

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

"На правах рукопису"
УДК 004.65.2:658.5

«До захисту допущено»

Завідувач кафедри

Наталія АУШЕВА

“ ” _____ 2025р.

Магістерська дисертація

на здобуття ступеня другого (магістерського) рівня вищої освіти
за освітньою програмою “Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему Методи повнотекстового пошуку інформації в системі ODOO

Виконав(ла): студентка 2 курсу, групи ТР-41мп

Лазоренко Анна Русланівна

(прізвище, ім’я, по батькові)

(підпис)

Науковий керівник д.т.н., професор Олексій ШУШУРА

(науковий ступінь, вчене звання, ім’я ПРІЗВИЩЕ)

(підпис)

Рецензент д.т.н., професор Андрій ПЕТРОВИЧ

(посада, науковий ступінь, вчене звання, ім’я ПРІЗВИЩЕ)

(підпис)

Н.контроль ст. викладач Ольга БЕСПАЛА

(посада, ім’я ПРІЗВИЩЕ)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка _____

(підпис)

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – другий (магістерський)

спеціальність 122 “Комп’ютерні науки”

Освітньо-професійна програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ

Завідувач кафедри ЦТЕ

Наталія АУШЕВА

«__» _____ 2025 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Лазоренко Анні Русланівні

(прізвище, ім’я, по батькові)

1. Тема дисертації Методи повнотекстового пошуку інформації в системі ODOO

керівник роботи Шушура Олексі Миколайович, д.т.н., професор

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «3» листопада 2025 р. № 4779-с

2. Термін подання роботи студентом 09.12.2025 року

3. Об’єкт дослідження: Процеси повнотекстового пошуку та обробки даних в ERP-системах, зокрема системи ODOO.

4. Вихідні дані: мова програмування Python, мова програмування JavaScript, СКБД PostgreSQL, середовище розробки PyCharm Community Edition

5. Перелік питань, які потрібно розробити _____

1) провести аналіз існуючих методів імплементації повнотекстового пошуку в ODOO

2) розробити архітектуру універсального модуля повнотекстового пошуку.

3) спроектувати та реалізувати алгоритм первинної підготовки та оновлення даних

для індексу;

4) обрати і обґрунтувати засоби розробки;

5) розробити програмне забезпечення та провести його тестування

6. Орієнтований перелік ілюстративного матеріалу **діаграми, що демонструють роботу алгоритмів веб-інтерфейсу, архітектуру системи, бази даних, взаємодію користувачів із системою, класи програмного забезпечення; приклади роботи з програмним продуктом.**

7. Орієнтований перелік публікацій: **тези в збірнику конференцій**

8. Дата видачі завдання **16 вересня 2024 р.**

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів підготовки магістерської дисертації	Термін виконання	Примітка
1	Аналіз методів дослідження за літературними джерелами	01.09.2025-18.09.2025	Виконано
2	Вибір та налаштування засобів реалізації, методів повнотекстового пошуку	19.09.2025-22.09.2025	Виконано
3	Програмна реалізація	23.09.2025-15.10.2025	Виконано
4	Проведення обчислювальних експериментів	16.10.2025-9.10.2025	Виконано
5	Аналіз результатів обчислювальних експериментів	10.10.2025-22.10.2025	Виконано
6	Захист програмного продукту	23.10.2025	Виконано
7	Оформлення магістерської дисертації	24.10.2025-29.11.2025	Виконано

Студент

(підпис)

Анна ЛАЗОРЕНКО

(ім'я, ПРІЗВИЩЕ)

Керівник роботи

(підпис)

Олексій ШУШУРА

(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Актуальність теми. Зі зростанням обсягів даних, що накопичуються в корпоративних інформаційних системах, особливо в ERP-платформах, проблема ефективного текстового пошуку стає критично важливою. У системі ODOO, що містить сотні моделей та тисячі взаємопов'язаних записів, стандартні інструменти пошуку обмежені межами окремих модулів і не забезпечують глобального доступу до інформації. Це ускладнює роботу користувачів, знижує продуктивність бізнес-процесів і створює потребу у створенні універсального повнотекстового пошуку, який працюватиме в межах самої системи без зовнішніх сервісів. Саме тому розробка програмного забезпечення, яка поєднує можливості PostgreSQL та архітектуру ODOO і забезпечує швидкий пошук у великих інформаційних базах, є актуальним і практично значущим завданням.

Метою роботи є розробка ефективної системи повнотекстового пошуку для ODOO, здатної оптимізувати процеси пошуку та забезпечити швидкий доступ до великих обсягів інформації з високою точністю результатів, використовуючи наявні в системі інструменти та з мінімальними налаштуванням.

Завдання дослідження:

- провести аналіз існуючих методів імплементації повнотекстового пошуку в ODOO;
- розробити архітектуру універсального модуля повнотекстового пошуку;
- спроектувати та реалізувати алгоритм первинної підготовки та оновлення даних для індексу;
- обрати і обґрунтувати засоби розробки;
- розробити програмне забезпечення та провести його тестування.

Об'єкт дослідження – процеси повнотекстового пошуку та обробки даних в ERP-системах, зокрема системи ODOO.

Предмет дослідження – методи та засоби реалізації універсального повнотекстового пошуку в ODOO на основі PostgreSQL, включно з архітектурою індексації, чергою обробки та алгоритмами виконання запитів.

Практична цінність полягає в забезпеченні повнотекстовий пошук по всій базі ODOO, підтримці дотримання прав доступу, мультикомпанійності, індексування через PostgreSQL без потреби зовнішніх сервісів. Це дозволяє оптимізувати роботу користувачів, значно скоротити час пошуку інформації, покращити навігацію великими даними та підвищити ефективність бізнес-процесів.

Апробація результатів дисертації. Основні положення даної роботи доповідались та обговорювались на XXII Міжнародній науково-практичній конференції молодих вчених і студентів, 22-25 квітня 2025 року м. Київ, підготовлена одна стаття у науковому виданні, включеному до Переліку фахових видань України категорії Б.

Дисертація складається з вступу, п'яти розділів та висновків. Повний обсяг дисертації складає 97 сторінок, 5 таблиць, 26 рисунків, 3 сторінки списку використаних джерел у кількості 32 найменування.

Ключові слова: *повнотекстовий пошук, ODOO, PostgreSQL, JSON, індексування, ERP-система, пошуковий запит, фільтрація, глобальний пошук.*

ABSTRACT

Relevance of the topic. With the growth of data volumes accumulated in corporate information systems, especially in ERP platforms, the problem of effective text search becomes critically important. In the ODOO system, which contains hundreds of models and thousands of interrelated records, standard search tools are limited to individual modules and do not provide global access to information. This complicates the work of users, reduces the productivity of business processes, and creates a need for a universal full-text search that will work within the system itself without external services. That is why the development of software that combines the capabilities of PostgreSQL and the ODOO architecture and provides fast search in large databases is a relevant and practically significant task.

The aim of this research is to develop an effective full-text search system for ODOO that can optimize search processes and provide fast access to large amounts of information with high accuracy of results, using the tools available in the system and with minimal configuration.

Research objectives:

- to analyze existing methods of implementing full-text search in ODOO;
- to develop the architecture of a universal full-text search module;
- to design and implement an algorithm for initial data preparation and updating for the index;
- to select and justify development tools;
- to develop software and test it.

The object of research is the processes of full-text search and data processing in ERP systems, in particular the ODOO system.

The subject of research is the methods and means of implementing universal full-text search in ODOO based on PostgreSQL, including indexing architecture, processing queue, and query execution algorithms.

The practical value lies in ensuring full-text search across the entire ODOO database, supporting access rights compliance, multi-company functionality, and

indexing via PostgreSQL without the need for external services. This allows you to optimize user experience, significantly reduce information search time, improve navigation of large data sets, and increase the efficiency of business processes.

Approbation of the dissertation results. The main provisions of this work were reported and discussed at the XXII International Scientific and Practical Conference of Young Scientists and Students, April 22-25, 2025, Kyiv, and one article was prepared in a scientific publication included in The list of professional publications of Ukraine, category B.

The dissertation consists of an introduction, five chapters, and conclusions. The full volume of the dissertation is 97 pages, 5 tables, 26 figures, and 3 pages of references, totaling 32 titles.

Keywords: full-text search, ODOO, PostgreSQL, JSON, indexing, ERP system, search query, filtering, global search

ЗМІСТ

ВСТУП.....	11
1 ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ЕФЕКТИВНОГО МОДУЛЯ ПОВНОТЕКСТОВОГО ПОШУКУ ДЛЯ ODOO	13
1.1 Мета і завдання роботи.....	13
1.2 Особливості системи ODOO	14
1.3 Визначення функціональних вимог до модуля пошуку.....	16
1.4 Технологічні Вимоги	17
Висновки до розділу 1	17
2 АНАЛІЗ СУЧАСНИХ РІШЕНЬ ВПРОВАДЖЕННЯ ПОВНОТЕКСТОВОГО ПОШУКУ	19
2.1 Спеціалізовані інструменти	19
2.1.1 Платформа Elasticsearch	19
2.1.2 Платформа Apache Solr.....	22
2.1.3 Порівняння платформ Elasticsearch та Apache Solr	24
2.2 Повнотекстовий пошук у сучасних ERP-системах.....	26
2.2.1 Система Microsoft Dynamics 365	27
2.2.2 Система Oracle ERP Cloud	29
2.3 Модулі повнотекстового пошуку в системі ODOO.....	30
2.3.1 Підходи до реалізації	31
2.3.2 Модуль Bloomup Fulltext Search	32
2.3.3 Модуль SH Global Search	33
Висновки до розділу 2	34
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	36
3.1 Вибір засобів розробки	36
3.1.1 Мова програмування Python	37
3.1.2 Середовище розробки PyCharm.....	38
3.1.3 Система управління базами даних PostgreSQL	39
3.1.4 Використання API ODOO	41

3.2 Основи повнотекстового пошуку в PostgreSQL	43
3.2.1 Векторна модель представлення тексту в PostgreSQL.....	43
3.2.2 Морфологічна нормалізація та полімовний пошук.....	44
3.2.3 Розширена логіка запитів tsquery і позиційні оператори.....	45
3.2.4 Алгоритми ранжування релевантності: ts_rank і ts_rank_cd	46
3.2.5 Контекстне підсвічування	47
3.3 Архітектура та реалізація модуля повнотекстового пошуку.....	48
3.3.1 Загальна архітектура.....	48
3.3.2 Підготовка даних та первинне індексування	49
3.3.3 Алгоритм підтримки актуальності	51
3.3.4 Обробка запиту і здійснення пошуку.....	52
3.3.5 Структура бази даних	55
3.3.6 Функціональні можливості	57
Висновки до розділу 3	58
4 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ.....	59
4.1 Встановлення модуля.....	59
4.2 Здійснення пошуку.....	61
4.3 Фільтрація результатів.....	65
Висновки до розділу 4	66
5 РОЗРОБКА СТАРТАП ПРОЕКТУ	68
5.1 Ідея стартап проекту	68
5.2 Технологічний аудит стартап-проекту.....	72
5.3 Аналіз ринкових можливостей стартап-проекту	74
5.3.1 Загальна характеристика ринку корпоративного пошуку в системах ODOO	74
5.3.2 Сегментація цільової аудиторії.....	75
5.3.3 Ринкова потреба у модулі Full-text search	77
5.3.4 Аналіз конкурентного середовища	79
5.3.5 Аналіз тенденцій ринку та перспектив розвитку.....	81
5.4 Розроблення ринкової стратегії стартап-проекту	82

5.5 Розроблення маркетингової програми стартап-проєкту	83
Висновки до розділу 5	85
ВИСНОВКИ.....	87
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	89
ДОДАТОК А.....	92
ДОДАТОК Б	926

ВСТУП

Стрімке зростання обсягів цифрових даних, що супроводжує розвиток сучасних інформаційних систем, формує нові вимоги до засобів їх опрацювання та пошуку. У корпоративних платформах, зокрема ERP-системах, де зберігаються великі масиви взаємопов'язаних текстових даних, питання швидкого доступу до інформації стає критично важливим для підтримання ефективності бізнес-процесів. Стандартні засоби пошуку в таких системах часто не забезпечують достатньої продуктивності, точності та гнучкості, що знижує швидкість роботи користувачів та ускладнює навігацію даними [1].

