компанії які розмістили вакансії на посаду системного аналітика

company_id	company_name
-1-	**-2-**
1	CRM.UA
2	ПриватБанк
3	SkyUp Airlines
4	Linkos Group
5	Ajax Systems
6	ВІПЦЕМ, ПрАТ
7	ВАША ТАРА
8	КЛІРИНГОВИЙ ДІМ, АБ
9	TechExpert
10	Нова пошта
11	Укренерго, НЕК

Продовження табл. 2.1

-1-	-2-
12	Smart Solutions
13	ЮРіЯ-ФАРМ, Фармацевтична група компаній
14	SI BIS
15	Омега, ТОВ
16	DNIPRO-M, ООО
17	Укрексімбанк, АТ
18	КРЕДОБАНК, АТ
19	EY (Ernst & Young)
20	Група Данон: Данон Україна та Нутриція Україна
21	МТСБУ
22	Оператор ГТС України, ТОВ
23	Альфа Люкс, ПП
24	Перший Український Міжнародний Банк, АТ / ПУМБ
25	Салтівський м’ясокомбінат

Таблиця 2.2

Перелік міст в яких є відкриті вакансії на посаду системного аналітика

city_id	city_name
-1-	**-2-**
1	Харків
2	Київ
3	Львів
4	Одеса
5	Дніпро
6	Степанці
7	Запоріжжя
8	Полтава
9	Софіївська Борщагівка
10	Вінниця
11	Інші країни
12	Черкаси

Продовження табл. 2.1

-1-	-2-
13	Житомир
14	Шептицький
15	Тернопіль

Таблиця 2.3

Фрагмент переліку вакансій на посаду системного аналітика

	A	B	C	D	E	F	G	H	I	J	K	L
1	vacancy_id	title	salaryFrom	salaryTo	aryComme	date	vacancy_ur	rtDescript	fullDescription	ompany_i	city_id	badges
2	10555629	Консультант/Аналітик ERP-систем	0	0		2025-05-0	https://ral	«CRM.UA»	<head></head><p style="font-style: normal; font-weight: 400">	1	1	Офис с генератором, Офис с бесперебойны
3	10517374	System Analyst (Data Management)	0	0		2025-05-0	https://ral	Ajax Syste	<head></head><p>Ajax Systems</	2	1	Офис с генератором, Офис с бесперебойны
4	10517375	Business / System analyst Finance	0	0		2025-05-0	https://ral	Ajax Syste	<html><head></head><body><p>A	2	1	Офис с генератором, Офис с бесперебойны
5	10517302	Системний аналітик 1с	0	0		2025-05-0	https://ral	Ми шукає	<html><head></head><body><p><font style="vertical-align: inh	3	1	Бонусы / премии, Регулярный пересмотр за
6	10459193	Technical Analyst	0	0		2025-05-0	https://ral	CRH (LSE:	<html><head></head><body><p>CRH (LSE: CRH, ISE: CF	4	1	
7	10536741	Системний аналітик (для мобільного засто	0	0	Стабільно	2025-05-0	https://ral	Шукаємо	<html><head></head><body><p>Шукаємо системного ана	5	1	Регулярний перегляд зарплати, Компенсаці
8	9655669	Системний аналітик	0	0		2025-05-0	https://ral	Вітаємо!М	<html><head></head><body><p>Вітаємо!</p	6	1	Офис с бесперебойным интернетом, Беспла
9	10423505	Системний аналітик	0	0		2025-05-0	https://ral	Привіт!Ми	<html><head></head><body><p style="font-weight: 400; font-s	7	1	
10	10516657	Системний аналітик / аналітикиня	0	0		2025-05-0	https://ral	Системни	<html><head></head><body><p>Системний аналітик	8	1	Медицинская страховка, Офис с бомбоубе
11	10386684	Senior System Analyst (Mobile)	0	0		2025-05-0	https://ral	Приват Ба	<html><head></head><body><p style="font-style: normal; font-	9	1	
12	10546985	Аналітик комп'ютерних систем	0	0		2025-05-0	https://ral	Міжнарод	<html><head></head><body><p style="font-style: normal; font-	10	1	
13	10557455	Системний аналітик/Product Owner	0	0		2025-05-0	https://ral	ЮРІЙ-ФАР	<p>ЮРІЙ-ФАРМ — це українська фармацевт	11	1	
14	10557215	Аналітик 1С/BAS	0	0		2025-05-0	https://ral	Компанія	<html><head></head><body><p style="font-style: normal; font-	12	1	
15	10465434	Системний аналітик 1С	0	0		2025-05-0	https://ral	Компанія	<html><head></head><body><p>Компанія «Омега» — <strong	13	2	Компенсация обучения, Регулярный пересм
16	10524061	Системний аналітик	0	0	за результ	2025-05-0	https://ral	Привіт, на	<html><head></head><body><p style="font-weight: 400; font-s	14	1	
17	10489532	IT System Analyst	0	0		2025-05-0	https://ral	УКРЕКСІМ	<html><head></head><body><p>УКРЕКСІМБАНК запр	15	1	Бесплатное обучение, Медицинская страхо
18	9522270	Аналітик карткових систем	0	0		2025-05-0	https://ral	Ви в пошу	<html><head></head><body><p>Ви в пошуках цікавої роботи	16	3	Бонусы / премии, Медицинская страховка
19	10325893	Провідний спеціаліст з системного аналіз	0	0		2025-05-0	https://ral	ПриватБа	<head></head><p>ПриватБанк - є найбільш	9	4	
20	10540988	Senior IBM Planning Analytics & TM1 Consult	0	0		2025-05-0	https://ral	At EY, our	<head></head><p style="font-weight: 400; font-style: normal">	17	1	Медицинская страховка, Бесплатное обуче
21	10417237	Фахівець/-чиня з аналітичних систем	0	0		2025-05-0	https://ral	Danone or	<head></head><p>Danone оголошує про ві	18	1	Медицинская страховка, Офис с бомбоубе
22	10529786	Провідний аналітик програмного забезпе	0	0		2025-05-0	https://ral	KredoBank	<html><head></head><body><p style="font-weight: 400; font-s	16	3	Офис в центре, Бонусы / премии, Бесплатно
23	10480171	Аналітик комп'ютерних систем	0	0		2025-05-0	https://ral	Приєднуй	<html><head></head><body><p>Приєднуйтесь до ко	19	1	Медицинская страховка, Корпоративные м
24	10195858	Аналітик комп'ютерних систем	0	0		2025-05-0	https://ral	У ТОВ «Ог	<html><head></head><body><p>У ТОВ «Оператор га	20	1	Бонусы / премии, Бесплатное обучение
25	10107041	Системний аналітик	0	0		2025-05-0	https://ral	Vega Tele	<html><head></head><body><p><b style="font-weight: 700; fo	21	1	Бонусы / премии, Медицинская страховка,
26	10125889	Системний аналітик IT інфраструктури	0	0		2025-05-0	https://ral	ПРО НАС	<html><head></head><body><p style="font-weight: 400; font-s	22	1	Бонусы / премии, Офис с бесперебойным и
27	10551346	Системний аналітик (програміст BAS KUP)	0	0	На співбес	2025-05-0	https://ral	Ми — ком	<html><head></head><body><p style="font-style: normal; font-	23	2	Бонусы / премии, Офис с генератором, Офи

2.3 Повторюваний збір інформації для актуалізації даних

Інформація якою були заповнені таблиці з вакансіями, містами та компаніями, втрачає свою актуальність через певний час. Це спричинене тим, що кожного дня додаються нові вакансії, а також видаляються застарілі або ті, що втратили свою актуальність.

Тому для відображення, а також роботи з актуальними даними для аналізу та їх обробки, було прийнято рішення щоб не запускати код Python вручну створити файл run_information_gathering.bat. Який полегшує процедуру запуску скрипту до подвійного кліку лівою кнопкою миші на нього. Це дасть змогу оновлювати інформацію за власним бажанням в будь-який момент часу, для цього достатньо запустити файл. А також при необхідності за допомогою Task Scheduler можна задати автоматичне оновлення кожного

дня. Під оновленням мається на увазі наступний алгоритм дій програмного коду:

- парсинг вакансій
- контроль унікальності записів
- збереження інформації
- видалення зайвих елементів
- перевірка та нормалізація відповідно до принципів 3НФ
- збереження інформації у .xlsx файл

Для створення .bat-файлу необхідно відкрити блокнот, записати скрипт (Рис. 2.16) та зберегти файл з відповідним форматом документа (Рис. 2.17). Було прийнято рішення не створювати автоматичний запуск програми за розкладом, проте за потребою оновлювати інформацію примусовим запуском. Оскільки для тих завдань, що виконується під час даного дослідження не є затребуваним регулярне оновлення о певній годині кожного дня. Виконання розглянутих під час написання роботи завдань має здебільшого ситуативний характер та виконується на персональному комп'ютері, тому додаткове фонове навантаження не є бажаним. За необхідності можна створити задачу в планувальнику завдань Windows (Task Scheduler). Також можна передбачити розсилку оновленого файлу на електрону пошту користувачів які працюють з ним або викладення його до загального репозиторію який використовують в компанії.

Після запуску run_information_gathering.bat розпочнеться виконання Python коду записаного у information_gathering.py (Рис. 2.18), структуру якого було розглянуто до цього. Завершенням роботи програми буде збереження оновленого файлу system_analyst_data.xlsx (Рис. 2.19).

```
@echo off
cd /d C:\Users\Admin\Desktop\information_gathering\information_gathering
py information_gathering.py
pause
```

Рисунок 2.16 – Скрипт який виконується при запуску файл run_information_gathering.bat