Система ODOO, яка широко застосовується підприємствами різних масштабів, містить значну кількість бізнес-моделей та об'єктів, серед яких часто необхідно здійснювати глобальний пошук [2–4]. Проте штатні засоби пошуку обмежені межами окремих моделей та не охоплюють усієї бази даних, що створює труднощі у роботі з великими інформаційними масивами. Існуючі підходи, що базуються на зовнішніх пошукових сервісах, потребують окремої інфраструктури та складного адміністрування, що не завжди є прийнятним для компаній малого та середнього бізнесу. Це підтверджує актуальність створення універсального засобу повнотекстового пошуку, який працює в межах самої системи ODOO, забезпечуючи високу швидкість обробки запитів і зручність використання [5].

Актуальність обраної теми полягає в необхідності розроблення ефективного інструменту глобального пошуку, який би поєднував можливості сучасних підходів до обробки текстової інформації та специфіку роботи ERP-системи ODOO. Критичний аналіз існуючих рішень засвідчив, що вони або обмежуються окремими модулями, або вимагають додаткових сервісів, або покладають складні етапи налаштування на користувача. Розв'язання цієї проблеми є важливим і для вдосконалення платформи ODOO, і для розширення інструментарію роботи з великими базами даних.

Метою магістерської роботи є розробка ефективної системи повнотекстового пошуку для ODOO, здатної оптимізувати процеси пошуку та забезпечити швидкий

доступ до великих обсягів інформації з високою точністю результатів, використовуючи наявні в системі інструменти та з мінімальними налаштуванням.

Для досягнення поставленої мети були визначені такі завдання:

- провести аналіз існуючих методів імплементації повнотекстового пошуку в ODOO;
- розробити архітектуру універсального модуля повнотекстового пошуку;
- спроектувати та реалізувати алгоритм первинної підготовки та оновлення даних для індексу;
- обрати і обґрунтувати засоби розробки;
- розробити програмне забезпечення та провести його тестування.

Об'єктом дослідження є процеси повнотекстового пошуку та обробки даних в ERP-системах, зокрема системи ODOO.

Предметом дослідження є методи та засоби реалізації універсального повнотекстового пошуку в ODOO на основі PostgreSQL, включно з архітектурою індексації, чергою обробки та алгоритмами виконання запитів.

У роботі використано аналітичний підхід для опрацювання наявних рішень повнотекстового пошуку, методи структурно-функціонального моделювання для побудови архітектури модуля та проектування взаємодії його складових, а також об'єктно-орієнтовані методи для реалізації програмної частини і структури модулю в середовищі ODOO. Системний аналіз застосовано для перевірки відповідності модулю вимогам щодо доступу, узгодженості даних та коректності роботи під навантаженням, а експериментальний підхід – для тестування індексації, обробки запитів і формування результатів у реальних умовах.

Отримані результати мають значну практичну цінність, оскільки розроблений модуль забезпечує глобальний пошук по всій базі ODOO, враховує права доступу користувачів і мультикомпанійність, не потребує зовнішніх сервісів та може бути інтегрований у будь-яке середовище ODOO без додаткових складнощів

На основі матеріалів роботи підготовлено дві тези доповідей на всеукраїнській та міжнародній конференціях та одна стаття у науковому виданні, включеному до Переліку фахових видань України категорії Б [6], [7].

1 ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ЕФЕКТИВНОГО МОДУЛЯ ПОВНОТЕКСТОВОГО ПОШУКУ ДЛЯ ODOO

Система ODOO, як одна з провідних ERP-платформ, є важливим інструментом в автоматизації бізнес-процесів різних типів підприємств. За для забезпечення ефективного управління система має оперативно керувати великими масивами даних та швидко знаходити необхідну інформацію.

1.1 Мета і завдання роботи

Основна мета роботи полягає у створенні масштабованої та високопродуктивної системи повнотекстового пошуку для ODOO, яка забезпечуватиме швидкий, точний і гнучкий доступ до даних. Ідея проєкту полягає не лише у підвищенні швидкодії пошуку, а й у простій інтеграції з архітектурою ODOO, використовуючи лише можливості PostgreSQL. Модуль має працювати стабільно навіть за умови роботи з величезними обсягами інформації, не втрачаючи ефективності.

Розробка такого модуля має низку переваг. Інтеграція ефективного пошукового інструменту значно скоротить час, необхідний користувачам для знаходження потрібної інформації, підвищуючи їх продуктивність і комфорт роботи. Використання сучасних технологій таких як стемінг та токенізація дозволить отримувати більш точні результати навіть для складних або неоднозначних запитів [8]. Крім того, система передбачатиме динамічне фільтрування, логічні оператори та інші інструменти, що дадуть змогу адаптувати пошук під індивідуальні потреби користувача.

Використання PostgreSQL забезпечує надійну індексацію та високу швидкодію запитів. Завдяки цьому система здатна підтримувати стабільну роботу навіть за умов

значного зростання обсягу даних. Особливо важливо, що запропонована архітектура легко інтегрується в ODOO без необхідності масштабних змін у базовій структурі, що робить її впровадження простим та ефективним. Це дозволяє зберегти сумісність із поточними бізнес-процесами, водночас розширюючи функціональні можливості системи.

Розробка модуля включає кілька взаємопов'язаних етапів таких як аналіз існуючих підходів до побудови архітектури, вибір оптимальних технологій і розробки рішень для ефективної обробки великих масивів інформації. Після реалізації ключових компонентів здійснюється тестування для перевірки стабільності, точності та швидкодії пошуку. Особлива увага приділяється тому, щоб система не лише працювала швидко, але й залишалася надійною в умовах інтенсивного навантаження.

У результаті реалізації проєкту формується потужний інструмент, що суттєво розширює можливості ODOO у сфері пошуку та управління даними. Завдяки новій системі користувачі отримують змогу миттєво знаходити потрібну інформацію, підвищувати ефективність роботи та зберігати високу продуктивність навіть при взаємодії з великими обсягами даних.

1.2 Особливості системи ODOO

Система ODOO є сучасною інтегрованою ERP-системою (Enterprise Resource Planning), яка слугує універсальною платформою для комплексного управління бізнес-процесами. Її архітектура побудована за модульним принципом, що дозволяє поєднувати широкий спектр функціональних можливостей у межах єдиного середовища. Система охоплює практично всі ключові напрями діяльності підприємства: від управління продажами, бухгалтерського та фінансового обліку до контролю виробництва, логістики, складського господарства, управління персоналом, маркетингових кампаній і навіть внутрішніх комунікацій.

На рисунку 1.1 представлено головну сторінку ODOO, яка ілюструє структуру інтерфейсу користувача та різноманіття доступних модулів. Візуальне оформлення системи вирізняється сучасним дизайном, логічною організацією елементів і високим

рівнем інтуїтивності, що полегшує роботу користувача навіть без спеціальної технічної підготовки.



Рисунок 1.1 – Головний екран системи ODOO

Завдяки гнучкості архітектури ODOO може адаптуватися під потреби як невеликих компаній, так і масштабних корпорацій, що мають складну структуру управління та значні обсяги даних.

Однією з найсуттєвіших переваг системи є її відкрита архітектура, яка надає можливість не лише підключати нові модулі, а й модифікувати або розширювати існуючі без необхідності глибокого втручання в ядро програми. Такий підхід забезпечує масштабованість і гнучкість — характеристики, що особливо важливі для підприємств, які перебувають у стані динамічного розвитку або мають нестандартні бізнес-процеси.

Високий ступінь інтегрованості функціональних підсистем сприяє автоматизації більшості операцій, мінімізує людський фактор і зменшує ризики, пов'язані з ручним введенням даних. Єдина база інформації дозволяє уникати

дублювання, забезпечує узгодженість показників між різними модулями та сприяє безперервному інформаційному обміну між структурними підрозділами підприємства. У результаті цього підвищується точність управлінських рішень і загальна ефективність бізнес-операцій.

Окремо варто відзначити розвинену спільноту користувачів і розробників, яка постійно підтримує еволюцію ODOO. Відкрита екосистема проєкту сприяє створенню нових функціональних модулів, розробці розширень, адаптації системи під специфічні потреби окремих галузей і компаній. Активна взаємодія між учасниками спільноти забезпечує постійне вдосконалення програмного продукту, стабільні оновлення, обмін досвідом і високий рівень технічної підтримки. Завдяки цьому ODOO залишається не лише конкурентоспроможною ERP-платформою, а й одним із найдинамічніших рішень на ринку корпоративного програмного забезпечення.

1.3 Визначення функціональних вимог до модуля пошуку

Розробка вдосконаленого модуля повнотекстового пошуку має забезпечити користувачів зручним на ефективним інструментом для знаходження необхідної інформації. Особлива увага зосереджена на універсальності і гнучкості використання.

Для досягнення поставлених цілей було сформовано наступний список функціональних вимог:

- модуль повинен забезпечувати можливість виконання запитів як за окремими словами, так і за точними фразами, охоплюючи всі наявні модулі ODOO;
- забезпечити можливість використання логічних операторів для комбінації пошукових запитів;
- підтримка фільтрації результатів за моделями, в яких вони були знайдені;
- забезпечити підтримку режиму мультикомпанійності;
- забезпечити швидкий доступ до пошукового модуля з будь-якого місця в системі.

1.4 Технологічні Вимоги

Нефункціональні вимоги визначають характеристики системи, які не описують її безпосередні функції, але суттєво впливають на якість, продуктивність і стабільність роботи модуля. Вони встановлюють обмеження та технічні параметри, що забезпечують надійну інтеграцію рішення з ODOO, його ефективну взаємодію з базою даних PostgreSQL і коректну роботу навіть у середовищі з великими обсягами інформації.

До основних нефункціональних вимог належать:

- модуль повинен бути сумісний з останніми стабільними версіями ODOO з можливістю інтеграції до наявних модулів в системі без значних змін в архітектурі;
- модуль має зберігати свою функціональність при роботі з великими базами даних та не переривати роботу системи під час встановлення;
- забезпечити роботу пошуку покладаючись на доступні в системі інструментів, таких як PostgreSQL, без потреби в додаткових сервісів;
- врахувати права доступу різних користувачів до читання записів;
- забезпечити постійну підтримку актуальності даних у фоновому режимі без блокування доступу користувача до системи.

Описані вимоги забезпечать просту та інтуїтивно зрозумілу взаємодію користувача з модулем гарантуючи безпеку стабільність на надійну роботу навіть при великих навантаженнях.

Висновки до розділу 1

У першому розділі було сформовано підґрунтя для подальшої розробки модуля повнотекстового пошуку в системі ODOO. Визначено мету роботи та сформульовано завдання, які охоплюють аналіз існуючих підходів, вибір інструментів розробки, побудову архітектури та створення ефективного інструменту пошуку.

Окрему увагу приділено особливостям системи ODOO, що визначають специфіку реалізації повнотекстового пошуку в її середовищі: значна кількість

моделей, велика варіативність типів даних і залежність результатів від прав доступу користувача. Це підкреслює необхідність створення рішення, яке враховує структуру ERP-платформи, забезпечує стабільну роботу незалежно від обсягу інформації та інтегрується з існуючими можливостями PostgreSQL.

Сформульовані функціональні та нефункціональні вимоги дозволили окреслити ключові критерії майбутнього програмного продукту, серед яких — стабільність роботи з великими базами даних, підтримка актуальності інформації, відповідність політикам доступу користувачів та відсутність залежності від зовнішніх сервісів.

Таким чином, у розділі 1 було визначено концептуальні засади, на яких ґрунтується подальше проєктування і реалізація модуля повнотекстового пошуку в ODOO, що забезпечує логічний перехід до аналізу існуючих рішень у наступному розділі.