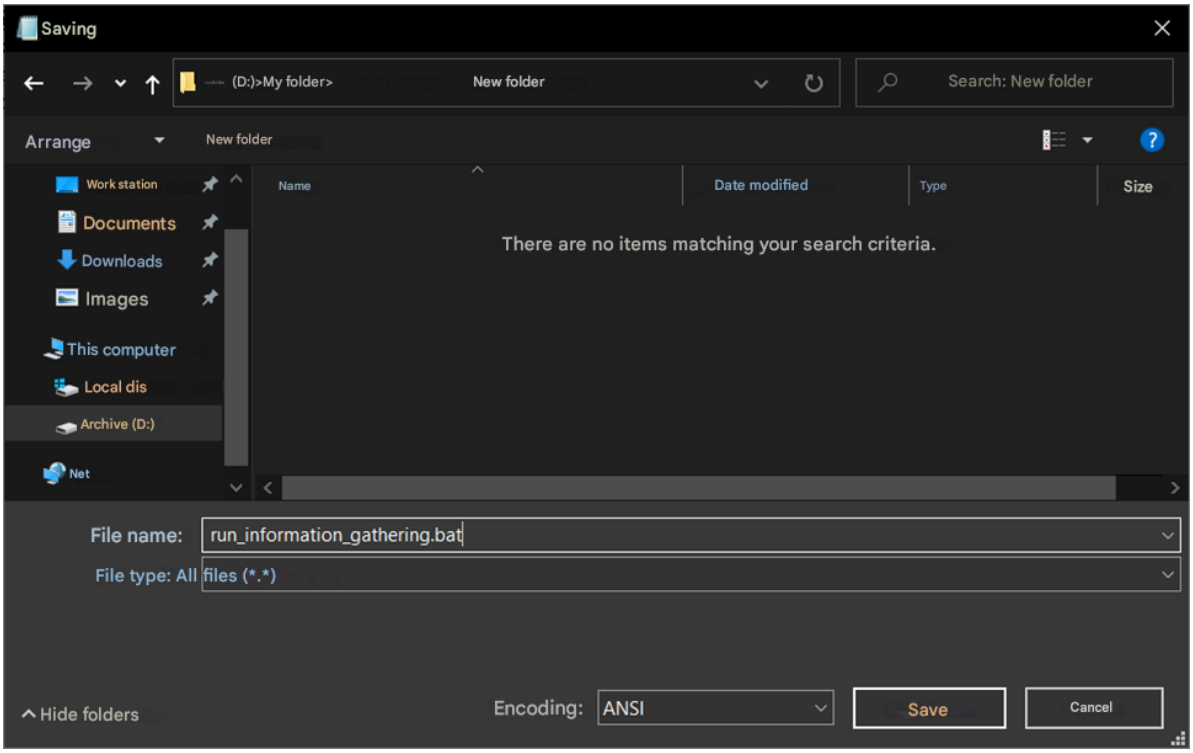


Рисунок 2.17 – Налаштування для збереження файл run_information_gathering.bat

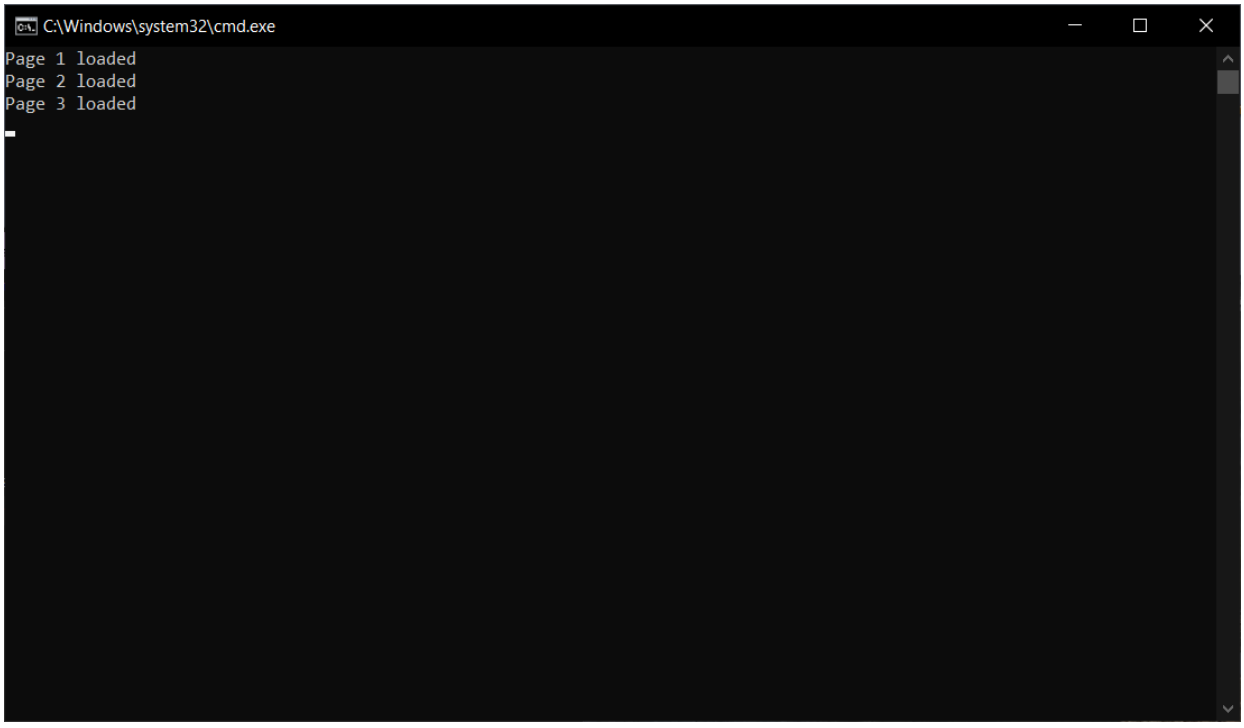


Рисунок 2.18 – Початок роботи програми information_gathering.py

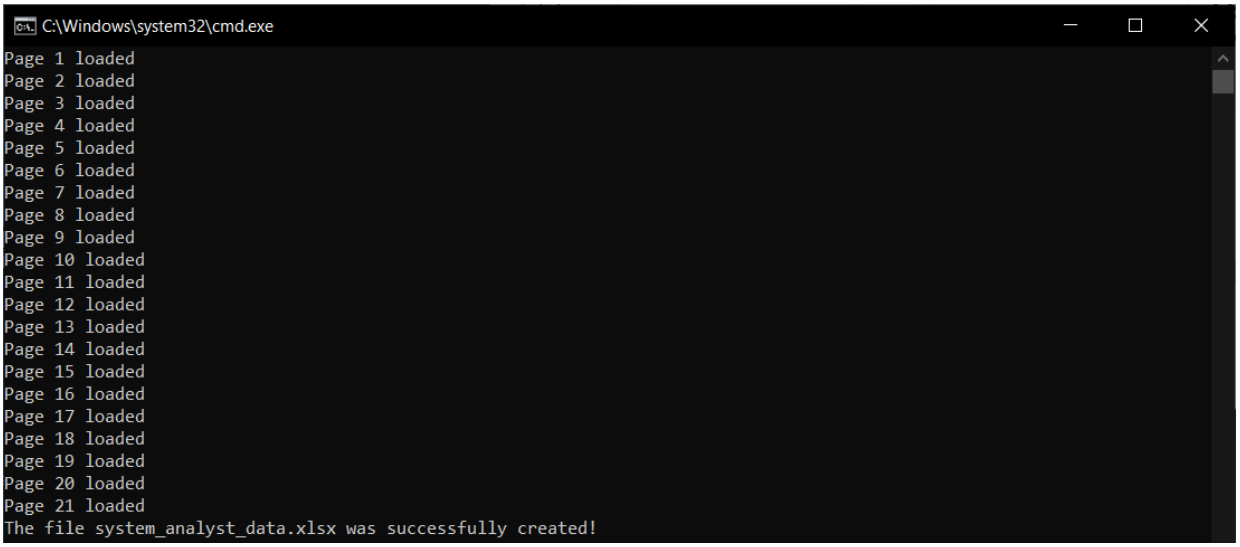


Рисунок 2.19 – Вигляд консолі після завершення роботи програми information_gathering.py

2.4. Структура даних в збереженому файлі system_analyst_data.xlsx

Для подальшого імпорту отриманих даних до бази даних та Power BI необхідно побудувати структуру згідно якої будуть будуватися моделі та зв’язки між ними. Основною моделлю є Vacancies яка містить в собі дані про vacancy_id, title, salaryFrom, salaryTo, salaryComment, date, vacancy_url, description, company_id , city_id, badges. Вона зберігає повну інформацію про кожну вакансію (Таблиця 2.4). Має первинний ключ vacancy_id та зовнішні ключі city_id та company_id.

Модель Cities (довідник міст) містить унікальні назви міст, в яких розміщені вакансії. Має стовпці city_id (первинний ключ) та city_name (Таблиця 2.5). Один запис у Cities може бути пов’язаний з багатьма вакансіями, зв’язок «один до багатьох» з Vacancies.

Модель Companies (довідник міст) містить унікальні компанії, які опублікували вакансії. Має стовпці company_id (перинний ключ) та company_name (Таблиця 2.6). Один запис у Companies може бути пов’язаний з багатьма вакансіями, зв’язок «один до багатьох» з Vacancies.

Таблиця 2.4

Структура даних в таблиці Vacancies

Назва стовпця	Опис інформації яку він зберігає
vacancy_id	Унікальний ідентифікатор вакансії на сайті Rabota.ua
title	Назва посади (вакансії)
salaryFrom	Мінімальна запропонована заробітна плата
salaryTo	Максимальна запропонована заробітна плата
salaryComment	Коментар до зарплати
date	Дата та час публікації вакансії
vacancy_url	Пряме посилання на сторінку вакансії на сайті Rabota.ua
description	Повний опис вакансії
company_id	ID компанії, який відповідає таблиці Companies
city_id	ID міста, який відповідає таблиці Cities
badges	Додаткові переваги які надає компанія

Таблиця 2.5

Структура даних в таблиці Cities

Назва стовпця	Опис інформації яку він зберігає
city_id	Унікальний ідентифікатор міста
city_name	Назва міста

Таблиця 2.6

Структура даних в таблиці Companies

Назва стовпця	Опис інформації яку він зберігає
company _id	Унікальний ідентифікатор компанії
company _name	Назва компанії, що опублікувала вакансію

Візуальна схема зв’язків між даними в таблицях створена за допомогою онлайн сервісу draw.io який створений для побудови блок-схем та інших візуальних елементів та зображено на рисунку 2.20. Доступ до нього можна отримати за посиланням: https://app.diagrams.net.

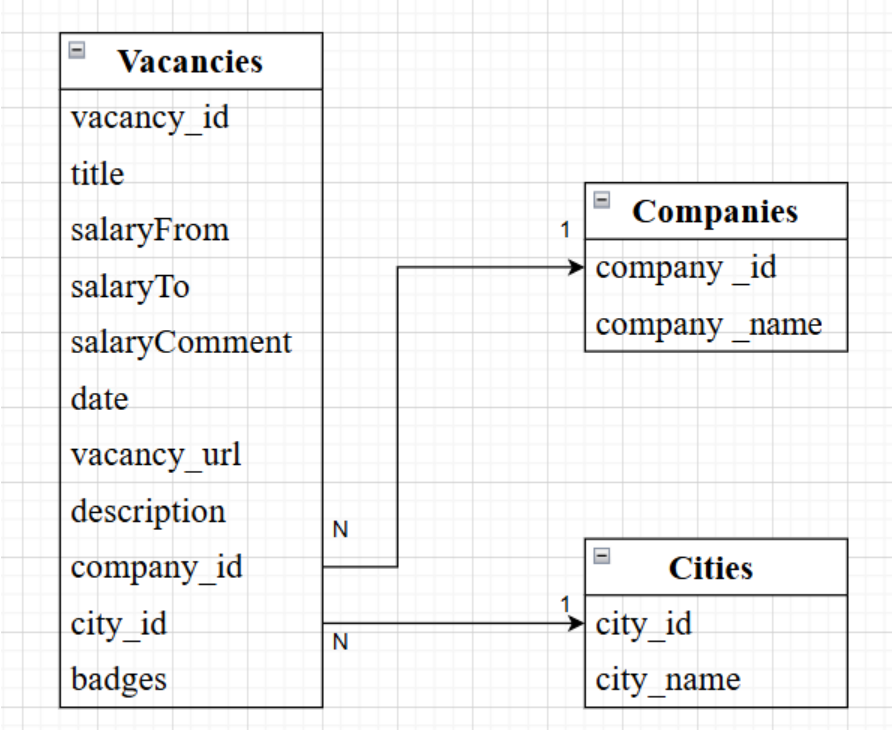


Рисунок 2.20 – Схема зв’язків між таблицями для бази даних

2.5 Структура та розділи вакансій

У результаті роботи, виконаної автоматичними скриптами у інформаційно-аналітичному розділі, були отримані певні дані з вакансіями системних аналітиків. Дані були збережені у файл формату .xlsx для зручного користування, обміну та подальшого імпортування до інших програм або баз даних.

Таблиця має стовпці: vacancy_id, title, salaryFrom, salaryTo, salaryComment, date, vacancy_url, description, company_id, city_id, badges. Кожний стовпець містить певну інформацію про вакансію компанії. Для безпосереднього аналізу даних про вимоги до кандидатів потрібні лише стовпці «description», який відповідає за опис вакансії. Також необхідне поле «id», що є ідентифікатором для кожної вакансії.

Усі вакансії мають різноманітну структура, через що їх важко стандартизувати. Зазвичай вакансія складається з таких розділів: «Про компанію» («About us»), «Обов’язки» («Responsibilities»), «Вимоги»,

(«Requirements») та «Ми пропонуємо» («We offer»). Докладніше цю структуру проілюстровано на рисунку 2.21.

Перший розділ містить загальну інформацію про компанію: її історію створення, сфера діяльності та перспективи, які вона може запропонувати кандидату. Другий розділ діяльності описує завдання та обов'язки кандидата, завдання чому уявити типовий робочий день у компанії. Найважливішим є третій розділ – «Вимоги». Саме на ньому першочергово концентрують увагу всі потенційні кандидати, оскільки він містить ключову інформацію щодо компетенцій системних аналітиків, яку необхідно детально проаналізувати. Останній розділ «Ми пропонуємо» включає перелік переваг, які компанія надає співробітнику після працевлаштування, таких як подовжена відпустка, медичне страхування, можливість працювати дистанційно тощо.

«CRM.UA» – лідер диджиталізації в Україні. Наша спеціалізація охоплює не лише розробку програмних рішень для бізнесу, але й всі види ліцензування, продажу та комплектації ліцензійних продуктів Microsoft. «CRM.UA» з 2002 року має статус Золотого партнера та є одним з найбільших партнерів Microsoft у Східній Європі. Наша команда успішно реалізувала 195 проєктів в Україні та світі.

Наш ідеальний кандидат має :

- Досвід роботи з ERP-системами. Microsoft Business Central (NAV) - **буде перевагою**
- Досвід з інтеграціями програмного забезпечення
- Розуміння архітектури IT-рішень
- Досвід у створенні складної бізнес-логіки
- Досвід проведення тренінгів/навчання для співробітників замовника - буде плюсом.

Основні обов'язки:

- Інтерв'ювання ключових осіб замовника для визначення бізнес-вимог;
- Формування та узгодження функціональних специфікацій, технічного завдання, тест-кейсів;
- Формування постановок розробникам для реалізації розширених функцій/ інтеграції;
- Тестування впроваджених рішень;

Ми пропонуємо:

- Формат роботи офіс (ремоут проговорюється окремо, в залежності від проекту)
- Можливість керувати розробкою глобальних проєктів;
- Сертифікацію Microsoft;
- Конкурентний рівень винагороди.
- Підвищення кваліфікації та професійний розвиток

Рисунок 2.21 – Приклад структури вакансії на сайті rabota.ua

2.6 Структуризація розділів вакансій

У результаті опрацювання значної кількості вакансій IT-фахівців було встановлено, що розділ із вимогами до кандидата, як правило, розміщується всередині тексту вакансії, між іншими інформаційними блоками. Для забезпечення можливості подальшого аналізу виникла потреба у попередній структуризації даних, зокрема у виокремленні розділу з вимогами до кандидатів.

Першочерговим завданням стало видалення надлишкової інформації, яка передує блоку з вимогами. З цією метою було здійснено ручний перегляд ряду вакансій, у ході якого сформовано набір ключових слів, що вказують на початку відповідного розділу. Спочатку до цього списку входили лише базові маркери: «вимоги», «requirements». Проте в подальшому аналіз охопив вакансії, які не містили зазначених слів, але використовували альтернативні терміни, такі як: «experience in», «ми шукаємо», «qualifications», «soft skills».

У результаті було сформовано розширення перелік ключових слів, серед яких: «requirements», «вимоги», «основні вимоги», «experience in», «вимоги до кандидата», «ми очікуємо», «наш ідеальний кандидат», «необхідні навички», «знання та навички», «qualifications», «нам важливо», «очікування від кандидата», «expectation», «competencies», «requirements», «required qualifications», «required experience», «required skills», «must have», «must-have skills», «skills required», «essential skills», «key skills», «we are looking for», «what we expect», «you should», «you will», «your profile», «your skills», «ideal candidate», «who you are», «responsibilities include», «expectations», «competencies», «experience in», «soft skills», «hard skills», «кваліфікаційні вимоги», «вимоги до кандидата», «що потрібно від кандидата», «кандидат має», «що ми шукаємо», «ми очікуємо», «ми шукаємо», «кандидат повинен», «очікується», «знання та навички», «професійні якості», «нам важливо», «важливо для ролі», «успішний кандидат», «наш ідеальний кандидат», «ти наш кандидат», «ви наш кандидат», «маєш досвід», «повинен мати»,

«кандидат володіє», «відповідає наступним вимогам», «наші очікування», «основні обов’язки», «ключові обов'язки», «ваші майбутні завдання», «якщо ви маєте», «професійні навички», «від тебе ми очікуємо», «які компетенції ми шукаємо», «нам потрібен саме ти, якщо маєш», «освіта / кваліфікація», «профіль кандидата», «для цього знадобиться», «для цієї ролі важливо мати», «вам до нас, якщо ви», «ключові завдання посади», «твій досвід та навички», «що потрібно», «очікування від кандидата», «ти наш кандидат», «що для нас важливо», «від вас очікуємо», «кого ми шукаємо», «для даної посади необхідні» (Рис. 2.22).

```
# Requirements start markers
requirement_markers = [
    "вимоги", "вимоги до кандидата", "основні вимоги", "необхідні навички",
    "вимоги та обов'язки", "що потрібно від кандидата", "кандидат має",
    "що ми шукаємо", "ми очікуємо", "ми шукаємо", "кандидат повинен", "очікується",
    "знання та навички", "професійні якості", "нам важливо", "важливо для ролі",
    "успішний кандидат", "наш ідеальний кандидат", "ти наш кандидат",
    "ви наш кандидат", "маєш досвід", "повинен мати", "кандидат володіє",
    "відповідає наступним вимогам", "очікування від кандидатів", "наші очікування",
    "основні обов’язки", "ключові обов'язки", "ваші майбутні завдання",
    "якщо ви маєте", "професійні навички", "від тебе ми очікуємо",
    "які компітенції ми шукаємо", "нам потрібен саме ти, якщо маєш",
    "освіта / кваліфікація", "очікуємо від кандидата", "профіль кандидата", "для цього знадобиться",
    "для цієї ролі важливо мати", "вам до нас, якщо ви", "ключові завдання посади",
    "твій досвід та навички", "що потрібно", "очікування від кандидата",
    "ти наш кандидат, якщо маєш", "про тебе", "що для нас важливо",
    "від вас очікуємо", "кого ми шукаємо", "необхідні", "для даної посади необхідні",
    "requirements", "required qualifications", "required experience",
    "required skills", "must have", "must-have skills", "qualifications",
    "skills required", "essential skills", "key skills",
    "we are looking for", "what we expect", "you should", "you will",
    "your profile", "your skills", "ideal candidate", "who you are",
    "responsibilities include", "expectations", "competencies", "experience in",
    "soft skills", "hard skills"
]
```

Рисунок 2.22 – Перелік маркерів з яких починається блок «вимог» у структурі тексту вакансії

На основі автоматичного підрахунку частоти згадування маркерів у текстах вакансій, було отримано результат. У якому близько 220 вакансій містили в собі слово «вимоги» після якого починається перелік необхідних компетенцій кандидата. Після слова «requirements» – близько 60 вакансій, «experience in» – близько 40 вакансій. Слова «ми шукаємо», «наші

очікування», «required skills», «must have», «кандидат має», «soft skills» та «qualifications» – відповідають близько 50 вакансіям.

Аналогічно до процесу видалення зайвої інформації у верхній частині над вимогами у вакансіях, було сформовано перелік ключових слів для видалення непотрібної інформації в низу під вимогами. У перелік з ключовими словами увійшли: «що пропонуємо», «що ми пропонуємо», «ми пропонуємо», «умови праці», «умови роботи», «що отримаєте», «ми гарантуємо», «що даємо», «переваги», «гарантуємо», «what we offer», «we offer», «conditions», «benefits», «основні обов'язки», «своїм співробітникам ми пропонуємо», «пропонуємо», «в обов’язки входитиме», «умови», «стратегічний вплив та кар'єрне зростання», «обов’язки», «що потрібно буде робити», «функціональні обов’язки», «чому варто долучитися», «чому», «компанія пропонує», «забезпечуємо», «місце роботи і графік», «що тебе очікує на цій позиції», «ми пропонуємо» (Рис. 2.23).

```
# End of requirement block markers
requirement_end_markers = [
    "що пропонуємо", "що ми пропонуємо", "ми пропонуємо", "умови праці",
    "умови роботи", "що отримаєте", "ми гарантуємо", "що даємо", "переваги",
    "гарантуємо", "what we offer", "we offer", "conditions", "benefits",
    "основні обов'язки", "своїм співробітникам ми пропонуємо", "пропонуємо",
    "в обов’язки входитиме", "умови", "стратегічний вплив та кар'єрне зростання",
    "обов’язки", "що потрібно буде робити", "функціональні обов ’язки",
    "чому варто долучитися", "чому", "компанія пропонує", "забезпечуємо",
    "місце роботи і графік", "що тебе очікує на цій позиції", "ми пропонуємо"
]
```

Рисунок 2.23 – Перелік маркерів які сигналізують про закінчення блоку «вимог» у структурі тексту вакансії

Також було передбачено видалення html-тегів. Оскільки інформація в описі вакансії, отримана з бази даних сайту rabota.ua має формат відображення даних HTML (Рис. 2.24).

H
description
<html><head></head><body> Шукаємо системного аналітика в нашу команду, яка розвиває мобільний застосунок. Ми пропонуємо: - всі соціальні гарантії згідно законодавству України   - стабільне офіційне працевлаштування   - гарантовану заробітну плату 2 рази/місяць   - щорічне перегляд заробітної плати   - режим роботи офіс/віддалено   - медичне страхування.   Що треба робити: 1. Проектна робота   - збір та трансформація бізнес-вимог у функціональні та нефункціональні вимоги.   - оцінка впливу змін на існуючі системи та процеси.   - участь у проєктуванні архітектури та визначенні ключових компонентів системи.   - розбиратися з документацією сторонніх систем, тестувати API-запити.   - підготовка й актуалізація проєктної документації та схем взаємодії між компонентами.   - розробка специфікацій і технічних завдань для розробки серверного ПЗ (RestAPI, мікросервіси) та мобільних додатків (нативні iOS, Android).   2. Взаємодія з командою розробки: - забезпечення розробки на основі вимог і специфікацій.   - консультування розробників щодо технічних питань, пов'язаних із функціоналом і бізнес-логікою.   - проведення рев'ю та контроль виконання вимог на етапах розробки та тестування.   3. Впровадження та експлуатація   - участь у процесі впровадження рішень, включно з підтримкою при інтеграції та запуску в промислову експлуатацію.   - розробка інструкцій і посібників для користувачів і адміністраторів.   Необхідні навички   - досвід роботи в ролі системного аналітика не менше 3 років.   - володіння основами аналізу та управління вимогами, вміння застосовувати їх на практиці   - розуміння API зв'язку з клієнтськими програмами   - розуміння механізмів та основних способів інтеграції між інформаційними системами   - досвід підготовки специфікацій на мобільні програми iOS/Android   - розуміння життєвого циклу розробки ПЗ та методологій управління проєктами (Waterfall, Agile)   Буде плюсом   - досвід роботи із СУБД   - досвід розробки прототипів інтерфейсу, моделювання бізнес-процесів (BPMN, UML)   - досвід роботи з управління проєктною документацією (JIRA, Confluence або аналогічні)   - робота в банківській сфері   Особисті якості та освіта: - вища технічна освіта (інформаційні технології, прикладна математика або аналогічні).   - критичне мислення.   - вміння працювати в команді.   - відповідальність, стресостійкість, уважність до деталей і орієнтація на результат.   Висилайте своє резюме і ми з радістю поспілкуємось з Вами особисто. </body></html>

Рисунок 2.24 – Приклад розділу з описом вакансії до видалення HTML-тегів

Перед початком обробки текстового вмісту необхідно здійснити очищення від зайвих символів HTML-кодування. Зокрема, слід видалити елементи розмітки, які не несуть змістовного навантаження: , , , </br>, , </u>, , , <ul>,
, <u>, <ol>, &amp., . Залишаємо та <li>, оскільки вони зазвичай містять структуровані пункти вимог.

Було імпортований клас BeautifulSoup з бібліотеки bs4 для ефективного парсингу HTML документу у Python, оскільки ця бібліотека дозволяє зручно витягувати дані з веб-сторінок. Перед використанням BeautifulSoup необхідно встановити її за допомогою менеджера пакетів pip (Рис. 2.25), ця команда встановлює останню версію бібліотеки BeautifulSoup4 яка є актуальною та підтримується спільнотою.

Після встановлення бібліотеки, для її використання у скрипті Python необхідно імпортувати клас BeautifulSoup з модуля bs4 (Рис. 2.26). Данна бібліотека надає широкий спектр функціональності для роботи з HTML документами.

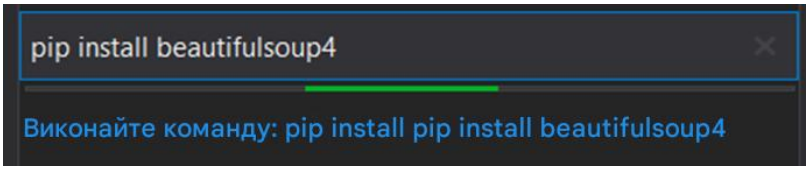


Рисунок 2.25 – Встановлення бібліотеки bs4 за допомогою менеджера пакетів pip

Бібліотека підтримує декілька парсерів для обробки HTML/XML в тому числі html.parser який є вбудованим у стандартну бібліотеку Python та не потребує додаткової установки. Для роботи алгоритму, що розглядається його функціоналу буде достатньо.

```
from bs4 import BeautifulSoup
```

Рисунок 2.26 – Імпорт клас BeautifulSoup з модуля bs4 в середовище програми

Також слід імпортувати бібліотеку re (Рис. 2.27), яка дозволяє шукати, перевіряти, обрізати або замінювати текст за заданим шаблоном. Що є необхідним для роботи алгоритму.

```
import re
```

Рисунок 2.27 – Імпорт бібліотеки re в середовище програми

Для автоматизації процесу очищення від зайвої інформації розміщеної перед та після блоку з вимогами, очищення тексту від HTML-тегів та зайвих пробілів було розроблено блок коду на мові програмування Python який представлено на рисунку 2.28.

Початковий обсяг текстової інформації у файлі з вакансіями становив 1 287 273 символів. З метою оптимізації та підготовки даних до подальшого аналізу було здійснено поетапне вилучення нерелевантної інформації. На першому етапі – після очищення верхньої частини вакансії, яка зазвичай містить загальні або рекламні блоки – обсяг стовпця «description» зменшився до 425 387 символів. Подальше «відсікання» зайвого контенту у нижній

частині вакансії, де найчастіше розміщуються дублюючі блоки та загальні пропозиції від компанії, дозволило скоротити обсяг тексту до 314 455 символі.

Таким чином, у результаті реалізованої процедури попередньої обробки загальна кількість символів зменшилась у 4,1 рази (1 287 273 / 314 455 ≈ 4,09), що свідчить про ефективність застосованих методів фільтрації.

```python
# HTML Cleanup
def full_sanitize_html(text):
    soup = BeautifulSoup(text, "html.parser")
    for tag in soup(["style", "script", "noscript", "iframe"]):
        tag.decompose()
    for tag in soup.find_all(True):
        if tag.name not in ["p", "li"]:
            tag.unwrap()
    clean = soup.get_text(separator=" ").replace("\xa0", " ").replace(" ", " ")
    return re.sub(r"\s+", " ", clean).strip()

# Search by template
def find_marker_index(text, markers):
    for marker in markers:
        pattern = re.compile(r'\b' + re.escape(marker) + r'\b', re.IGNORECASE)
        match = pattern.search(text)
        if match:
            return match.start()
    return None

# Text clipping between markers
def trim_between_requirements(text):
    clean_text = full_sanitize_html(text)
    text_lower = clean_text.lower()
    start_idx = find_marker_index(text_lower, requirement_markers)
    end_idx = find_marker_index(text_lower, requirement_end_markers)
    if start_idx is not None:
        if end_idx is not None and end_idx > start_idx:
            return clean_text[start_idx:end_idx]
        return clean_text[start_idx:]
    if len(clean_text) > 300:
        return clean_text[:1000] + "..."
    return None
```

Рисунок 2.28 – Скрипт для видалення зайвої інформації зверху та знизу блоку «вимог» у структурі тексту вакансії

Очищений та структурований масив містить лише розділ «Вимоги», який є ключовим для подальшого аналізу вакансій (Рис. 2.29). Саме в цьому розділі роботодавці, як правило, зазначають перелік необхідних компетенцій, навичок і знань, яких вони очікують від потенційного кандидата. Така

інформація є основою для подальшого проведення аналізу вимог та статистичних досліджень.