2 АНАЛІЗ СУЧАСНИХ РІШЕНЬ ВПРОВАДЖЕННЯ ПОВНОТЕКСТОВОГО ПОШУКУ

Поступово повнотекстовий пошук перетворюється з додаткового просунутого функціоналу на необхідність, стандартний інструмент в роботі. Враховуючи обсяг даних з яким стикаються сучасні бізнес системи необхідність швидко орієнтуватись в об'ємах інформації стає вирішальною в успішності корпорацій. Створюються окремі сервіси, додаються розширення до модулів в системах чи окремі функції в середині самих систем управління базами даних. Далі буде розглянуто деякі підходи до реалізації повнотекстового пошуку.

2.1 Спеціалізовані інструменти

Спеціалізовані пошукові платформи розроблені для індексації та обробки великих масивів даних. Їхня головна перевага в тому, що здійснення пошуку винесено за межі основної бази даних, що зменшує навантаження на саму інформаційну систему. Такі рішення впроваджуються через власне API, зазвичай підтримують кластеризацію, використання логічних операторів, ранжування результатів та масштабування. Завдяки цьому, вони можуть бути інтегровані з різними ERP-, CRM- чи іншими бізнес системами.

2.1.1 Платформа Elasticsearch

Платформа Elasticsearch – це розподілена, відкрита, пошукова система і аналітики, для обробки великих обсягів даних. Побудована на бібліотеці Apache Lucene, і є основною частиною пакету інструментів Elastic Stack. До складу входять такі сервіси як: Logstash для обробки даних, Kibana для візуалізації, Beats для збору даних, та інші [9], [10]. Сторінку інтерфейсної системи зображено на рисунку 2.1.

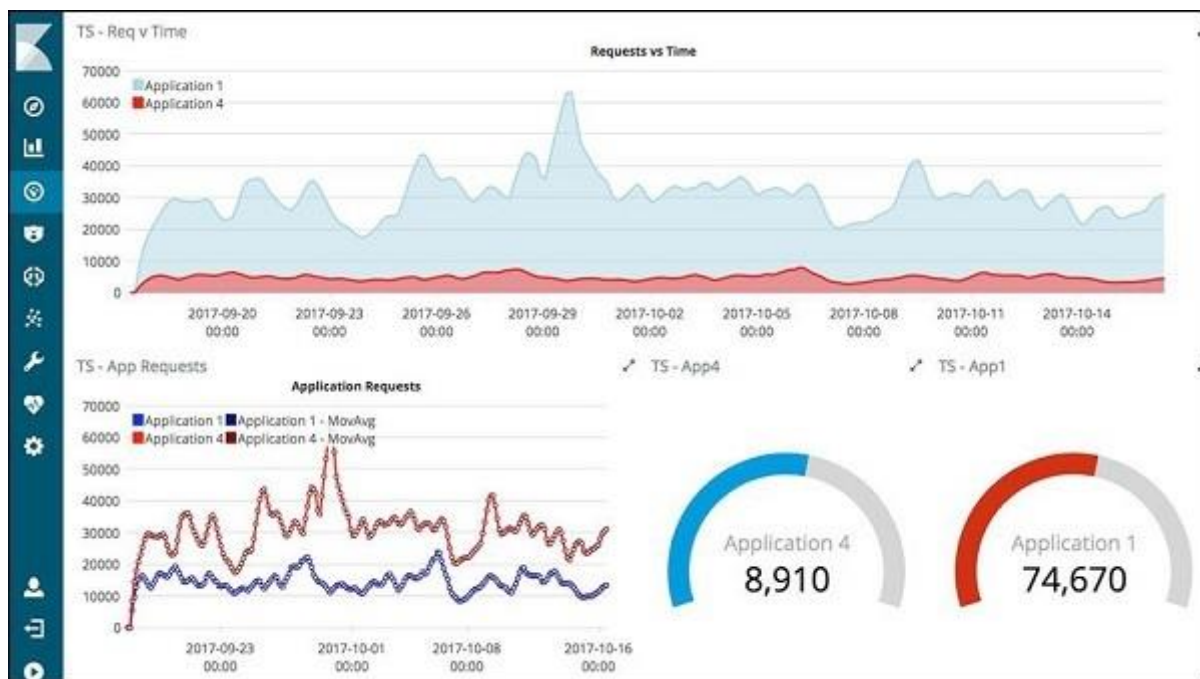


Рисунок 2.1 – Інтерфейс системи Kibana

Продукти Elasticsearch призначені для спільного використання, а їхні випуски синхронізовані для спрощення процесу інсталяції та оновлення.

Є декілька варіантів розгортання Elastic Stack відповідно до потреб користувача. Існує можливість розгорнути його на власному обладнанні, у хмарі або скористатися керованою послугою на Elastic Cloud.

Система Elasticsearch є як NoSQL базою даних, так і пошуковим двигуном, який дозволяє зберігати, індексувати та шукати дані у форматі JSON. Підтримує горизонтальне масштабування, що робить його ефективним для роботи з великими обсягами даних.

Дані в Elasticsearch серіалізуються та зберігаються у вигляді документів JSON. Документ – це набір полів, які є парами ключ-значення, що містять дані. Кожен документ має унікальний ідентифікатор, який можна створити самостійно або автоматично згенерувати. Документи групуються в індекси, які є логічними контейнерами, подібними до баз даних у реляційних СУБД. Кожен індекс має унікальну назву і може містити документи з подібною структурою. Індекси діляться на шарди, що дозволяє розподілити дані між вузлами кластера для паралельної обробки та масштабованості. Для забезпечення відмовостійкості

кожен первинний шард може мати одну або кілька реплік, що зберігаються в різних вузлах кластеру. Таким чином, навіть при виникненні збою в одному з вузлів, кластер продовжить роботу.

Серед переваг використання Elasticsearch варто виділити швидкість роботи. Завдяки інвертованому індексу пошук за ключовими словами в мільйонах документів виконується за мілісекунди. Також підтримуються складні аналітичні запити, як агрегація, що робить ідеальним рішенням для моніторингу та бізнес-аналітики.

Ще одним плюсом є розподілена архітектура. Як було зазначено вище, Elasticsearch працює як кластер із вузлів, що дозволяє горизонтально масштабувати систему шляхом додавання нових серверів. Автоматичне балансування навантаження оптимізує розподіл шардів, що зменшує ризик перевантаження окремих вузлів.

Крім того, Elasticsearch має гнучкий інструментарій запитів, який дозволяє комбінувати пошукові умови, використовувати логічні оператори, фільтри, розширені алгоритми ранжування, синоніми, автодоповнення, пошук за частинами слова та інші корисні можливості.

Базова версія Elasticsearch є безкоштовною та з відкритим кодом та гарною документацією. Велика спільнота розробників та наявність спеціалізованих курсів для користування і налаштування полегшує опанування платформою та допомагає вирішувати проблеми, коли такі виникають.

Однак, важливо зазначити недоліки, що можуть стати критичним при виборі інструменту для реалізації повнотекстового пошуку. Найбільший з них – складність налаштування та управління. Платформа потребує окремого серверного середовища, налаштування кластерів, моніторингу продуктивності та регулярного обслуговування. Наприклад, неправильний розподіл шардів може призвести до нерівномірного навантаження, що, в свою чергу, призведе до збоїв в роботі.

До того ж, проблемою може стати і високе споживання ресурсів. Elasticsearch активно використовує оперативну пам'ять, особливо для кешування та індексації, що робить його вимогливим до апаратного забезпечення. У невеликих компаній це може викликати додаткові витрати або потребу модернізувати

обладнання.

Таким чином, Elasticsearch є потужним інструментом для повнотекстового пошуку. Його гнучкість, масштабованість і інтеграція з Elastic Stack забезпечують широке застосування в різних галузях, від e-commerce до DevOps. Однак, для ефективного його використання необхідно інвестувати час у навчання, планування архітектури та оптимізацію.

2.1.2 Платформа Apache Solr

Пошуковий сервер Apache Solr – це готова до використання у продакшн-середовищі платформа без необхідності додаткового програмування базової інфраструктури [11], [12]. В основі лежить ядро Apache Lucene, яке відповідає за створення та обслуговування інвертованого індексу, сегментне зберігання даних, алгоритми ранжування та злиття сегментів. Ця система відрізняється від самої бібліотеки тим, що є готовим до використання пошуковим сервером з повноцінною інфраструктурою, а саме з HTTP API, веб-інтерфейсом адміністратора, системою плагінів, алгоритмами реплікації, відмовостійкості та горизонтального масштабування.

Інтерфейс цієї системи наведено на рисунку 2.2.

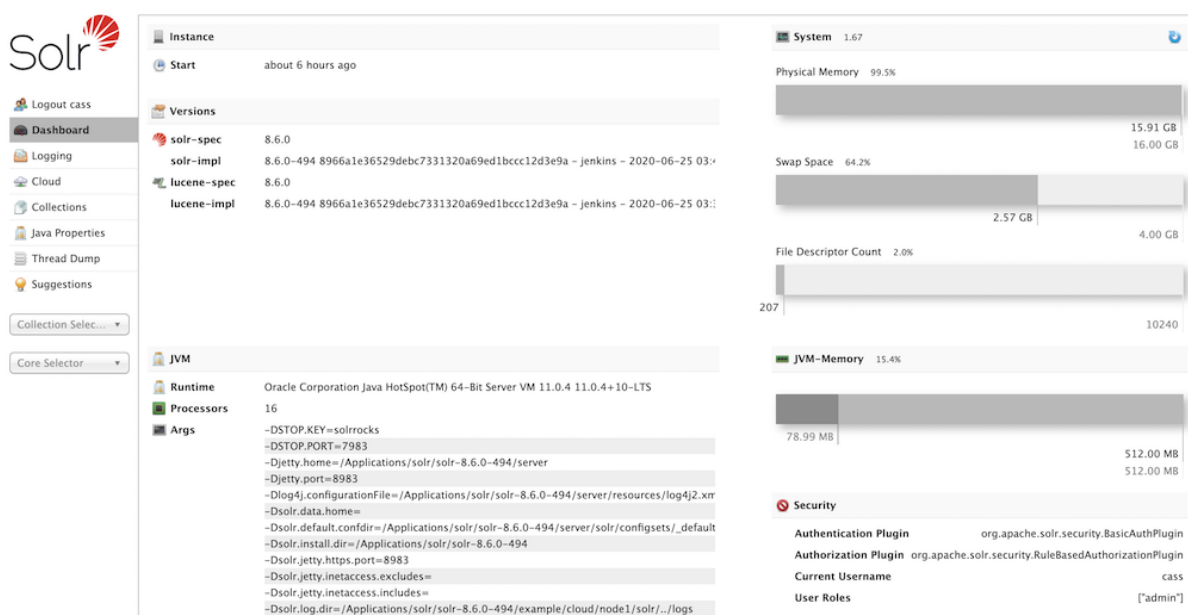


Рисунок 2.2 – Інтерфейс системи Solr

Система Solr може працювати у двох основних режимах: Standalone та SolrCloud. Перший є одновузловим режимом, найпростіший для початку роботи. Один екземпляр Solr, що зазвичай працює на вбудованому веб-сервері Jetty, обслуговує одну або кілька колекцій. Весь індекс лежить на одному сервері. Підходить для невеликих та середніх проєктів. Інший варіант роботи – розподілений кластерний режим, який підійде для великих корпорацій, бо здатний горизонтально масштабуватись до сотень серверів і трильйонів документів.

Однією з ключових переваг Solr є його здатність забезпечувати високу швидкість та масштабованість. Система оптимізована для індексації та пошуку в масивах, що можуть налічувати мільярди документів, а завдяки розподіленій архітектурі SolrCloud підтримує горизонтальне масштабування: нові вузли можуть додаватися без зупинки роботи, після чого шарди та репліки автоматично рівномірно розподіляються між серверами. Цю технічну гнучкість доповнює широкий набір функцій, який у багатьох випадках недоступний у традиційних реляційних базах даних. Solr підтримує фасетний пошук, підсвічування знайдених фрагментів, перевірку правопису, автодоповнення, групування результатів, розвинуті алгоритми ранжування, геопошук та інші функції, які роблять його універсальним інструментом для роботи з текстом.