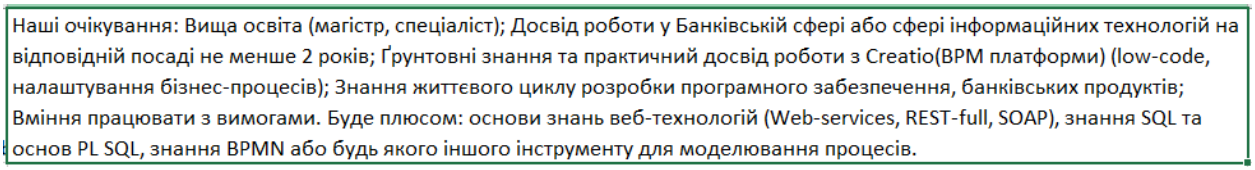


Рисунок 2.29 – Приклад блоку «вимог» після очищення від зайвої інформації

Також слід перевести на українську мову блок з вимогами у вакансіях написаними іншими мовами. Для цього скористаємося функціоналом бібліотеки googletrans, спочатку встановимо бібліотеку за допомогою менеджера пакетів pip (Рис 2.30). Також слід встановити клас detect з модуля langdetect який буде визначати якою мовою написаний текст (Рис 2.31). Це необхідно для оптимізації роботи програми та перекладати лише той текст який написаний іншими мовам. Після чого імпортуємо їх у середовище Python (Рис 2.32).

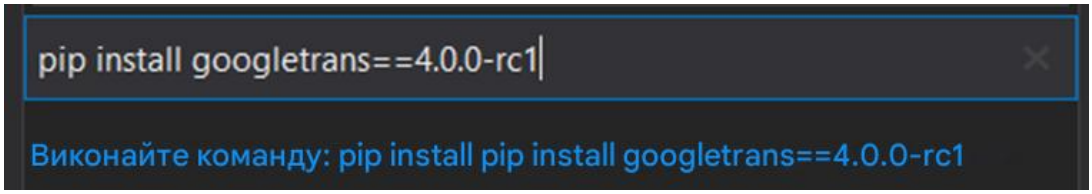


Рисунок 2.30 – Встановлення бібліотеки googletrans за допомогою менеджера пакетів pip

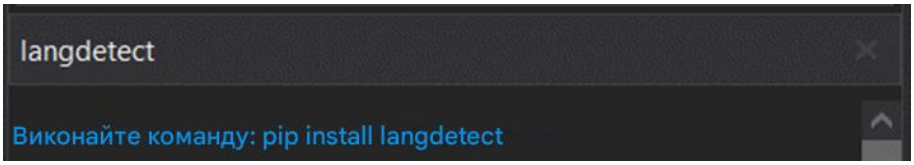


Рисунок 2.31 – Встановлення бібліотеки langdetect за допомогою менеджера пакетів pip

```
from googletrans import Translator
from langdetect import detect
```

Рисунок 2.32 – Імпорт клас Translator з модуля googletrans та клас detect з модуля langdetect в середовище програми

Створюємо та додаємо до скрипту програми функцію def translate_to_ukrainian_if_needed яка виконує переклад тексту на українську мову у випадку коли його написано іншою мовою, код якої представлено на рисунку 2.33. Також оголошуємо змінну translator яка використовує відповідний клас Translator бібліотеки googletrans.

```
# Connecting a translator
translator = Translator()

def translate_to_ukrainian_if_needed(text):
    try:
        lang = detect(text)
        if lang != 'uk':
            translated = translator.translate(text, dest='uk')
            return translated.text
        return text
    except Exception as e:
        print(f"Translation error: {e}")
        return text
```

Рисунок 2.33 – Скрипт функції def translate_to_ukrainian_if_needed

Також слід передбачити переклад назви міст на українську. Для цього створимо словник зі ста найбільших міст, згідно якому буде здійснюватися переклад (Рис. 2.34). Було вирішено робити переклад за допомогою словника тому, що виникають проблеми з правильністю написання назви міст при перекладі через перекладачі. Списку зі ста міст має вистачити для покриття всіх вакансій, оскільки посада системного аналітика затребувана лише у великих компаніях, які мають офіси або представництва лише у великих містах. При необхідності цей список легко піддається масштабуванню. Також такий підхід є найшвидшим в роботі програми серед усіх наявних варіантів.

Створимо функцію def translate_city_manual яка відповідає за переклад назви міст (Рис. 2.35), а також додаємо використання цієї функції в основному циклі програми (Рис. 2.36)

```
city_manual_map = {
    "Киев": "Київ", "Харьков": "Харків", "Одесса": "Одеса", "Днепр": "Дніпро",
    "Донецк": "Донецьк", "Запорожье": "Запоріжжя", "Львов": "Львів",
    "Кривой Рог": "Кривий Ріг", "Николаев": "Миколаїв", "Мариуполь": "Маріуполь",
    "Луганск": "Луганськ", "Винница": "Вінниця", "Севастополь": "Севастополь",
    "Симферополь": "Сімферополь", "Херсон": "Херсон", "Полтава": "Полтава",
    "Чернигов": "Чернігів", "Черкассы": "Черкаси", "Житомир": "Житомир",
    "Сумы": "Суми", "Хмельницкий": "Хмельницький", "Черновцы": "Чернівці",
    "Ровно": "Рівне", "Кропивницкий": "Кропивницький", "Ивано-Франковск": "Івано-Франківськ",
    "Тернополь": "Тернопіль", "Белая Церковь": "Біла Церква", "Кременчуг": "Кременчук",
    "Мелитополь": "Мелітополь", "Никополь": "Нікополь", "Славянск": "Слов'янськ",
    "Бердянск": "Бердянськ", "Ужгород": "Ужгород", "Павлоград": "Павлоград",
    "Бровары": "Бровари", "Ирпень": "Ірпінь", "Буча": "Буча", "Чорноморськ": "Чорноморськ",
    "Kyiv": "Київ", "Kharkiv": "Харків", "Odessa": "Одеса", "Dnipro": "Дніпро",
    "Donetsk": "Донецьк", "Zaporizhzhia": "Запоріжжя", "Lviv": "Львів",
    "Kryvyi Rih": "Кривий Ріг", "Mykolaiv": "Миколаїв", "Mariupol": "Маріуполь",
    "Luhansk": "Луганськ", "Vinnytsia": "Вінниця", "Sevastopol": "Севастополь",
    "Simferopol": "Сімферополь", "Kherson": "Херсон", "Poltava": "Полтава",
    "Chernihiv": "Чернігів", "Cherkasy": "Черкаси", "Zhytomyr": "Житомир",
    "Sumy": "Суми", "Khmelnytskyi": "Хмельницький", "Chernivtsi": "Чернівці",
    "Rivne": "Рівне", "Kropyvnytskyi": "Кропивницький", "Ivano-Frankivsk": "Івано-Франківськ",
    "Ternopil": "Тернопіль", "Bila Tserkva": "Біла Церква", "Kremenchuk": "Кременчук",
    "Melitopol": "Мелітополь", "Nikopol": "Нікополь", "Sloviansk": "Слов'янськ",
    "Berdiansk": "Бердянськ", "Uzhhorod": "Ужгород", "Pavlohrad": "Павлоград",
    "Brovary": "Бровари", "Irpin": "Ірпінь", "Bucha": "Буча", "Chornomorsk": "Чорноморськ",
    "Другие страны":"Інші країни"
}
```

Рисунок 2.34 – Словник для перекладу назви міст

```
def translate_city_manual(city):
    return city_manual_map.get(city.strip(), city.strip())
```

Рисунок 2.35 – Скрипт функції def translate_city_manual

```
original_city = vac.get("cityName", "").strip()
city = translate_city_manual(original_city)
if city not in cities:
    cities[city] = city_counter
    city_counter += 1
city_id = cities[city]
```

Рисунок 2.36 – Використання функції def translate_city_manual в основному циклі

Результатом роботи програми «information_gathering» є Excel файл з назвою «system_analyst_data.xlsx». Який зручно переглядати, передавати та зберігати. А також використовувати в подальшій аналітиці. Повний код програми представлено в Додатку Г.

Висновки до другого розділу

У цьому розділі було реалізовано процес автоматизованого збору актуальної інформації про вакансії системних аналітиків із відкритого API сайту Rabota.ua. Було досліджено можливості побудови запитів до API із використанням протоколу HTTP та інструментів Swagger. Запропоновано підхід, який дозволяє здійснювати повний парсинг усіх доступних сторінок із результатами пошук, із врахуванням динамічної зміни кількості вакансій.

Для кожної вакансії забезпечено окремий запит з метою отримання повного опису, а зібрані дані були очищені, перевірені на унікальність (за vacancy_id) та приведені до третьої нормальної форми. В результаті було сформовано три взаємопов’язані таблиці – Vacancies, Cities та Companies, що зберігаються у форматі .xlsx. Така структура є зручною для перегляду в Excel, а також забезпечує простий імпорт у бази даних або аналітичні системи типу Power BI.

Додаткового реалізовано механізм оновлення даних за допомогою запуску .bat-файлу, що дозволяє виконувати збір інформації у зручний час або за розкладом.

Також було розглянуто процес попередньої обробки, фільтрації та структуризації текстових описів вакансій системних аналітиків з метою виокремлення релевантної інформації для подальшого аналізу. Особливу увагу приділено виявленню та локалізації розділу «Вимоги», який містить найбільш цінні дані щодо необхідних компетенцій.

Для забезпечення точності структуризації було сформовано розширений перелік ключових маркерів, що сигналізує про початок і кінець блоку з вимогами. У результаті застосування алгоритму, заснованого на цих маркерах, загальний обсяг текстових даних скоротився більш ніж у чотири рази, що свідчить про ефективність застосованого підходу до фільтрації. Крім того, було реалізовано автоматичне очищення HTML-тегів та нормалізацію мови опису за допомогою бібліотек BeautifulSoup, re, googletrans та langdetect.

Окрему увагу приділено перекладу назв міст – для цього було створено словник із ста найбільш вживаних варіантів, що забезпечило точність і високу швидкодію при обробці географічних даних.

Реалізовані засоби очистки, перекладу та стандартизації тексту дозволили сформувати уніфікований, структурований масив даних, придатних для подальшого аналітичного опрацювання.

Таким чином, створено надійний інструмент для постійного моніторингу ринку праці у сфері системного аналізу, який може використовуватися у HR-аналітиці, для досліджень або бізнес-прогнозування.

РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНО-АНАЛІТИЧНИЙ

3.1 Дослідження та первинний аналіз даних щодо вимог

Було проведено роботу з дослідження блоку з вимогами до кандидатів на посаду системних аналітиків. Опрацювавши масив інформації яка складається з більш ніж трьохсот п’ятдесяти актуальних на момент дослідження вакансій з сайту robota.ua. Було виокремлено наступні найбільш згадуванні навички та вміння:

1. Hard Skills (технічні навички): SQL, UML, BPMP, REST, SOUP, JSON, XML, API, JIRA, Confluence, Postman, Swagger, MS Visio, ENL, Tableau, Power BI, Excel, Python, Business Intelligence, Use case.
2. Soft Skills (особисті якості): аналітичне мислення, увага до деталей, комунікабельність, відповідальність, самостійність, ініціативність, критичне мислення, тайм-менеджмент, робота в команді, клієнтоорієнтованість.
3. Методології та процеси: Agile, Scrum, Kanban, SDLC, Waterfall, User stories, Acceptance criteria.
4. Додатково: англійська, українська, документація, комунікація, презентація, технічне завдання, взаємодія з командою, тестування, інтеграція, автоматизація, ERP, CRM, fintech.

Після чого було розроблено програму мовою Python. Яка проводить підрахунок кількості вакансій в яких згадується певна компетентність. А також сортування їх від найбільш розповсюджених до тих, що зустрічаються рідше. Дана програма буде використовувати файл під назвою «system_analyst_data.xlsx», який містить інформацію отримана в наслідок роботи програмного коду розглянутого у попередньому розділі даної кваліфікаційної роботи.

Також слід передбачити можливість збереження нового Excel файлу формату .xlsx, який буде містити назву компетентності, кількість її згадувань у вибірці з вакансіями, а також показник розповсюдженості у відсотках. Це необхідно для проведення подальшої аналітики в програмі Power BI. А також для можливості відслідковування та проведення порівняльного аналізу змін вимог до кандидатів на ринку праці з плином часу.