Платформа є досить гнучкою, користувач може визначати різні типи полів, створювати їх динамічні варіанти, використовувати аналізатори для різних мов, включно з українською, а також формувати власні функції ранжування. Крім того, Solr має відкритий вихідний код, активно підтримується спільнотою Apache Software Foundation і регулярно оновлюється. Екосистема включає численні плагіни, бібліотеки та інтеграції для різних типів даних і сценаріїв використання. Роботу з платформою полегшує зручний вебінтерфейс адміністрування, REST-подібний JSON API, а також інтеграція з популярними системами моніторингу, такими як Grafana чи Prometheus. Solr здатен працювати з різними форматами даних і легко інтегрується з Hadoop, Spark, Kafka, SQL-базами та навіть файлами типу PDF або Office-документів завдяки використанню Apache Tika.

Попри численні переваги, Solr має й певні недоліки. Насамперед це складність розгортання та експлуатації. Робота з SolrCloud потребує значного

досвіду DevOps, оскільки помилки в налаштуванні кластерів, ZooKeeper або схеми шардування можуть серйозно вплинути на продуктивність. Платформа також вимоглива до ресурсів: більша частина індексу зберігається в оперативній пам'яті, тому для великих розгортань необхідні сервери з великою кількістю RAM. Окремою проблемою може стати налаштування релевантності — хоча Solr пропонує надзвичайно широкі можливості, їхня ефективність часто залежить від глибокого розуміння внутрішньої роботи Lucene, а без відповідного досвіду результат може бути менш точним, ніж у більш «автоматизованих» систем.

Деякі сучасні функції також впроваджувалися в Solr із затримкою. Наприклад, підтримка векторного пошуку та kNN-алгоритмів з'явилася лише в гілці 9.x, у той час як конкуренти пропонували подібні можливості раніше. Крім того, оновлення між основними версіями нерідко потребують повної переіндексації, що ускладнює модернізацію існуючих інсталяцій. Для нових команд Solr може виявитися недостатньо “готовим до виробництва”, оскільки потребує окремого налаштування безпеки, резервного копіювання й моніторингу, на відміну від хмарних аналогів, що надають усе це з коробки.

У підсумку Apache Solr залишається одним із найпотужніших і найгнучкіших інструментів для реалізації повнотекстового пошуку, особливо корисним у великих корпоративних середовищах, де важливо мати повний контроль над індексацією, релевантністю та інфраструктурою. Разом із тим платформа потребує значної технічної експертизи та ресурсів, що робить її менш придатною для невеликих або середніх проєктів [13].

2.1.3 Порівняння платформ Elasticsearch та Apache Solr

Elasticsearch і Apache Solr належать до одного класу рішень – спеціалізованих платформ для повнотекстового пошуку, створених на основі Apache Lucene. Незважаючи на спільну технологічну основу, підходи цих систем до індексації, масштабування та адміністрування суттєво відрізняються. Саме ці відмінності визначають, коли доцільно використовувати одне чи інше рішення.

Elasticsearch зазвичай сприймається як більш сучасний і гнучкий інструмент, орієнтований на простоту інтеграції та масштабування. Його архітектура спочатку

проектувалася як розподілена, тому створення кластерів, додавання нових вузлів і автоматичний перерозподіл індексів є частиною базового функціоналу. Elasticsearch також пропонує готові інструменти для шпаркого підключення до вебсервісів завдяки REST API та має розвинуту екосистему інструментів, серед яких особливо популярний ELK-стек для логування й аналітики. Завдяки цьому Elasticsearch часто використовують у системах, де важливо швидко додавати нові джерела даних, обробляти значні обсяги тексту в режимі реального часу або працювати з динамічними, постійно оновлюваними індексами.

Solr, у свою чергу, традиційно вважається більш стабільним та «консервативним» рішенням, яке довгий час було стандартом у корпоративних середовищах. Його сильна сторона — глибока керованість і можливість детально налаштовувати схему індексації, аналізатори, текстову обробку та логіку ранжування. Платформа пропонує більш формальний та структурований підхід до конфігурації, що може бути перевагою у системах, де критично важливою є передбачуваність та строгість роботи. Solr відзначають також за сильний фасетний пошук і розвинені інструменти кешування, що робить його особливо ефективним у складних каталогах або аналітичних середовищах.

Різниця між платформами помітна й у питанні складності адміністрування. Elasticsearch зазвичай простіший у початковому розгортанні, але водночас у великих кластерах може вимагати ретельного моніторингу для запобігання перевантаженням. Solr у режимі SolrCloud, навпаки, потребує значно більше попередньої підготовки і глибшого розуміння кластерної архітектури, особливо при роботі із ZooKeeper або при переході між основними версіями.

У контексті ресурсів обидві платформи є вимогливими до оперативної пам'яті, однак Solr частіше потребує значних резервів RAM через особливості роботи з DocValues та інвертованими індексами. Elasticsearch у цьому плані трохи гнучкіший, але теж вимогливий при високих навантаженнях, особливо в сценаріях активної аналітики логів [14].

З точки зору розвитку функціональності, Elasticsearch швидше адаптується до сучасних тенденцій. Підтримка векторного пошуку, семантичних моделей та інтеграції з ML-інструментами з'явилася в ньому раніше, тоді як Solr впроваджував

ці можливості із помітним запізненням. Проте Solr продовжує активно розвиватися й залишається сильним варіантом для проєктів, де перевага надається відкритості, керованості та можливості повністю контролювати інфраструктуру без прив'язки до конкретного вендора.

Загалом, Elasticsearch найкраще підходить для динамічних, розподілених систем, де важливі масштабованість, швидкість оновлення індексів та простота інтеграції. Apache Solr, у свою чергу, залишається оптимальним вибором у проєктах, де потрібне тонке налаштування, висока передбачуваність роботи, строгий контроль над релевантністю та інтеграція у класичні корпоративні середовища. У виборі між цими платформами вирішальним фактором зазвичай стають вимоги до гнучкості, ресурсоемності, складності адміністрування та специфіки бізнес-завдань.

2.2 Повнотекстовий пошук у сучасних ERP-системах

У багатьох корпоративних платформах повнотекстовий пошук є невід'ємною частиною загальної архітектури системи, адже саме вони забезпечують швидкий і точний доступ до бізнес-даних, що зберігаються в різних модулях ERP. На відміну від спеціалізованих пошукових платформ, які працюють як зовнішні сервіси, ERP-системи зазвичай реалізують пошук у тісному зв'язку зі своєю внутрішньою структурою, бізнес-логікою та моделями даних. Такий підхід дозволяє отримати максимально узгоджені результати, враховуючи роль користувача, права доступу, типи бізнес-об'єктів та специфіку їх взаємодії.

Сучасні ERP-рішення дедалі більше переходять до використання розширених інструментів індексації, машинного ранжування та семантичного пошуку, інтегруючи можливості хмарних платформ і аналітичних сервісів. Серед найпоширеніших прикладів можна навести Microsoft Dynamics 365 та Oracle ERP Cloud, які впроваджують власні пошукові підходи, орієнтовані на високу масштабованість, інтелектуальну обробку запитів і швидкий доступ до великих корпоративних сховищ даних.

У цьому підрозділі розглянуто принципи роботи повнотекстового пошуку в зазначених ERP-системах, особливості їхніх технологічних рішень та ті можливості, які вони пропонують для ефективної роботи користувачів у складних бізнес-середовищах.

2.2.1 Система Microsoft Dynamics 365

Система Microsoft Dynamics 365 є однією з найпоширеніших хмарних ERP- та CRM-платформ, яка поєднує модульну архітектуру, глибоку інтеграцію з іншими сервісами компанії Microsoft та розширені можливості для автоматизації бізнес-процесів. Система побудована за принципом єдиної екосистеми, де окремі додатки, такі як фінанси, продажі, маркетинг, управління персоналом, логістика чи виробництво, можуть працювати як незалежні модулі або як частини цілісної платформи. Dynamics 365 використовує хмарну інфраструктуру Microsoft Azure, що забезпечує гнучкість масштабування, високу доступність та безпечне зберігання даних у розподіленому середовищі [15].

Однією з характерних рис Dynamics 365 є тісна взаємодія з іншими продуктами Microsoft, як Office 365, Teams, Power BI, Power Automate та Dataverse. Це дозволяє організувати роботу користувачів у межах єдиного інформаційного простору, де дані не просто зберігаються, а активно циркулюють між модулями та сервісами. Центральним елементом архітектури виступає Dataverse, яка виконує роль універсального сховища бізнес-об'єктів. Саме на базі Dataverse реалізовані ключові алгоритми пошуку, аналітики й взаємодії з бізнес-логікою.

Повнотекстовий пошук у Dynamics 365 побудований на основі Azure Cognitive Search, який інтегрований у платформу на рівні інфраструктури. Цей сервіс забезпечує індексацію даних із різних модулів ERP та CRM і виконує обробку запитів за допомогою спеціалізованих текстових аналізаторів. Однією з головних переваг цього підходу є можливість працювати з великими розподіленими обсягами даних без навантаження на основну базу, оскільки пошуковий інструмент функціонує як окремий мікросервіс у хмарі Azure [16].

У Dynamics 365 пошук організовано таким чином, що кожен бізнес-об'єкт — наприклад, клієнт, документ продажу чи запис про транзакцію — проходить

попередню обробку та індексацію, після чого може бути знайдений за частиною слова, за фразою або на основі контекстного аналізу. Система підтримує не лише точний збіг, а й семантичний пошук, який враховує синоніми, морфологічні форми, вагові коефіцієнти й навіть поведінку користувачів під час роботи з даними.

Завдяки інтеграції з Azure Cognitive Search Dynamics 365 пропонує розширений функціонал, який виходить за межі класичного повнотекстового пошуку. Наприклад, система підтримує природно-мовні запити, автоматичне виправлення помилок, ранжування результатів за релевантністю, пошук за атрибутами об'єктів та інтелектуальне групування знайдених даних. На рисунку 2.3 продемонстровано архітектуру такого пошуку. Описані можливості роблять пошук не просто інструментом для знаходження записів, а важливою складовою аналітичної роботи користувачів, особливо у великих організаціях, де обсяги інформації є значними.

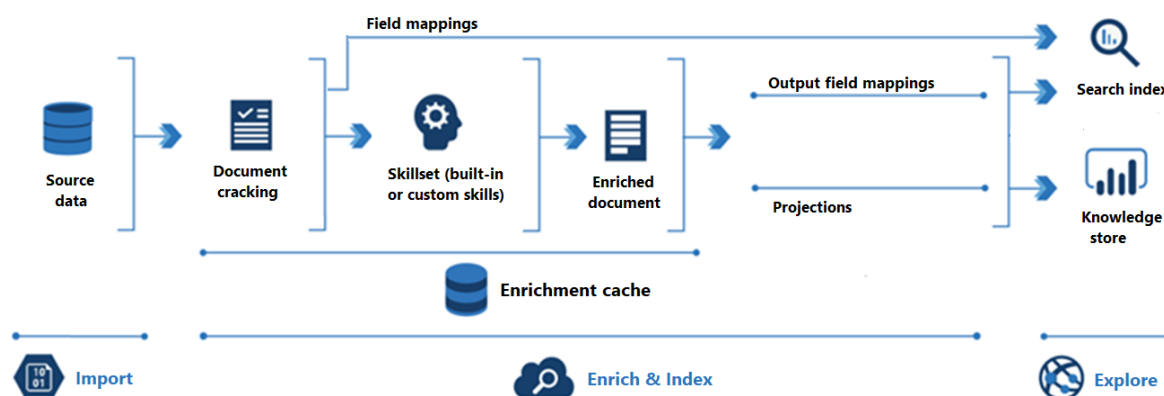


Рисунок 2.3 – Схема когнітивного пошуку Azure Cognitive Search

Загалом, реалізація повнотекстового пошуку в Dynamics 365 демонструє поєднання корпоративної надійності, хмарної масштабованості та інтелектуальних алгоритмів обробки тексту. Система орієнтована на забезпечення високої швидкодії та максимальної точності результатів, а інтеграція з Azure дозволяє використовувати додаткові інструменти машинного навчання та семантичного аналізу. Це робить Dynamics 365 одним із найбільш гнучких та технологічно розвинених рішень у сфері корпоративного пошуку.