Для цього створимо список з ключовими словами для їх пошуку та підрахунку (Рис. 3.1). Повний код програми під назвою «requirements analytics» представлений в додатку Д.

```
# Keywords
keywords = [
    # Hard Skills
    "sql", "uml", "bpmn", "rest", "soap", "json", "xml", "api", "jira",
    "confluence", "postman", "swagger", "ms visio", "etl", "tableau",
    "power bi", "excel", "python", "business intelligence", "use case",

    # Soft Skills
    "аналітичне мислення", "увага до деталей", "комунікабельність", "відповідальність",
    "самостійність", "ініціативність", "критичне мислення", "тайм-менеджмент",
    "робота в команді", "клієнтоорієнтованість",

    # Methodologies
    "agile", "scrum", "kanban", "sdlc", "waterfall", "user stories", "acceptance criteria",

   # Additionally
    "англійська", "українська", "документація", "комунікація", "презентація",
    "технічне завдання", "взаємодія з командою", "проектна документація",
    "тестування", "інтерв'ювання", "інтеграція", "автоматизація", "цифрова трансформація",
    "erp", "crm", "fintech"
]
```

Рисунок 3.1 – Список з ключовими компетенціями

Результат роботи програми представлено на рисунку 3.2. Фрагмент даних, що зберігаються в файлі «keyword_frequency_results.xlsx» представлено в таблиці 3.1.

```
Top 10 most sought-after competencies and skills in job vacancies:
sql                              – 97 вакансій (26.58%)
excel                            – 91 вакансій (24.93%)
bpmn                             – 84 вакансій (23.01%)
тестування                       – 75 вакансій (20.55%)
power bi                         – 61 вакансій (16.71%)
jira                             – 60 вакансій (16.44%)
uml                              – 54 вакансій (14.79%)
аналітичне мислення              – 50 вакансій (13.7%)
api                              – 47 вакансій (12.88%)
erp                              – 43 вакансій (11.78%)

The result is saved to a file 'keyword_frequency_results.xlsx'
Press any key to continue . . .
```

Рисунок 3.2 – Результат роботи програми «requirements analytics»

Таблиця 3.1

Фрагмент даних, що зберігається в файлі «keyword_frequency_results»

№	Keyword	Vacancy_Count	Presence (%)
1	sql	100	26,81
2	excel	94	25,2
3	bpmn	86	23,06
4	тестування	75	20,11
5	power bi	61	16,35
6	jira	60	16,09
7	uml	53	14,21
8	api	51	13,67
9	аналітичне мислення	48	12,87
10	erp	45	12,06
11	відповідальність	42	11,26
12	confluence	41	10,99
13	agile	39	10,46
14	sdlc	31	8,31
15	англійська	28	7,51

3.2 Візуалізація та подальша аналітика даних

Для проведення подальшого аналізу та покращення сприйняття інформації за допомогою візуального її подання скористуємося функціоналом програми Power BI.

Для цього нам слід імпортувати до звіту файли «system_analyst_data.xlsx» та «keyword_frequency_results.xlsx». Імпортуємо всі наявні сторінки в цих звітах (Рис 3.3 та 3.4).

Після чого будуємо схему зв'язку між таблицями, яка представлена на рисунку 3.5. Видаляємо зайву інформацію, яка не потрібна для даного типу аналітики. А також за допомогою функціоналу редактора Power Query змінюємо формат подання дати в стовпці «date» на відображення лише дати, без вказання точного часу публікації вакансії (Рис. 3.6).

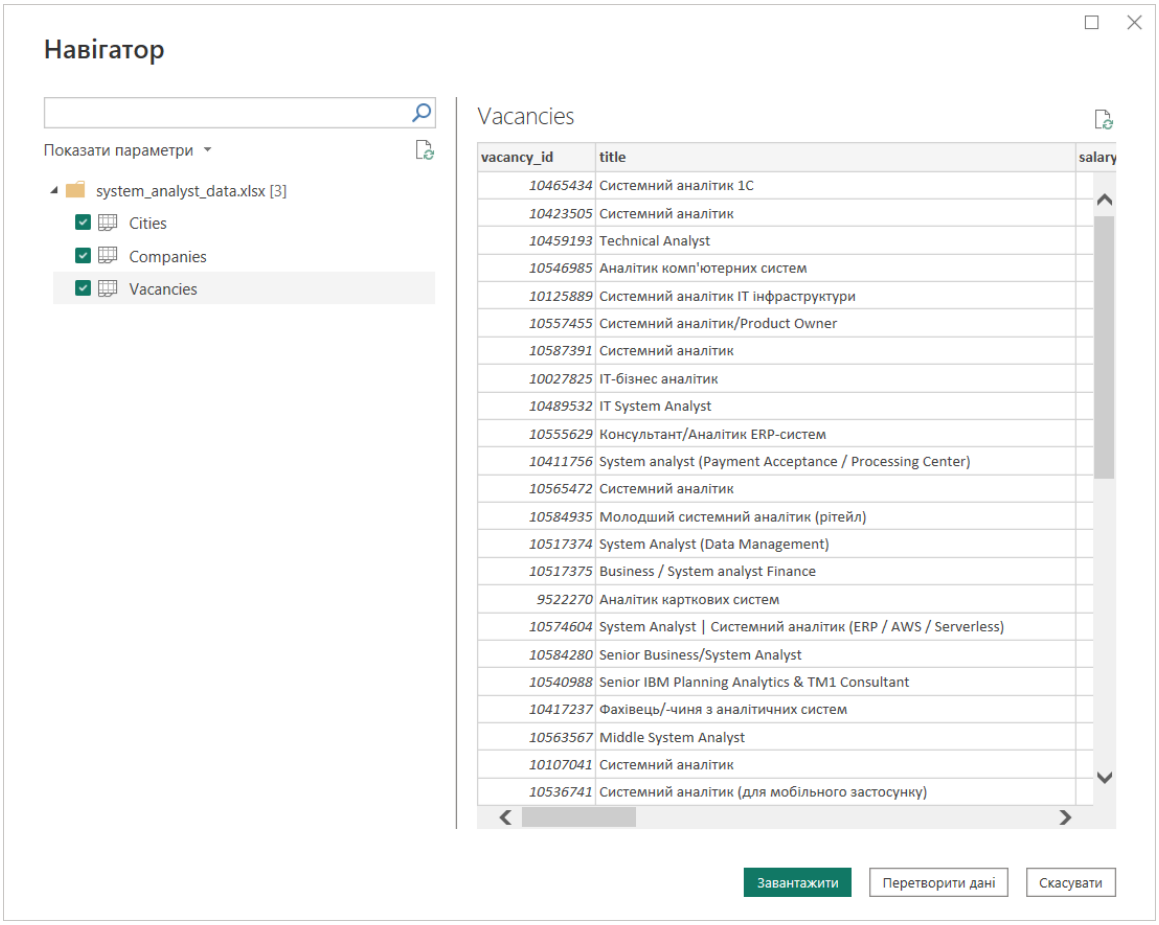
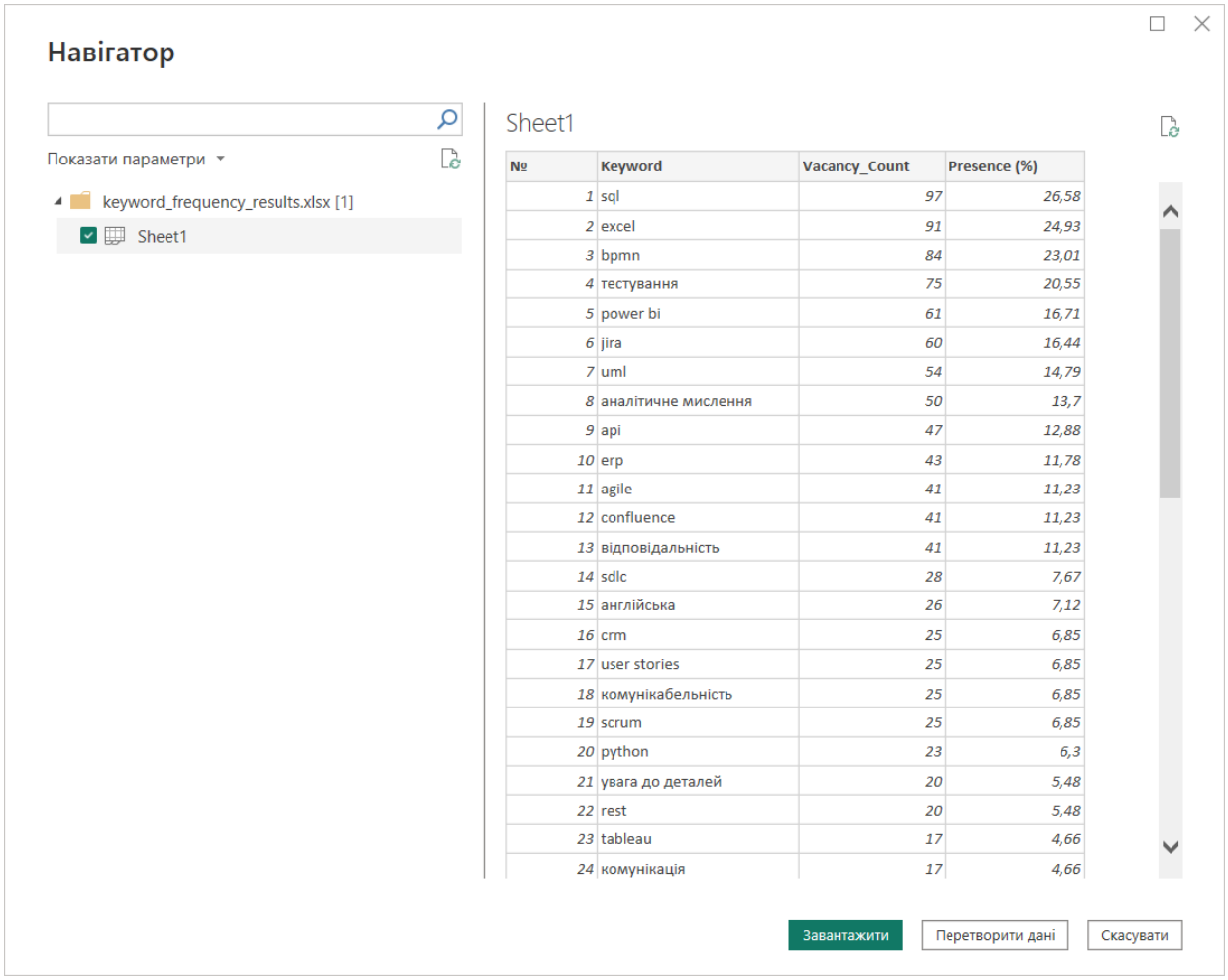


Рисунок 3.3 – Завантажуємо до середовища Power BI дані з файлу «system_analyst_data.xlsx»



№	Keyword	Vacancy_Count	Presence (%)
1	sql	97	26,58
2	excel	91	24,93
3	bpmn	84	23,01
4	тестування	75	20,55
5	power bi	61	16,71
6	jira	60	16,44
7	uml	54	14,79
8	аналітичне мислення	50	13,7
9	api	47	12,88
10	erp	43	11,78
11	agile	41	11,23
12	confluence	41	11,23
13	відповідальність	41	11,23
14	sdlc	28	7,67
15	англійська	26	7,12
16	crm	25	6,85
17	user stories	25	6,85
18	комунікабельність	25	6,85
19	scrum	25	6,85
20	python	23	6,3
21	увага до деталей	20	5,48
22	rest	20	5,48
23	tableau	17	4,66
24	комунікація	17	4,66

Рисунок 3.4 – Завантажуємо до середовища Power BI дані з файлу «keyword_frequency_results.xlsx»

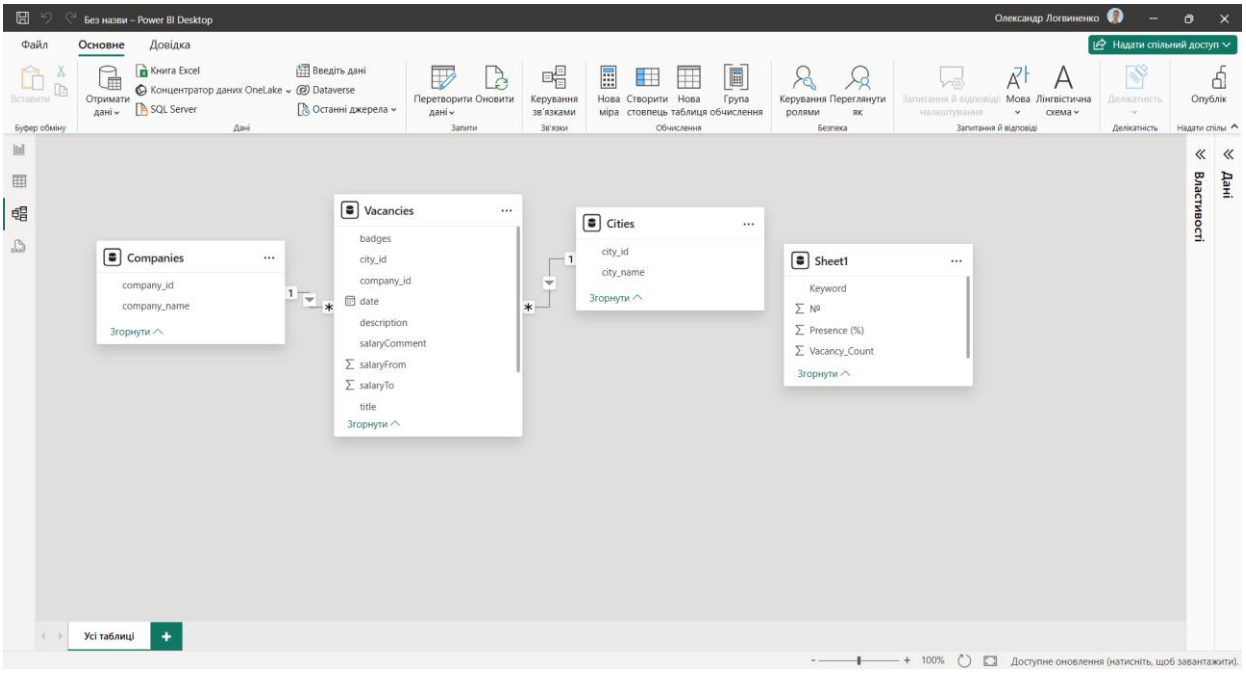


Рисунок 3.5 – Схема зв’язків між таблицями

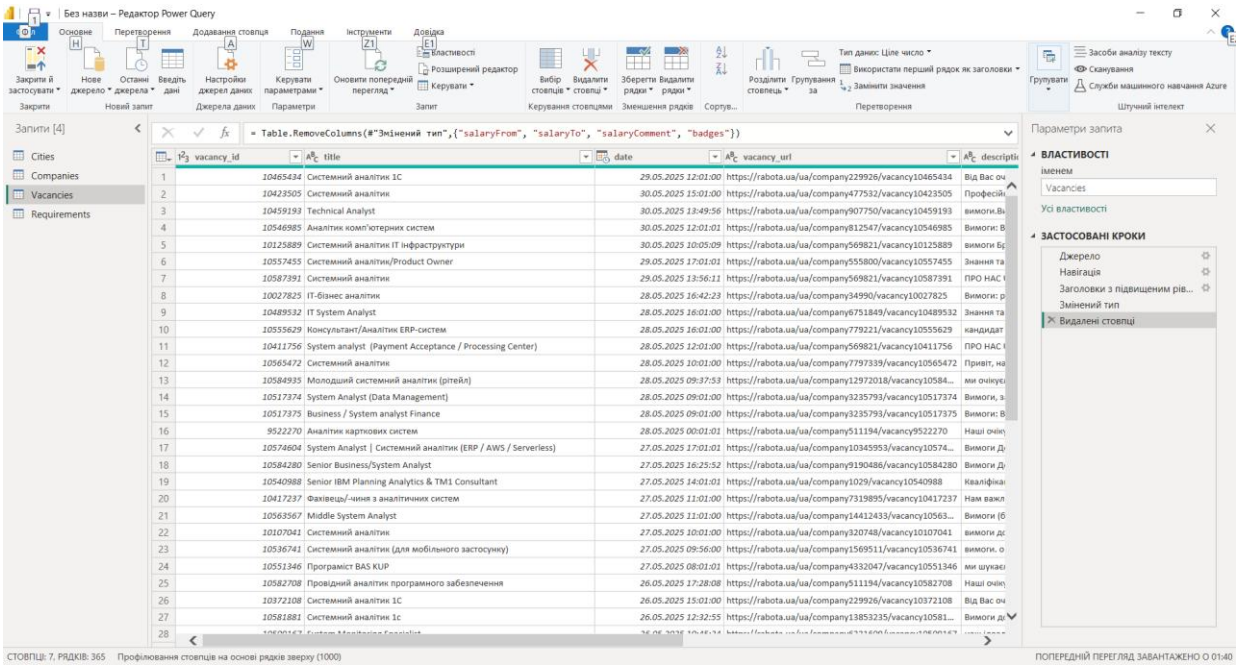
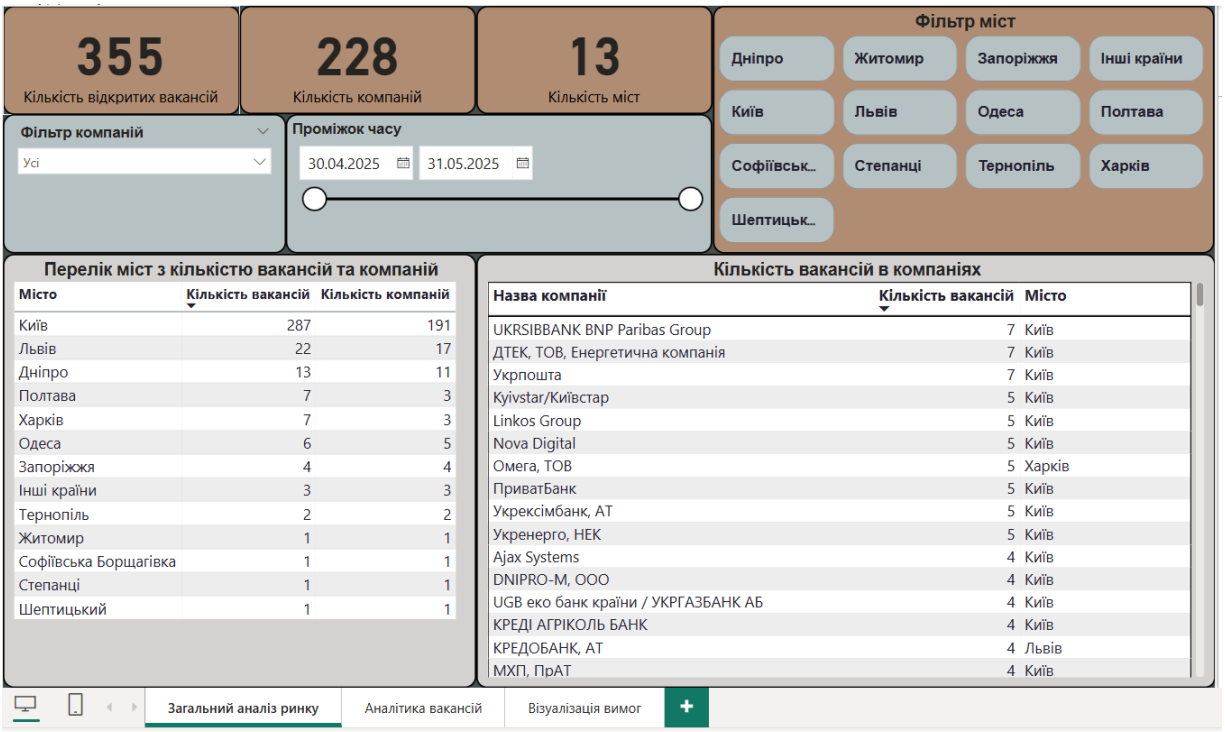


Рисунок 3.6 – За допомогою функціоналу редактора Power Quarry готуємо дані до аналізу

Створюємо три сторінки з динамічними дашбордами. Перша сторінка «Загальний аналіз ринку» передбачена для аналізу компаній та міст за кількість відкритих вакансій на посаду системного аналітика (system analyst). Зовнішній вигляд цієї сторінки представлено на рисунку 3.7.

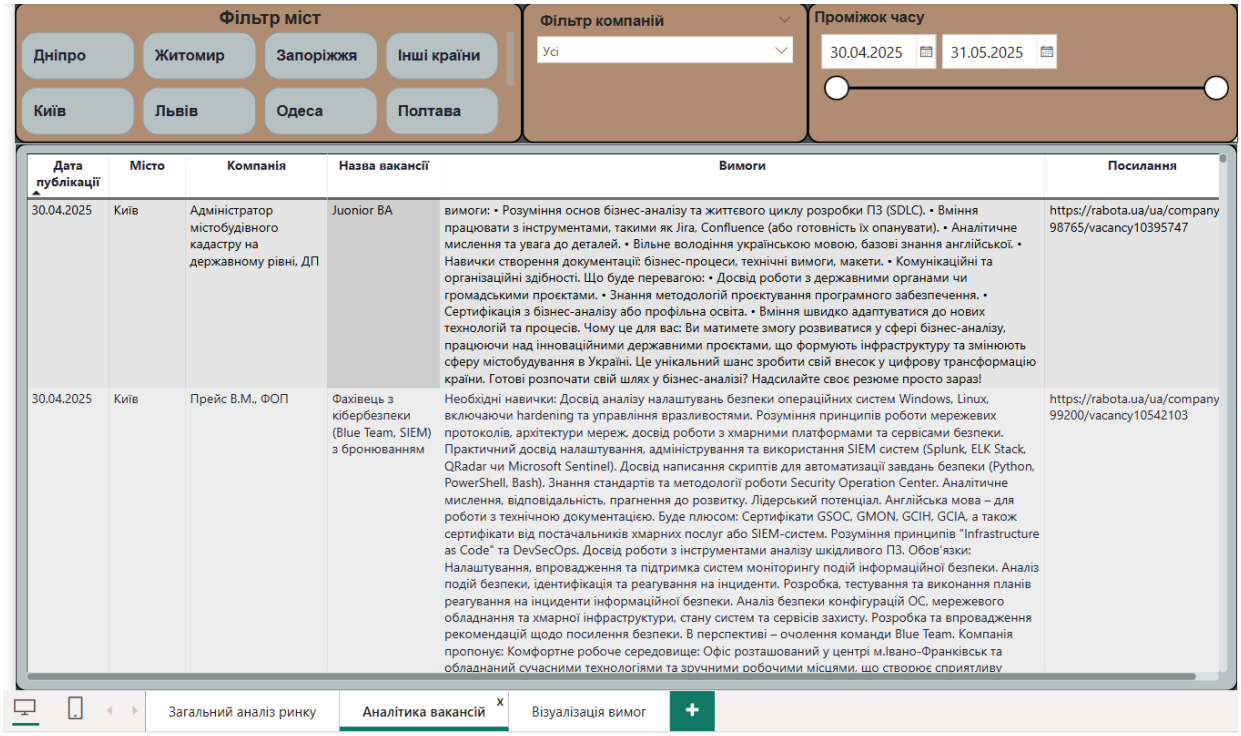


Місто	Кількість вакансій	Кількість компаній
Київ	287	191
Львів	22	17
Дніпро	13	11
Полтава	7	3
Харків	7	3
Одеса	6	5
Запоріжжя	4	4
Інші країни	3	3
Тернопіль	2	2
Житомир	1	1
Софіївська Борщагівка	1	1
Степанці	1	1
Шептицький	1	1

Назва компанії	Кількість вакансій	Місто
UKRSIBBANK BNP Paribas Group	7	Київ
ДТЕК, ТОВ, Енергетична компанія	7	Київ
Укрпошта	7	Київ
Kyivstar/Київстар	5	Київ
Linkos Group	5	Київ
Nova Digital	5	Київ
Омега, ТОВ	5	Харків
ПриватБанк	5	Київ
Укрексімбанк, АТ	5	Київ
Укренерго, НЕК	5	Київ
Ajax Systems	4	Київ
DNIPRO-M, ООО	4	Київ
UGB еко банк країни / УКРГАЗБАНК АБ	4	Київ
КРЕДІ АГРІКОЛЬ БАНК	4	Київ
КРЕДОБАНК, АТ	4	Львів
МХП, ПрАТ	4	Київ

Рисунок 3.7 – Зовнішній вигляд сторінки «Загальний аналіз ринку»

Основним завдання другої сторінки «Аналітика вакансій» є надання можливості аналізу та виокремлення особливостей вимог до кандидатів у певних містах або компаніях. Також передбачено можливість вибору часового відрізку в якому були опубліковані вакансії. Ознайомитись з зовнішнім виглядом та вмістом цієї сторінки можна на рисунку 3.8.