2.2.2 Система Oracle ERP Cloud

Хмарна платформа Oracle ERP Cloud підходить для управління фінансами, ланцюгами постачання, закупівлями, виробництвом і аналітикою, вона поєднує широкі функціональні можливості з високим рівнем масштабованості та автоматизації бізнес-процесів. Система належить до класу корпоративних рішень, орієнтованих на великі організації, що працюють у динамічному середовищі та потребують централізованої обробки великих обсягів структурованих і неструктурованих даних. Архітектура Oracle ERP Cloud тісно пов'язана з інфраструктурою Oracle Cloud Infrastructure (OCI), що забезпечує високу доступність, безпеку та стабільність роботи системи навіть у мультинаціональних корпораціях з глобально розподіленими даними [17].

Однією з ключових складових платформи є Oracle Fusion Applications, у межах яких реалізовано наскрізну бізнес-логіку, модульну організацію даних та можливість інтеграції з інструментами Oracle Analytics, Oracle Integration Cloud і базами даних Oracle Autonomous Database. Усі ці компоненти працюють разом, формуючи єдиний простір даних, де інформація циркулює між фінансовими, операційними та аналітичними модулями. Завдяки цьому пошук у системі охоплює не лише окремі бізнес-об'єкти, а й складні зв'язки між ними.

Повнотекстовий пошук в Oracle ERP Cloud базується на технологіях Oracle Text та засобах індексації, які підтримує базова реляційна база даних Oracle Database. Oracle Text забезпечує повноцінний інструмент обробки текстових даних за допомогою інверсних індексів, морфологічного аналізу, токенізації, підтримки стоп-слів і тематичного групування, що дозволяє працювати не лише з окремими полями, а й з великими документами, файлами та текстовими атрибутами бізнес-об'єктів.

На відміну від багатьох інших ERP-рішень, Oracle ERP Cloud використовує пошукові алгоритми, глибоко інтегровані у структуру самої бази даних, що забезпечує високу узгодженість результатів, дотримання політик доступу й коректне відображення міжмодульних зв'язків. Усі процеси індексації виконуються безпосередньо в Oracle Database, тому дані практично не мають ризику бути застарілими або десинхронізованими, як це іноді трапляється у системах зі зовнішніми пошуковими серверами.

Пошук у системі Oracle ERP Cloud підтримує широкий набір можливостей, включаючи контекстний аналіз, пошук за частинами слова, обробку фраз, лінгвістичні правила для різних мов та ранжування на основі релевантності. Крім того, Oracle Text у поєднанні з Oracle Secure Enterprise Search дозволяє здійснювати пошук одночасно в структурованих і неструктурованих даних, охоплюючи як записи ERP, так і документи, вкладення, PDF-файли чи корпоративні сховища.

Додатковою перевагою платформи є інтеграція можливостей машинного навчання та аналітики. За допомогою Oracle Machine Learning система може виконувати семантичний пошук, автоматично розпізнавати теми текстів, групувати результати та формувати інтелектуальні рекомендації. Це робить пошук не лише інструментом навігації, а й засобом підтримки прийняття рішень, що особливо цінно в складних операційних процесах.

Разом із тим високий рівень інтеграції означає, що Oracle ERP Cloud значною мірою залежить від інфраструктури Oracle Database і OCI. Для великих підприємств це зазвичай є перевагою, однак для невеликих компаній така екосистема може бути надто ресурсомісткою. Крім того, складність архітектури та висока ступінь автоматизації процесів вимагають спеціалізованої експертизи під час впровадження та налаштування системи.

У цілому повнотекстовий пошук у Oracle ERP Cloud поєднує надійність реляційної бази даних, гнучкість Oracle Text і сучасні інструменти аналітики, створюючи ефективне середовище для роботи з великими масивами корпоративної інформації. Система вирізняється точністю, комплексністю та високим рівнем інтегрованості, що робить її одним із найбільш зрілих рішень у сфері корпоративного пошуку.

2.3 Модулі повнотекстового пошуку в системі ODOO

У спільноті ODOO існує значна кількість рішень, спрямованих на розширення базових можливостей пошуку, подолання обмежень стандартного ORM та підвищення швидкодії роботи з великими обсягами даних. Різні модулі

пропонують різні підходи — від інтеграції з зовнішніми пошуковими рушіями до використання розширених інструментів PostgreSQL. У цьому розділі розглянуто найпоширеніші підходи до реалізації повнотекстового пошуку в ODOO, а також проаналізовано два універсальні модулі, які не вимагають підключення сторонніх сервісів.

2.3.1 Підходи до реалізації

В екосистемі ODOO склалося три принципово різні підходи до реалізації повнотекстового пошуку, кожен з яких по-своєму вирішує задачу балансу між швидкістю, простотою розгортання та відсутністю зовнішніх залежностей.

Перший підхід передбачає використання зовнішніх спеціалізованих пошукових рушіїв, насамперед Elasticsearch. У цьому випадку ODOO виступає лише конектором: модулі синхронізують дані з зовнішнім індексом і передають запити на обробку сторонньому серверу. Серед прикладів такої реалізації можна виокремити Elastic Stack Connector, OCA connector_elasticsearch та wk_backend_elasticsearch. Такий варіант забезпечує найвищу швидкість і практично необмежену масштабовість, але вимагає окремого серверного середовища та додаткового адміністрування [18].

Другий підхід — це локальні модулі, орієнтовані на конкретні підсистеми. Вони, як правило, використовують вбудований Full Text Search PostgreSQL і створюють індекси лише в межах окремих моделей (наприклад, проекти, контакти, форуми). На такому підході є `fts_base` з похідними модулями та `website_forum_fulltext_search`. Такі рішення не потребують зовнішніх сервісів і легко встановлюються, проте не формують єдиного глобального пошуку, а більшість із них давно не оновлювалась і залишилася сумісною лише зі старими версіями ODOO.

Третій підхід — сучасні універсальні внутрішні рішення, які будують єдиний централізований індекс усередині самої бази PostgreSQL, автоматично охоплюють десятки моделей і забезпечують глобальний пошук без жодного зовнішнього сервісу. Саме цей напрям зараз вважається найперспективнішим для актуальних версій ODOO, оскільки повністю усуває зовнішні залежності й водночас зберігає прийнятну продуктивність при великих обсягах даних. До нього належать модулі Bloomip Fulltext Search та SH Global Search, які будуть розглянуті далі [19], [20].

2.3.2 Модуль Bloomup Fulltext Search

Модуль Bloomup Fulltext Search реалізує повнотекстовий пошук в ODOO на базі PostgreSQL, дозволяючи швидко налаштувати глобальний пошук по полях моделей та пов'язаних повідомленнях без потреби в зовнішніх сервісах чи програмуванні. Головна особливість — гнучке налаштування: для кожної моделі користувач вручну переходить у вкладку "Fulltext Search", вибирає конкретні поля (наприклад, назву чи опис), активує опцію "Search in Messages" для включення чатів і коментарів, а потім натискає кнопку "CREATE TSVECTOR COLUMN", щоб створити спеціальну колонку для індексації — цей процес може зайняти час на великих даних і вимагає повторення для кожної моделі окремо.

На рисунку 2.4 зображено скріншот з вкладкою налаштувань.

Model Description Helpdesk Ticket

Model helpdesk.ticket

Order [?] priority desc, id desc

Transient Model ☐

Has Mail Thread ☒

Has Mail Activity ☒

Has Mail Blacklist ☐

EN Type Base Object

In Apps [?] crm_helpdesk, helpdesk, helpdesk_sms

Fields Access Rights Record Rules Notes Views Website Forms Fulltext Search

Search in Messages [?] ☒

Field Name	Field Label	Model	Field Type	Type	Indexed	Stored	Readonly	Related Model
description	Description	Helpdesk Ticket	html	Base Field	☐	☒	☐	✕
name	Subject	Helpdesk Ticket	char	Base Field	☒	☒	☐	✕

Add a line

Create TsVector Column

Рисунок 2.4 – Вікно налаштувань модуля Bloomup Fulltext Search

Після налаштування пошук запускається через окреме меню "Fulltext Searcher App", де вибирається модель, вводиться запит (з підтримкою PostgreSQL-синтаксису на кшталт фраз у лапках, OR чи заперечень з мінусом), а результати виводяться списком з ранжуванням за релевантністю, де кожен рядок веде до запису чи повідомлення — щоправда, один ID може дублюватися, якщо пов'язано кілька повідомлень.

Переваги полягають у простоті інтеграції без коду та залежностей, ефективному ранжуванні результатів і природному інтерфейсі запитів, що робить його зручним для середніх систем з фокусом на комунікаціях. Однак недоліки очевидні: ручне внесення полів і створення індексу для кожної моделі ускладнює масштабування, можливі дублікати в результатах, а на гігантських обсягах даних продуктивність поступається спеціалізованим рушіям на кшталт Elasticsearch. Загалом, це компактне рішення для тих, хто цінує автономність, але готові до початкового налаштування.

2.3.3 Модуль SH Global Search

Модуль SH Global Search реалізує глобальний повнотекстовий пошук в ODOO, дозволяючи шукати по будь-яких моделях і полях без переходу до окремих меню — пошуковий рядок з'являється зверху в головному меню, доступний усім внутрішнім користувачам. Налаштування відбувається через окреме меню "Show Global Search Configuration", де для кожної моделі (наприклад, партнери чи продажі) вручну вибираються ключові поля (назва, референс, one-to-many зв'язки через майстер), активуються опції для пошуку в повідомленнях (тема, контент чи суб'єкт), а зміни застосовуються автоматично, з урахуванням прав доступу та мультимережовості (результати маркуються компаніями).

Пошук запускається простим введенням тексту, літер чи цифр у рядок — результати виводяться в новій вкладці з ранжуванням за релевантністю, диференціацією за моделями та компаніями, а клік веде безпосередньо до запису.

Для очищення використовується кнопка скасування, а опція "Enable Menu Global Search" додає пошук у підменю. Синтаксис базовий, без складних операторів PostgreSQL, але підтримує природний ввід для повідомлень і полів, з повагою до доступів користувача.

Вигляд результатів пошуку зображено на рисунку 2.5.

Переваги даного модуля в зручності: повна конфігурація без коду, швидкий доступ до будь-яких даних (включно з one-to-many), красивий інтерфейс результатів і підтримка мультикомпанійності, що робить його ідеальним для середніх бізнесів з активним використанням ODOO.