Дата публікації	Місто	Компанія	Назва вакансії	Вимоги	Посилання
30.04.2025	Київ	Адміністратор містобудівного кадастру на державному рівні, ДП	Juonior BA	вимоги: • Розуміння основ бізнес-аналізу та життєвого циклу розробки ПЗ (SDLC). • Вміння працювати з інструментами, такими як Jira, Confluence (або готовність їх опанувати). • Аналітичне мислення та увага до деталей. • Вільне володіння українською мовою, базові знання англійської. • Навички створення документації: бізнес-процеси, технічні вимоги, макети. • Комунікаційні та організаційні здібності. Що буде перевагою: • Досвід роботи з державними органами чи громадськими проєктами. • Знання методологій проєктування програмного забезпечення. • Сертифікація з бізнес-аналізу або профільна освіта. • Вміння швидко адаптуватися до нових технологій та процесів. Чому це для вас: Ви матимете змогу розвиватися у сфері бізнес-аналізу, працюючи над інноваційними державними проєктами, що формують інфраструктуру та змінюють сферу містобудування в Україні. Це унікальний шанс зробити свій внесок у цифрову трансформацію країни. Готові розпочати свій шлях у бізнес-аналізі? Надсилайте своє резюме просто зараз!	https://rabota.ua/ua/company98765/vacancy10395747
30.04.2025	Київ	Прейс В.М., ФОП	Фахівець з кібербезпеки (Blue Team, SIEM) з бронюванням	Необхідні навички: Досвід аналізу налаштувань безпеки операційних систем Windows, Linux, включаючи hardening та управління вразливостями. Розуміння принципів роботи мережевих протоколів, архітектури мереж, досвід роботи з хмарними платформами та сервісами безпеки. Практичний досвід налаштування, адміністрування та використання SIEM систем (Splunk, ELK Stack, QRadar чи Microsoft Sentinel). Досвід написання скриптів для автоматизації завдань безпеки (Python, PowerShell, Bash). Знання стандартів та методології роботи Security Operation Center. Аналітичне мислення, відповідальність, прагнення до розвитку. Лідерський потенціал. Англійська мова – для роботи з технічною документацією. Буде плюсом: Сертифікати GSOC, GMON, GCIH, GCIA, а також сертифікати від постачальників хмарних послуг або SIEM-систем. Розуміння принципів "Infrastructure as Code" та DevSecOps. Досвід роботи з інструментами аналізу шкідливого ПЗ. Обов'язки: Налаштування, впровадження та підтримка систем моніторингу подій інформаційної безпеки. Аналіз подій безпеки, ідентифікація та реагування на інциденти. Розробка, тестування та виконання планів реагування на інциденти інформаційної безпеки. Аналіз безпеки конфігурацій ОС, мережевого обладнання та хмарної інфраструктури, стану систем та сервісів захисту. Розробка та впровадження рекомендацій щодо посилення безпеки. В перспективі – очолення команди Blue Team. Компанія пропонує: Комфортне робоче середовище: Офіс розташований у центрі м.Івано-Франківськ та обладнаний сучасними технологіями та зручними робочими місцями, що створює сприятливу	https://rabota.ua/ua/company99200/vacancy10542103

Рисунок 3.8 – Зовнішній вигляд сторінки «Аналітика вакансій»

На третій сторінці «Візуалізація вимог» за допомогою гістограми продемонстровано кількість згадувань певних вимог. А також представлено таблицю на якій продемонстровано кількість згадувань та відсоток від загальної кількості вакансій в яких згадується компетентність (Рис. 3.9).

Також робимо адаптацію сторінок для перегляду на мобільних пристроях, зовнішній вигляд яких представлено на рисунках 3.10, 3.11 та 3.12.

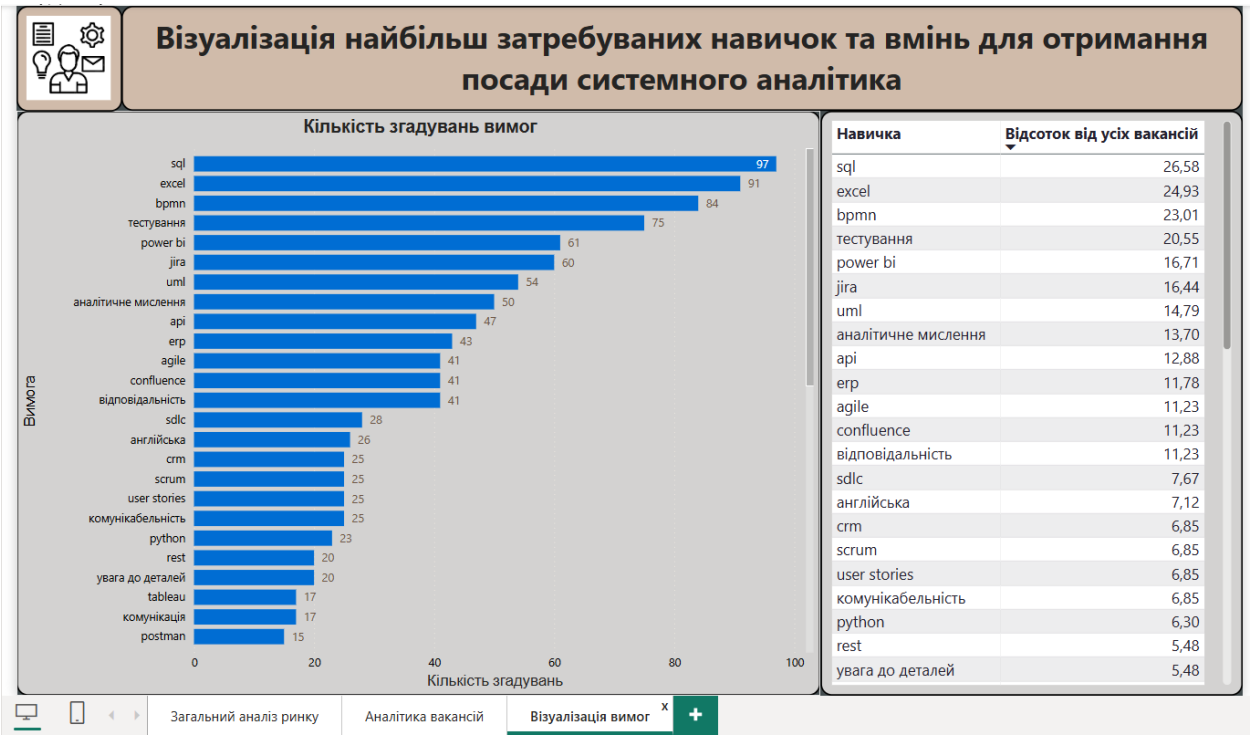


Рисунок 3.9 – Зовнішній вигляд сторінки «Візуалізація вимог»

Рисунок 3.10 – Адаптація сторінки «Загальний аналіз ринку» до перегляду на мобільних пристроях

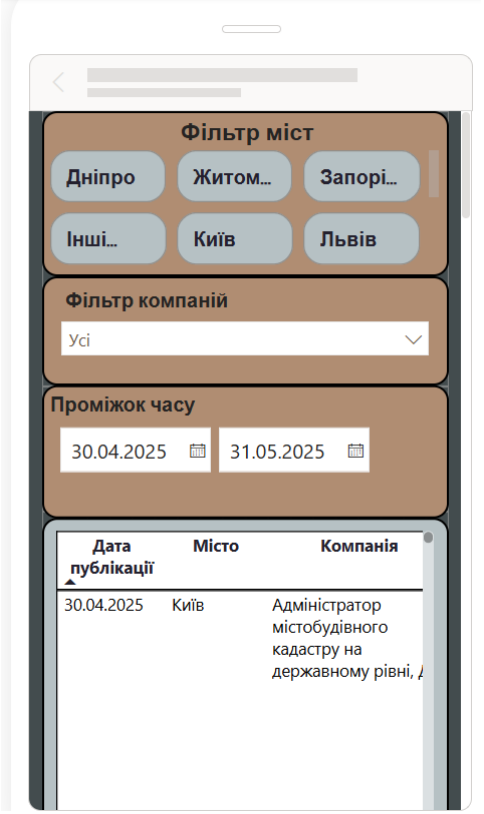


Рисунок 3.11 – Адаптація сторінки «Аналітика вакансій» до перегляду на мобільних пристроях

Рисунок 3.12 – Адаптація сторінки «Візуалізація вимог» до перегляду на мобільних пристроях

Публікуємо звіт «Аналіз вимог» до особистої робочої області задля отримання можливості легко переглядати на будь яких пристроях з доступом до інтернету. А також ділитися ним, налаштовувати рівні доступу та оновлювати данні одночасно для всіх користувачів (Рис. 3.13).

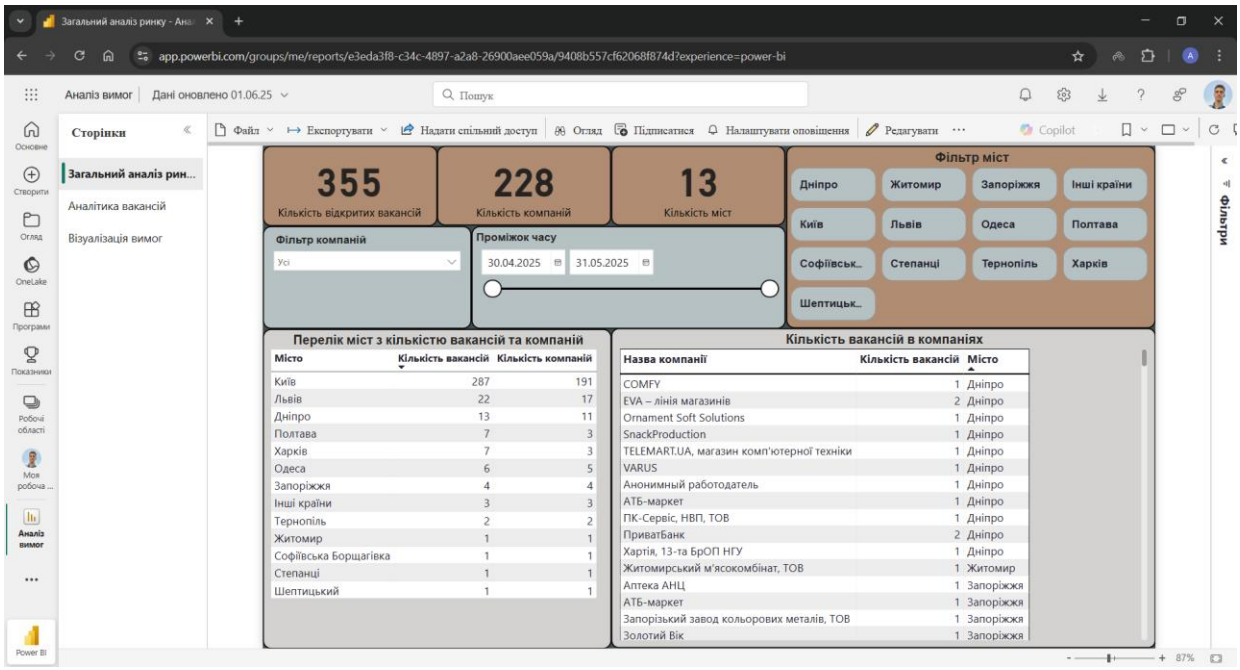


Рисунок 3.13 – Вигляд звіту на онлайн платформі Power BI

3.3 Кінцевий аналіз отриманих даних

На основі візуалізації результатів контент-аналізу текстових описів вакансій системного аналітика на сайтах працевлаштування було сформовано список найбільш часто згадуваних вимог. Візуальна гістограма та відповідна таблична форма відображають як абсолютну кількість згадувань, так і відносну частку (у відсотках) від загальної кількості вакансій.

Було виокремлено найчастіше згадувані технічні навички та методології:

- SQL – 97 згадувань (26,58 %);
- Excel – 91 згадування (24,93%);
- BPMN – 84 згадування (23,01%);
- Тестування – 75 згадувань (20,55%);

- Power BI – 61 згадування (16,71%);
- Jira – 60 згадувань (16,44%);
- UML – 54 згадування (14,79%);
- API – 47 згадувань (12,88%);
- ERP – 43 згадування (11,78%);
- Agile – 41 згадування (11,23%);
- Confluence – 41 згадування (11,23%);
- SDLC – 28 згадувань (7,67 %);
- CRM – 25 згадувань (6,85%);
- User Stories – 25 згадувань (6,85%);
- Scrum – 25 згадувань (6,85%);
- Python – 23 згадування (6,30%);
- Rest – 20 згадувань (5,48%).

Окрім технічних компетенцій, у переліку вимог також простежується акцент на особистісних якостях кандидатів. Найчастіше роботодавці згадують такі характеристики:

- аналітичне мислення (13,70%);
- відповідальність (11,23%);
- комунікабельність та уважність до деталей (6,85%).

Це свідчить про те, що ефективне виконання функцій системного аналітика потребує не лише володіння професійними інструментами, але й розвитку м’яких навичок, що забезпечує якісну міжособистісну взаємодію, уважність до специфіки завдань і здатність до критичного осмислення інформації.

Крім того, висока частота згадувань гнучких методологій розробки програмного забезпечення таких як Agile, Scrum, SDLC, User Stories – підтверджує їх широке використання у практиці організацій. Це вказує на важливість для кандидатів мати розуміння принципів і практик роботи в умовах гнучких середовищ розробки.

Таким чином, отримані результати дозволяють окреслити портрет ідеального кандидат, який поєднує в собі знання мов запитів, уміння працювати з BI-інструментами, володіти стандартами бізнес-аналітики та дуже важливими особистісними якостями. Отримані дані можуть бути використані для побудови навчальних програм, а також самостійного розвитку кандидатів.

3.4 Портрет ідеального кандидата на посаду системного аналітика

На основі отриманих даних в ході контентного аналізу вакансій зібраних із використанням API платформи rabota.ua. А також відокремивши вимоги, які найчастіше висувалися до претендентів на момент проведення дослідження. Було отримано можливість сформувати портер ідеального кандидата на посаду системного аналітика (system analyst).

Аналіз вакансій показав, що найчастіше роботодавці очікують від кандидата наступні професійні навички (Hard Skills):

- володіння мовами моделювання бізнес-процесів (BPMN, UML);
- знання SQL, розуміння структури баз даних та принципів побудови запитів;
- досвід роботи з інструментами управління вимогами, такими як Jira, Confluence, а також з інструментами тестування API (Postman, Swagger);
- вміння працювати з форматами обміну даними: JSON, XML;
- написання та ведення документації відповідно до стандартів: SRS, BRD, Use case, ERD;
- користування візуальними та аналітичними інструментами: MS Visio, Enterprise Architect, Tableau, Power BI, Excel;
- розуміння принципів інтеграції та побудови системного дизайну.