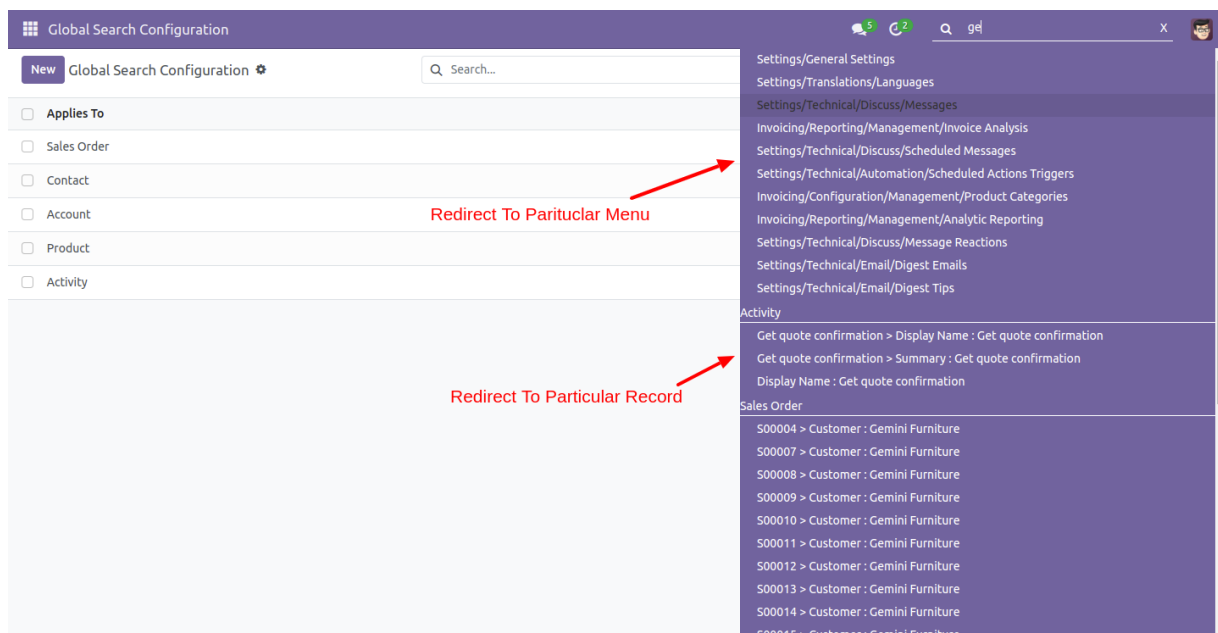


Рисунок 2.5 – Вікно налаштувань модуля Bloomup Fulltext Search

Однак недоліки включають несумісність з ODOO Online (SaaS), необхідність ручного налаштування полів для кожної моделі, що ускладнює початкове розгортання на великих системах, відсутність розширених синтаксичних можливостей чи автокомпліту. Загалом, це практичне рішення для глобального пошуку в локальних розгортаннях, але для складних сценаріїв може знадобитися доопрацювання.

Висновки до розділу 2

Проведений аналіз сучасних підходів до реалізації повнотекстового пошуку показує, що цей інструмент поступово перестав бути додатковою опцією і перетворився на один із ключових компонентів сучасних інформаційних систем, зокрема ERP-платформ.

Спеціалізовані платформи, такі як Elasticsearch та Apache Solr, демонструють максимальну гнучкість і продуктивність при роботі з великими масивами текстових даних. Вони забезпечують розподілену архітектуру, розвинуті засоби індексації, масштабування та аналітики, що робить їх придатними для високонавантажених розподілених середовищ. Водночас їх використання супроводжується підвищеною

складністю розгортання, необхідністю окремої інфраструктури, високими вимогами до ресурсів і потребою у спеціалізованій експертизі для підтримки та налаштування.

Реалізація повнотекстового пошуку у великих ERP-системах, таких як Microsoft Dynamics 365 та Oracle ERP Cloud, показує інший підхід – глибоку інтеграцію пошуку з внутрішньою архітектурою платформи та її хмарною інфраструктурою. Використання сервісів Azure Cognitive Search або Oracle Text дозволяє цим системам поєднувати повнотекстовий аналіз з бізнес-логікою, правами доступу, аналітикою та машинним навчанням.

Ситуація в екосистемі ODOO є більш різноманітною. З одного боку, існують конектори до зовнішніх пошукових рушіїв, які дозволяють використовувати можливості Elasticsearch, але водночас ускладнюють архітектуру та вимагають окремої інфраструктури. З іншого боку, тематичні модулі, засновані на PostgreSQL Full Text Search, показують ефективність у межах окремих підсистем, проте не формують єдиного глобального пошуку і часто обмежені за версіями. Сучасні універсальні рішення, такі як Bloomip Fulltext Search та SH Global Search, роблять важливий крок у напрямі глобального повнотекстового пошуку всередині ODOO без зовнішніх сервісів, однак також мають свої обмеження з погляду масштабованості, гнучкості налаштувань та автоматизації індексації у великих інсталяціях.

Отже, аналіз показує, що, попри наявність різних інструментів і підходів, все ще залишається ніша для спеціалізованого рішення, яке поєднувало б переваги вбудованого повнотекстового пошуку PostgreSQL з вимогами сучасних ERP-систем: підтримкою глобального пошуку по багатьох моделях, використанням логічних операторів і фразових запитів, стійкою роботою на великих базах даних та мінімальною залежністю від зовнішньої інфраструктури. Саме розробка такого модуля для ODOO і стає предметом подальшого дослідження в наступних розділах роботи.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У цьому розділі представлено практичну реалізацію програмного забезпечення, що забезпечує повнотекстовий пошук у системі ODOO. Розглянуто вибір інструментів розробки, принципи побудови архітектури модуля, процеси підготовки даних та первинного індексування, а також підходи до підтримання актуальності інформації під час експлуатації системи. Окрему увагу приділено обробці пошукових запитів, формуванню результатів з урахуванням прав доступу користувачів і мультикомпанійності, а також структурі бази даних та функціональним можливостям модуля. Представлені рішення формують повний цикл роботи системи та демонструють особливості її інтеграції в середовище ODOO.

3.1 Вибір засобів розробки

Для реалізації модуля використано технологічний стек, повністю збігається з архітектурою ODOO. Основною мовою програмування обрано Python, а системою управління базами даних – PostgreSQL, що є офіційно інструменти платформи. Це дозволило глибоко інтегрувати модуль у ядро ODOO, використовувати вбудовані інструменти повнотекстового пошуку, триграмний пошук та пакетні операції без залучення зовнішніх пошукових рушіїв.

Середовищем розробки став PyCharm, що забезпечує зручне відлагодження, підсвічування XML-шаблонів та роботу з ORM ODOO. Для перехоплення змін та організації асинхронної черги використано внутрішні методи ODOO ORM та стандартний інструмент запланованих дій. Такий вибір гарантує повну сумісність, мінімальну вартість володіння, простоту розгортання та високу продуктивність навіть на стандартному обладнанні, роблячи модуль доступним для будь-якої інсталяції ODOO Community чи Enterprise.

3.1.1 Мова програмування Python

Мова програмування Python була обрана як основний засіб розробки програмного забезпечення, оскільки вона є фундаментальною складовою серверної частини платформи ODOO та забезпечує оптимальні умови для створення модульних, розширюваних і підтримуваних рішень. Python належить до мов високого рівня абстракції, що вирізняється виразним, лаконічним синтаксисом, спрямованим на підвищення читабельності коду та зменшення кількості технічних помилок у процесі розробки. Завдяки цьому він особливо ефективний у сфері корпоративних інформаційних систем, де критичними є надійність бізнес-логіки, коректність обробки даних та стабільність інтеграційних процесів [21], [22].

Ключовою характеристикою Python є його здатність забезпечувати високий рівень продуктивності розробника: синтаксичні конструкції мови дозволяють стисло й однозначно формулювати логіку роботи системи, що значно полегшує як створення нових функціональних компонентів, так і подальшу їх підтримку. У контексті ODOO ця властивість має особливе значення, оскільки моделювання бізнес-об'єктів, опис залежностей, перевизначення методів та взаємодія з ORM-фреймворком вимагають максимальної чіткості та структурованості у визначенні поведінки системи.

Важливою перевагою Python є також наявність надзвичайно розгалуженої екосистеми бібліотек і фреймворків, що охоплюють практично всі сфери сучасної розробки: від роботи з базами даних і веб-технологіями до машинного навчання, обробки текстів і побудови інтерфейсів. Це дозволяє ефективно поєднувати можливості ODOO з додатковими інструментами, необхідними для реалізації алгоритмів повнотекстового пошуку, обробки текстових даних або взаємодії з PostgreSQL. Використання готових бібліотек скорочує час розробки, підвищує стабільність програмного продукту та забезпечує дотримання галузевих стандартів.

Оскільки архітектура ODOO повністю побудована навколо Python, вибір цієї мови для розробки нового модуля є технічно виправданим і необхідним. Інтеграція з внутрішнім устроєм платформи — моделями ORM, середовищем виконання, системами доступу, чергами завдань та інструментами автоматизації — відбувається природним чином, без додаткових адаптацій чи проміжних шарів. Це забезпечує повну сумісність розробленого модуля з існуючою інфраструктурою ODOO та дає

можливість уніфікувати логіку обробки даних у межах єдиної технологічної платформи.

У підсумку Python слід розглядати не лише як загальнопризначену мову програмування, а як ключовий елемент технологічної екосистеми ODOO, що забезпечує ефективність розробки, структурованість коду та можливість масштабування функціональності системи. Поєднання Python із потенціалом PostgreSQL створює надійну основу для впровадження продуктивного, гнучкого та інтегрованого модуля повнотекстового пошуку, який відповідає вимогам сучасних ERP-рішень.

3.1.2 Середовище розробки PyCharm

Інтегроване середовище розробки PyCharm, створене компанією JetBrains, є одним із найпотужніших інструментів для розробки програмного забезпечення мовою Python і широко застосовується у професійних проєктах, що передбачають високі вимоги до якості коду, масштабованості та організації програмної архітектури. Завдяки поєднанню інтелектуальних інструментів, розвинених засобів аналізу та зручних інтерфейсних рішень PyCharm забезпечує ефективну підтримку всіх основних етапів розробки — від написання та тестування коду до налагодження, роботи з залежностями та інтеграції зі сторонніми сервісами.

Однією з ключових переваг середовища є його високорівневий редактор коду, що підтримує автоматизовані процеси доповнення, статичний аналіз, виявлення помилок у реальному часі та рефакторинг. Інтелектуальна система підказок істотно прискорює написання коду і сприяє підвищенню його надійності, що є особливо важливим у контексті розробки модулів для ODOO, де зміни в логіці можуть впливати на значну кількість взаємопов'язаних компонентів.

PyCharm також містить повноцінний інтегрований засіб налагодження, який дає можливість встановлювати точки зупину, виконувати покрокове відлагодження, аналізувати стан змінних і відстежувати хід виконання програми. Використання вбудованого дебагера значно полегшує виявлення складних помилок у бізнес-логіці та алгоритмах обробки даних. Це є релевантним для проєктів, що передбачають взаємодію з ORM ODOO, складні процеси індексації або асинхронні операції.

Важливою перевагою PyCharm є глибока інтеграція з системами контролю версій, зокрема Git, GitHub та GitLab. Це дає змогу підтримувати структуровану командну роботу, керувати історією змін, проводити рев'ю коду та забезпечувати відтворюваність розробки. Інструменти для роботи з віртуальними середовищами Python (venv, pipenv, Poetry) дозволяють ізолювати залежності окремих проєктів та уникати конфліктів між бібліотеками, що є важливим під час розробки ODOO-модулів, які нерідко потребують специфічних пакетів.

Середовище PyCharm підтримує інтеграцію з низкою популярних Python-фреймворків і бібліотек, включаючи ті, що застосовуються у веб-розробці, наукових обчисленнях та обробці даних.

PyCharm доступний у двох редакціях: Community Edition, що містить базовий функціонал для Python-розробки, та Professional Edition, яка включає додаткові інструменти для роботи з базами даних, веб-фреймворками та системами віртуалізації. Хоча у проєктах, пов'язаних з ODOO, зазвичай достатньо Community-версії, використання Professional-редакції може суттєво спростити роботу з PostgreSQL та серверною архітектурою.

Таким чином, PyCharm виступає комплексним інструментом, який забезпечує повноцінне середовище для розробки, тестування та налагодження модулів ODOO. Його функціональні можливості сприяють підвищенню продуктивності розробника, структурованості коду та стабільності кінцевого програмного продукту, що робить його одним із найбільш придатних середовищ для реалізації рішень у сфері корпоративних ERP-систем.

3.1.3 Система управління базами даних PostgreSQL

Система управління базами даних PostgreSQL є однією з найпотужніших і найнадійніших реляційних СУБД з відкритим кодом, що широко застосовується в корпоративних інформаційних системах, критично важливих сервісах та масштабованих веб-додатках. PostgreSQL характеризується високою відповідністю стандартам SQL, розвинутими процесами транзакційності та широкими можливостями розширення, що вирізняє її серед інших реляційних систем. Завдяки

цим властивостям вона використовується як основна база даних у платформі ODOO, забезпечуючи зберігання, цілісність і консистентність корпоративних даних [23], [24].

Вбудована підтримка JSON та JSONB, можливість створення власних типів даних, процедур і функцій, а також тригерів і розширень дають змогу реалізовувати складну бізнес-логіку безпосередньо на рівні бази даних. У рамках розробки модулів для ODOO ці можливості особливо важливі, оскільки ORM платформи активно взаємодіє з PostgreSQL, покладаючись на її транзакційну модель, блокування, обмеження та індекси.

Крім класичних функцій реляційної СУБД, PostgreSQL має розвинуті засоби для високопродуктивного пошуку. До них належать інвертовані індекси, методи повнотекстового пошуку, вбудовані аналізатори мов, нормалізація тексту, а також індекси GIN і GiST, що забезпечують високу швидкість обробки запитів на великих обсягах текстових даних. Ці можливості стали основою для реалізації повнотекстового пошуку в межах даного проєкту. Використання розширення PostgreSQL Full Text Search дозволяє виконувати пошук фраз, лексем, логічних конструкцій та складних запитів без залучення зовнішніх пошукових рушіїв, що суттєво спрощує архітектуру системи та зменшує потребу в додаткових апаратних ресурсах.

PostgreSQL також відзначається стійкістю та високим рівнем безпеки. Підтримка транзакцій за моделлю ACID гарантує коректність змін у даних навіть у випадку високого навантаження або непередбачених збоїв. Процеси реплікації й резервного копіювання забезпечують безперервність роботи та можливість масштабування у великих інсталяціях. Для системи ODOO, яка може обробляти сотні тисяч або мільйони записів, ці властивості відіграють ключову роль.

Гнучкість та можливість розширення PostgreSQL є ще одним важливим аспектом. Завдяки відкритості структури бази даних, широкій спільноті та наявності численних розширень (таких як pg_trgm, unaccent, hstore, PostGIS), система може адаптуватися до нестандартних сценаріїв використання. У контексті цього проєкту модуль pg_trgm забезпечує ефективне порівняння текстових фрагментів і пришвидшує пошук за частковими збігами, що суттєво підвищує якість пошукових результатів [25].

Як основна СУБД платформи ODOO, PostgreSQL визначає значну частину її архітектурних рішень. Інструменти ORM ODOO спираються саме на особливості PostgreSQL, включаючи складні типи індексів, транзакції, обмеження цілісності та оптимізований SQL-процесор. Це робить PostgreSQL не просто інструментом зберігання даних, а ключовим елементом усієї системи, безпосередньо пов'язаним із продуктивністю, масштабованістю та надійністю розроблюваного модуля.

Тож, PostgreSQL є базовою технологічною платформою, що забезпечує надійну основу для реалізації повнотекстового пошуку в ODOO. Її поєднання продуктивності, гнучкості та розширюваності дає змогу створювати масштабовані та ефективні інструменти, здатні працювати з великими масивами корпоративних даних без залучення зовнішніх сервісів.

3.1.4 Використання API ODOO

Для реалізації модуля повнотекстового пошуку було обрано API ODOO як основний засіб програмної взаємодії з системою. API ODOO є потужним і універсальним програмним інтерфейсом, який надає повноцінний доступ до всіх моделей, полів, бізнес-логіки та методів як ядра платформи, так і встановлених модулів, не вимагаючи при цьому прямого втручання в код сервера чи бази даних. Він працює через віддалені виклики процедур за протоколами XML-RPC та JSON-RPC, завдяки чому будь-яка операція — читання, запис, пошук, запуск робочих процесів чи серверних дій — може виконуватися практично з будь-якої мови програмування або зовнішньої системи через звичайні HTTP-запити.

Саме використання API, а не прямого доступу до бази даних чи глибокої модифікації ядра, забезпечує надійну, безпечну та гнучку інтеграцію модуля повнотекстового пошуку з рештою екосистеми ODOO. Модуль легко отримує дані з моделей Продажів, Складу, CRM чи Документів через стандартні методи `search_read` і `read`, застосовує складні доменні фільтри та повертає результати іншим модулям, повністю зберігаючи єдине джерело правди та поважаючи всі налаштування прав доступу. Нові функції індексації, переіндексації чи розширеного пошуку можна реалізувати як серверні методи або контролери, а потім викликати їх однаково зсередини ODOO та ззовні, не змінюючи при цьому вже існуючий код.

API значно спрощує автоматизацію: переіндексацію великих обсягів даних чи запуск пошуку за розкладом можна організувати через планові задачі або вебхуки, що практично усуває необхідність ручного втручання. Водночас JSON-RPC інтерфейс відкриває можливість швидко підключати мобільні додатки, сторонні сервіси, аналітичні панелі чи власний сучасний фронтенд, залишаючи всю бізнес-логіку централізованою на стороні ODOO.

Важливо, що API працює через той самий ORM-шар, який використовує внутрішній код платформи, тому автоматично успадковує всі оптимізації кешування, попереднього завантаження пов'язаних записів і надання безпеки. Це гарантує високу продуктивність навіть при значному зростанні кількості користувачів і обсягу даних, а також робить систему передбачуваною та стійкою до помилок.

Таким чином, вибір API ODOO як основного інструменту розробки забезпечує повну інтеграцію модуля повнотекстового пошуку в екосистему платформи, зберігає його безпеку, масштабованість і простоту подальшого розширення, а також відкриває широкі можливості для використання функціональності модуля як всередині стандартного інтерфейсу ODOO, так і в зовнішніх додатках та сервісах.

3.1.5 Мова програмування JavaScript та фреймворк Owl

Для реалізації клієнтської частини модулів використовується сучасний JavaScript-фреймворк Owl (ODOO Web Library), який із версії 15 є основним інструментом побудови всього веб-інтерфейсу ODOO.

Фреймворк Owl – це легка, швидка та повністю реактивна бібліотека, розроблена спеціально для ODOO. Вона працює за принципом віртуального DOM і автоматично оновлює лише ті ділянки сторінки, де дійсно відбулися зміни. Завдяки цьому вдається досягти миттєвої реакції інтерфейсу: автодоповнення під час введення запиту, підсвічування знайденого тексту безпосередньо в документах, динамічне підвантаження результатів і плавне розгортання контексту збігів — усе це відбуватиметься без затримок і без перезавантаження сторінки, навіть коли пошук виконується по великих обсягах даних[26] – [27] .

Головні переваги вибору саме Owl полягають у тому, що він глибоко інтегрований в екосистему ODOO: компоненти автоматично отримують доступ до

поточної теми оформлення, локалізації, прав користувача та корпоративного брендування. Це означає, що пошукова панель, віджет підсвічування та спливаючі підказки виглядатимуть і працюватимуть так само природно, як будь-який стандартний елемент системи — у формах, списках, канбан-дошках чи дашбордах. Крім того, використання QWeb-шаблонів і чітке розділення логіки на невеликі перевикористовувані компоненти значно спрощує подальшу підтримку та розвиток модуля.

3.2 Основи повнотекстового пошуку в PostgreSQL

У цьому підрозділі розглянуто фундаментальні принципи, на яких ґрунтується повнотекстовий пошук у PostgreSQL. Спершу описується векторна модель подання документів, що визначає спосіб перетворення тексту у структуровану форму, придатну для індексації та логічного порівняння. Далі увага приділяється інструментам морфологічної нормалізації та підтримці полімовного пошуку, що забезпечують коректне опрацювання текстів різними мовами та підвищують точність збігів. Окремий акцент зроблено на логіці побудови запитів `tsquery` і використанні позиційних операторів, які дозволяють формувати складні семантичні конструкції та уточнювати структуру пошукового запиту.

Подальші підрозділи розкривають принципи ранжування документів за допомогою функцій `ts_rank` і `ts_rank_cd`, які визначають порядок відображення результатів відповідно до їхньої релевантності. Завершується розділ описом процесу формування сніпетів та контекстного підсвічування за допомогою `ts_headline`, важливої частини сучасних пошукових систем, що значно покращує зручність перегляду результатів пошуку.

3.2.1 Векторна модель представлення тексту в PostgreSQL

Методи повнотекстового пошуку в PostgreSQL ґрунтуються на векторній моделі подання документів (Vector Space Model), яка розглядає текст як набір нормалізованих лексем зі структурованими атрибутами. Такий підхід дає змогу

перетворювати неструктурований текст у формальний об'єкт, зручний для індексації, логічного порівняння та обчислення релевантності. У PostgreSQL ця модель реалізована за допомогою двох взаємодоповнювальних типів даних: `tsvector` та `tsquery`, а також інвертованих індексів, оптимізованих для пошуку за змістом.

Тип представляє документ як впорядкований набір лексем, кореневих або стемованих форм слів, які є результатом мовної нормалізації. Кожна лексема супроводжується позиційною інформацією та вагами, що дають можливість враховувати як частотність появи, так і важливість окремих сегментів тексту.

Така модель забезпечує можливість виконувати операції логічного зіставлення, позиційного пошуку, ранжування результатів та визначення релевантності на основі частотних характеристик.

Тип `tsquery`, у свою чергу, описує структуру пошукового запиту. Він має вигляд дерева логічних операторів, яке може включати кон'юнкцію, диз'юнкцію, заперечення, позиційні обмеження та фразові конструкції. Це забезпечує підтримку не лише простих пошукових запитів, але й складних логічних виразів, що дозволяє точно інтерпретувати намір користувача.

Сумісність цих типів забезпечується оператором відповідності `@@`, який визначає, чи задовольняє документ умовам запиту, після чого, за наявності інвертованого індексу GIN або GiST, пошук виконується з високою швидкістю навіть у базах даних зі значним обсягом записів [28].

Використання векторної моделі дозволяє PostgreSQL виконувати повнотекстовий пошук без зовнішніх рушіїв, підтримуючи ранжування результатів, пошук фраз, часткові та позиційні збіги. У межах розроблюваного модуля для ODOO ця модель забезпечує оптимальний баланс між продуктивністю, точністю пошуку та інтеграцією з існуючою архітектурою системи, де PostgreSQL виступає основним сховищем даних.

3.2.2 Морфологічна нормалізація та полімовний пошук

Морфологічна нормалізація є одним із центральних етапів побудови повнотекстового пошуку, оскільки вона визначає, у якій формі слова будуть представлені у векторному просторі та як саме PostgreSQL інтерпретує запити

користувачів. Основна її мета — зменшити розмірність ознакового простору та перетворити різні словоформи на узагальнену лексему, що дозволяє значно підвищити якість збігів і забезпечити більш точну релевантність результатів.

У PostgreSQL морфологічна нормалізація реалізована через систему конфігурацій повнотекстового пошуку (text search configurations), які визначають правила обробки тексту на певній мові. Кожна конфігурація включає кілька ключових компонентів:

- словникові модулі, що виконують стемінг або лематизацію (наприклад, стемер Snowball або модулі ISpell/Thesaurus)
- набори стоп-слів, які виключають з індексації надмірно поширені слова, незначущі для змістового пошуку;
- правила синтаксичного аналізу, що відповідають за токенізацію та класифікацію фрагментів тексту.

Для найбільш поширених мов, які часто використовуються в багатомовних ERP-системах, PostgreSQL надає готові конфігурації: `pg_catalog.ukrainian`, `pg_catalog.russian`, `pg_catalog.english`, кожна з яких містить спеціалізований набір лінгвістичних правил, оптимізованих під особливості певної мови.

3.2.3 Розширена логіка запитів `tsquery` і позиційні оператори

Запити `tsquery` у PostgreSQL істотно виходять за межі класичного булевого пошуку, пропонуючи розширений набір операторів, що підвищують точність, гнучкість і семантичну виразність пошукових конструкцій. На відміну від простого згортання лексем у вигляді перетинів або диз'юнкцій, `tsquery` дозволяє описувати структурні та позиційні залежності між словами, реалізуючи повноцінний проксиміті-пошук (proximity search) та фразове зіставлення.