У більшості вакансій згадуються вимоги до особистісних характеристик (Soft Skills), зокрема:

- аналітичне мислення та здатність до системного підходу в розв’язані завдань;
- уважність до деталей, відповідальність, самоорганізація;
- комунікабельність та здатність до ефективної командної взаємодії;
- клієнтоорієнтованість, ініціативність, критичне мислення, тайм-менеджмент.

Більшість роботодавців мають наступні очікування в питанні освіти та досвіду:

- наявність вищої технічної або економічної освіти (галузі: інформаційні технології, економіка, прикладна математика);
- досвід роботи на аналогічній посаді від 1-2 років;
- досвід участі у повному циклі проєктної діяльності за методологіями Agile/Scrum/SDLC.

Також дуже розповсюдженими є вимоги до:

- впевненого володіння українською та англійською мовами;
- навички ведення технічної, аналітичної та проєктної документації;
- розуміння ключових понять: інтеграція, тестування, автоматизація, цифрова трансформація, ERP, CRM, FinTech.

Таким чином, отриманий профіль кандидата відображає реальні очікування роботодавців і може слугувати основою для розробки навчальних програм, профорієнтаційних моделей та рекомендацій з розвитку професійних компетенцій у сфері системного аналізу.

Висновки до третього розділу

У цьому розділі було здійснено повноцінний контент-аналіз вакансій системного аналітика, розміщених на сайті працевлаштування robota.ua.

Зібрані текстові описи вимог до кандидатів були оброблені за допомогою спеціально створеного інструменту аналізу на мові Python, що дозволило здійснити кількісну оцінку частоти згадування конкретних компетенцій. Отримані результати подано у вигляді таблиць та гістограми, які візуалізовано в середовищі Power BI.

Встановлено, що найбільш поширеними вимогами до кандидатів є володіння мовою запитів SQL, офісними та аналітичними інструментами (Excel, Power BI), знання моделей бізнес-процесів (BPMN, UML) та методологій розробки ПЗ (Agile, Scrum). Значну увагу роботодавці також приділяють soft skills, зокрема аналітичному мисленню, відповідальності та вмінню працювати в команді.

Зібрана інформація дозволила не лише виявити тренди на ринку праці, а й сформувати узагальнений портрет ідеального кандидата. Результати мають практичну цінність для претендентів на посаду системного аналітика, закладів освіти, а також HR-фахівців, які займаються підбором персоналу в IT-сфері.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було всебічно досліджено сучасні вимоги до кандидатів на посаду системного аналітика, зокрема шляхом аналізу вакансій, розміщених на популярній онлайн-платформі з працевлаштування robota.ua. У межах роботи було реалізовано повноцінний цикл збору, обробки, аналізу й візуалізації текстових даних за допомогою інструментів Python та Power BI, що дозволило автоматизувати процес виявлення релевантних вимог до кандидатів.

З технічного боку було здійснено:

- Підключення до API сайту robot.ua з метою динамічного збору актуальних вакансій за ключовим запитом «system analyst».
- Обробка текстових полів опису вакансій: очищення HTML-розмітки, нормалізація тексту, вилучення тільки тих фрагментів, які стосуються вимог до кандидатів. Було враховано велику варіативність структури вакансій, що вимагало побудови списку маркерів початку та завершення блоку вимог.
- Автоматичне виявлення мови тексту та переклад на українську в разі такої необхідності. За допомогою використання бібліотеки langdetect та googletrans. Це забезпечило уніфікованість аналізу незалежно від мови оригіналу.
- Стандартизація назв міст та компаній, присвоєння унікальних ідентифікаторів для нормалізації даних відповідно до принципів побудови реляційних баз даних.
- Створення файлу «system_analyst_data.xlsx», який містить окремі листи з вакансіями, назвами компаній. Назвами міст та тими вакансіями, які були виключено через неможливість коректного вилучення вимог.

Для вивчення вмісту вакансій було побудовано словник ключових слів, що охоплює чотири основні категорії вимог:

1. Hard Skills – технічні навички (SQL, BPMN, UML, REST, API, Power BI, Excel, Python, Json, Swagger тощо);
2. Soft Skills – особистісні якості (аналітичне мислення, уважність до деталей, відповідальність, комунікабельність, ініціативність тощо);
3. Методології – підходи до організації процесу (Agile, Scrum, SDLC, Kandan, User Stories, Acceptance Criteria);
4. Інше – мови, сфери застосування, цифрові процеси, ведення документації, взаємодія з командою, автоматизація тощо.

У результаті було проаналізовано понад 360 унікальних вакансій, виділено ключові вимоги та розраховано частоту їх згадувань. Візуалізація даних у середовищі Power BI дозволила створити інтерактивні таблиці та гістограму, які відображають частотність згадувань кожного з критеріїв. Зокрема, було визначено:

- Найпопулярнішими технічними навичками є знання SQL (26,58%), Excel (24,93%), BPMN (23,01%), Power BI (16,71%), UML (14,79%);
- Серед soft skills найчастіше зустрічається аналітичне мислення (13,70%), відповідальність (11,23%), увага до деталей (5,48%).
- Методології Agile, Scrum, SDLC згадуються у понад 10% вакансій;
- Значна частина вакансій містить вимоги щодо англійської мови, здатності до комунікації та роботи в команді.

Також було визначено, що вимога досвіду від 1 до 3 років роботи на подібній посаді є найбільш поширеною серед відкритих вакансій на момент проведення досліджень.

Отримані результати мають практичне значення для:

- студентів, які прагнуть адаптувати власну траєкторію навчання під актуальні вимоги роботодавців;

- викладачів та укладачів навчальних програм, які можуть на основі висновків отриманих в результаті аналізу оновити зміст дисциплін для спеціальності 124 «Системний аналіз»;
- компаній, які шукають спеціалістів – для узагальнення та стандартизації вимог до кандидатів.

У результаті проведення дослідження було доведено доцільність застосування інструментів автоматизації аналізу текстових вакансій і візуалізації даних для вирішення практичних завдань у сфері кадрової політики. Запропонований підхід може бути розширений на інші професії та джерела, а також доповнений модулями машинного навчання для оцінки релевантності профілю кандидатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кваліфікаційна робота бакалавра [Електронний ресурс] : методичні рекомендації для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / уклад.: Т. А. Желдак, Т. В. Хом'як, А. В. Малієнко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 32 с. url: https://ir.nmu.org.ua/handle/123456789/170863

2. HTTP documentation [Електронний ресурс] // Mozilla Developer Network. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/HTTP

3. Передатестаційна практика [Електронний ресурс] : методичні рекомендації для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» спеціальності 124 Системний аналіз / уклад.: Т.А. Желдак, А.В. Малієнко, О.Д. Станіна ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 24 с. https://ir.nmu.org.ua/handle/123456789/173256

4. Gil T. What Do HTTP and HTTPS Stand For? [Електронний ресурс] // Lifewire.

5. Навчальна практика з обчислень [Електронний ресурс] : методичні рекомендації для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» спеціальності 124 Системний аналіз / уклад.: Л.С. Коряшкіна, О.М. Алєксєєв, Д.М. Гаранжа, Ю.О. Шевченко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 57 с. https://ir.nmu.org.ua/handle/123456789/173197

6. Програмування та алгоритмічні мови [Електронний ресурс] : методичні рекомендації до виконання лабораторних робіт для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз. У 2 ч. Ч 2 / уклад.: Т.В. Хом'як, Ю.О. Шевченко, Д.М. Гаранжа ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2026. – 48 с. Режим доступу: https://ir.nmu.org.ua/handle/123456789/173149

7. Python Introduction [Електронний ресурс] // W3Schools. – Режим доступу: https://www.w3schools.com/python/python_intro.asp

8. Introduction to Python [Електронний ресурс] // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/introduction-to-python/

9. Python Programming Language Tutorial [Електронний ресурс] // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/python-programming-language-tutorial/

10. Introduction to Python [Електронний ресурс] // Google Developers. – Режим доступу: https://developers.google.com/edu/python/introduction

11. Python Blurb [Електронний ресурс] // Python.org. – Режим доступу: https://www.python.org/doc/essays/blurb/

12. Python Applications [Електронний ресурс] // Python.org. – Режим доступу: https://www.python.org/about/apps/

13. Python (programming language) [Електронний ресурс] // Wikipedia. – Режим доступу: https://en.wikipedia.org/wiki/Python_%28programming_language%29

14. What is Python? [Електронний ресурс] // Amazon Web Services. – Режим доступу: https://aws.amazon.com/what-is/python/

15. Sharma A. Getting Started with the Requests Module in Python [Електронний ресурс] // Medium. – Режим доступу: https://theanujsharma.medium.com/getting-started-with-the-requests-module-in-python-323e9003ff1e

16. Quickstart [Електронний ресурс] // Requests Documentation. – Режим доступу: https://requests.readthedocs.io/en/latest/user/quickstart/

17. Python Requests Tutorial [Електронний ресурс] // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/python-requests-tutorial/

18. Python Requests Module [Електронний ресурс] // W3Schools. – Режим доступу: https://www.w3schools.com/python/module_requests.asp

19. Requests [Електронний ресурс] // PyPI. – Режим доступу: https://pypi.org/project/requests/

20. Requests (software) [Електронний ресурс] // Wikipedia. – Режим доступу: https://en.wikipedia.org/wiki/Requests_(software)

21. Beautiful Soup: Build a Web Scraper with Python [Електронний ресурс] // Real Python. – Режим доступу: https://realpython.com/beautiful-soup-web-scraper-python/

22. BeautifulSoup Tutorial [Електронний ресурс] // ScrapingDog. – Режим доступу: https://www.scrapingdog.com/blog/beautifulsoup-tutorial-web-scraping-with-python/

23. Build a Web Scraper with Beautiful Soup [Електронний ресурс] // SerpAPI. – Режим доступу: https://serpapi.com/blog/beautiful-soup-build-a-web-scraper-with-python/

24. Python Time Module [Електронний ресурс] // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/python-time-module/

25. Python Datetime Module [Електронний ресурс] // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/python-datetime-module

26. Introduction to Pandas in Python [Електронний ресурс] // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/introduction-to-pandas-in-python

27. Pandas Getting Started [Електронний ресурс] // W3Schools. – Режим доступу: https://www.w3schools.com/python/pandas/pandas_getting_started.asp

28. Getting Started [Електронний ресурс] // Pandas Documentation. – Режим доступу: https://pandas.pydata.org/docs/getting_started/overview.html

29. pandas.DataFrame.describe [Електронний ресурс] // Pandas Documentation. – Режим доступу: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html

30. Python Software Foundation. Regular expression operations [Електронний ресурс] // Python Documentation. – Режим доступу: https://docs.python.org/3/library/re.html

31. GeeksforGeeks. Regular Expressions in Python – Examples [Електронний ресурс]. – Режим доступу: https://www.geeksforgeeks.org/regular-expression-python-examples/

32. Google Developers. Python Class – Regular Expressions [Електронний ресурс]. – Режим доступу: https://developers.google.com/edu/python/regular-expressions

33. How to Translate List and Pandas Data Frame using Googletrans Library in Python [Електронний ресурс] // Medium. Analytics Vidhya. – 2021. – Режим доступу: https://medium.com/analytics-vidhya/translate-list-and-pandas-data-frame-using-googletrans-library-in-python-f28b8cb84f21.

34. Language Translator using Google API in Python [Електронний ресурс] // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/language-translator-using-google-api-in-python.

35. Detect an Unknown Language using Python [Електронний ресурс] // GeeksforGeeks. – Режим доступу: https://www.geeksforgeeks.org/detect-an-unknown-language-using-python.

36. Power BI overview [Електронний ресурс] // Microsoft Learn. – Режим доступу: https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview

37. Power BI product page [Електронний ресурс] // Microsoft. – Режим доступу: https://www.microsoft.com/en-us/power-platform/products/power-bi

38. Аналіз програмного забезпечення [Електронний ресурс] : методичні рекомендації до виконання практичних робіт для здобувачів ступеня бакалавра галузі знань 12 (F) Інформаційні технології / уклад.: О.С. Мінєєв, Ю.О. Шевченко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2026. – 28 с. https://ir.nmu.org.ua/handle/123456789/173326

39. Виробнича практика [Електронний ресурс] : методичні рекомендації для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» спеціальності 124 Системний аналіз / уклад.: Т.А. Желдак, Л.С. Коряшкіна, С.А. Ус ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 25 с. https://ir.nmu.org.ua/handle/123456789/173193

ДОДАТКИ

Додаток А

Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення	Найменування	Кількість аркушів	Примітки
1				
2		Документація		
3				
4	САУ.КР.25.35.ПЗ	Пояснювальна записка	N1	Формат А4
5				
6		Демонстраційний матеріал	N2	Презентація на CD-R
7				
8		Копія роботи	1	Диск CD-R
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

					САУ.КР.YY.ZZ.ДА.ПЗ.			
Змін.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		ПІБ			**Матеріали кваліфікаційної роботи**	Літ.	Аркуш	Аркушів
К. розд.		ПІБ						
Керівн.		ПІБ				НТУ «ДП», 12; 124-21-2		
Н.контр.		ПІБ						
Зав. каф.		ПІБ						