Одним із ключових інструментів є позиційні оператори, які визначають відстань між лексемами у вихідному тексті:

- оператор безпосередньої послідовності (`<->`), що вимагає появи лексем строго одна за одною;

- оператор відносної близькості ($\langle n \rangle$), що обмежує максимальну кількість позицій між лексемами;
- префіксний оператор ($\langle * \rangle$), який дозволяє зіставляти початок лексеми з довільним продовженням.

У сукупності ці оператори надають пошуку властивості, характерні для сучасних пошукових систем, дозволяючи визначати не лише наявність слів, а й їхню структурну організацію в документі. На практиці це суттєво підвищує релевантність результатів, особливо у випадках, коли текст містить складні описи, технічну документацію чи багатокomпонентні назви.

3.2.4 Алгоритми ранжування релевантності: `ts_rank` і `ts_rank_cd`

Алгоритми ранжування в PostgreSQL визначають порядок, у якому результати повнотекстового пошуку подаються користувачеві, і тим самим істотно впливають на зручність та ефективність роботи з інформацією. У системі реалізовано два основні підходи: `ts_rank` та `ts_rank_cd`, кожен з яких оцінює релевантність документа за власними критеріями.

Функція `ts_rank` працює за принципом класичної частотної моделі, де вищу оцінку отримують ті документи, у яких слова пошукового запиту трапляються частіше та мають більшу вагу відповідно до їхнього значення в тексті. Крім частоти, у процесі ранжування враховуються й структурні особливості документа: наприклад, лексеми, що трапляються у важливих полях або розділах тексту, можуть мати більшу вагомість. Інша функція, `ts_rank_cd`, доповнює частотний підхід аналізом так званої «щільності покриття». Ідея полягає в тому, що документ вважається більш релевантним, якщо слова запиту розташовані не лише часто, але й близько одне до одного, утворюючи компактні тематичні фрагменти. Таким чином, система надає перевагу документам, які мають чіткіші та логічно пов'язані появи всіх ключових термінів. Такий інструмент є корисним при роботі з довгими текстами або багатослівними запитами, де важливо не лише знайти слова, але й зрозуміти їхній контекст [29].

У межах реалізованого модуля використано алгоритм `ts_rank`. Це пояснюється тим, що він забезпечує оптимальне співвідношення точності та продуктивності, не

створює надлишкового навантаження на систему та відповідає стандартному підходу до представлення результатів пошуку: найрелевантніші документи відображаються першими.

3.2.5 Контекстне підсвічування

У системах повнотекстового пошуку важливо не лише визначити, чи містить документ потрібні користувачеві дані, а й швидко продемонструвати, у якому саме контексті ці дані були знайдені. Цю функцію виконує `ts_headline`, який автоматично формує короткі узгоджені уривки тексту, так звані сніпети. Завдяки цьому користувач одразу бачить, наскільки знайдений документ відповідає його запиту, без потреби відкривати чи переглядати його повністю.

Алгоритм роботи `ts_headline` складається з кількох послідовних кроків. Спочатку система аналізує весь текст і визначає позиції всіх лексем, що збігаються із запитом. Далі обираються найінформативніші місця, ті фрагменти, де ключові слова утворюють компактні групи або трапляються найближче одне до одного. Цей алгоритм ґрунтується на жадібному пошуку оптимального інтервалу, тобто такого уривку, який охоплює максимальну кількість збігів при мінімально можливій довжині. Параметри функції дозволяють регулювати кількість таких фрагментів і їхню приблизну довжину.

Після вибору фрагмента система застосовує підсвічування. Кожен збіг обгортається у спеціальні маркери, завдяки яким користувацький інтерфейс може відобразити ключові слова з візуальним акцентом – наприклад, виділити їх жирним, кольором або іншим стилем. Оскільки `ts_headline` генерує повністю готовий HTML-фрагмент, додаткове оброблення на клієнтському боці не потрібне, що значно зменшує затрати часу на рендеринг та підвищує швидкодію.

У проєкті цей інструмент використано для забезпечення інформативних результатів пошуку, де користувач отримує не лише список записів, але й невеликі, змістовні витяги з релевантних документів. Такий підхід істотно підвищує зручність навігації великими текстовими масивами та відповідає загальним тенденціям розвитку сучасних пошукових систем.

3.3 Архітектура та реалізація модуля повнотекстового пошуку

У цьому підрозділі наведено практичну реалізацію розробленого модуля повнотекстового пошуку в системі ODOO. Розглянуто загальну архітектуру програмного забезпечення, процес первинного збору та індексування даних, підтримку актуальності JSON-архіву, обробку користувацьких запитів і формування результатів з урахуванням прав доступу та мультикомпанійності. Окремо описано структуру бази даних модуля та проаналізовано функціональні можливості системи на основі діаграми прецедентів, що дозволяє цілісно оцінити роботу створеного рішення.

3.3.1 Загальна архітектура

Архітектура модуля складається з двох взаємопов'язаних підсистем, які спільно працюють із централізованим JSON-архівом, що умовно зображені на рисунку 31.



Рисунок 3.1 – Загальна архітектура модуля

Перша підсистема, управління даними, відповідає за отримання актуальної інформації з моделей ODOO. До її функцій належать початковий збір даних, формування черги змін, фіксація подій створення, редагування чи видалення записів та подальша індексація. Усі зібрані дані структуруються у форматі JSON та

потрапляють до спільного архіву, який використовується як індексоване сховище для пошуку.

Друга підсистема, здійснення пошуку, обробляє запити користувача та виконує пошук у підготовленому архіві. Вона включає алгоритм аналізу запиту, збереження історії, отримання та фільтрації релевантних даних відповідно до вибраних критеріїв.

3.3.2 Підготовка даних та первинне індексування

Після встановлення модуля система ініціює процес первинного збору та структурування даних, який слугує фундаментом для подальшого повнотекстового пошуку. Основним завданням цього етапу є формування повного архіву текстової інформації з усіх нетехнічних моделей ODOO та перетворення отриманих даних у стандартизований формат JSON. Використання саме цього формату дає змогу компактно представити складні об'єкти з їхньою внутрішньою структурою, зберегти ієрархію полів і забезпечити зручний подальший аналіз даних на рівні бази PostgreSQL.

Після активації модуля запускається спеціальна процедура, яка послідовно проходить через усі моделі системи, що містять релевантну бізнес-інформацію. Для кожного запису відбираються текстові поля, а також інші значущі атрибути з явним зберіганням у базі. На цьому етапі виключаються технічні моделі, службові об'єкти, внутрішні таблиці ODOO та елементи, робота з якими може призвести до рекурсії або дублювання.

Сформовані записи не потрапляють до пошукового архіву безпосередньо. Натомість модуль використовує спеціальну механіку пакетного наповнення через чергу. Кожна порція зібраних даних складається у таблицю `queue`, що дозволяє розподілити навантаження та уникнути масових одночасних транзакцій. Такий підхід робить можливим початкове індексування навіть на системах із великими масивами даних без блокування основних бізнес-процесів.

Далі за обробку черги відповідає заплановане завдання, яке через регулярні проміжки часу, забирає порцію елементів і переносить їх до центральної таблиці `json_archive`. Для додавання та оновлення застосовується конструкція `INSERT ... ON CONFLICT DO UPDATE`, що гарантує консистентність і відсутність дублювання.

Важливо, що процес первинного індексування адаптується до можливостей конкретної бази. Якщо PostgreSQL підтримує розширення `pg_trgm`, модуль автоматично використовує триграмні індекси, які значно пришвидшують пошук за частковими збігами, автодоповненням і нечіткими відповідностями. Якщо ж розширення недоступне, спочатку намагатиметься встановити його, і якщо все ще не вийде система переходить на стандартні інвертовані індекси, зберігаючи повну сумісність і коректність роботи.

Для кращого розуміння процесу на Рисунку 3.2 зображено ключові етапи встановлення модуля.

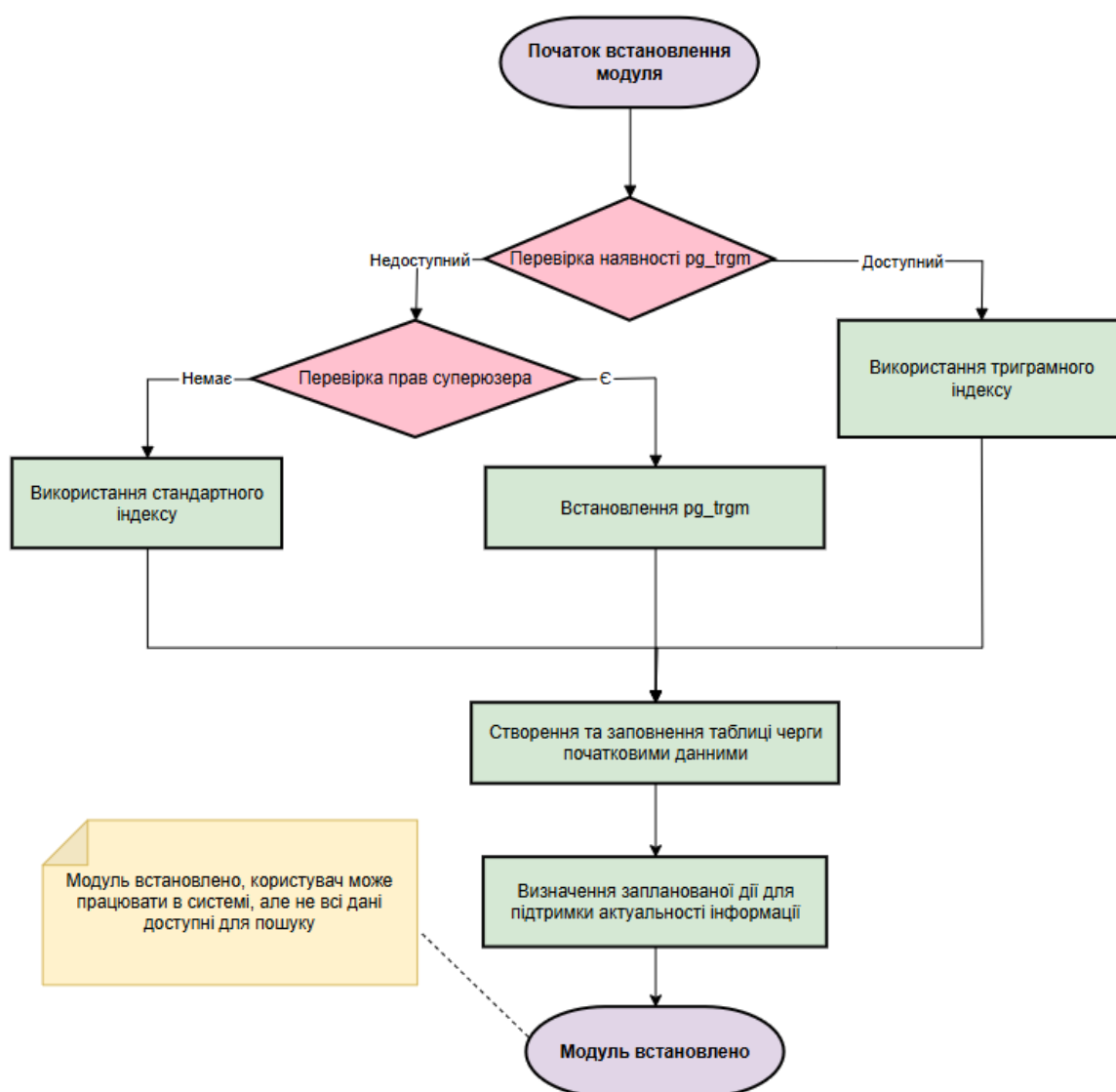


Рисунок 3.2 – Процес первинного збору інформації під час встановлення модуля

Процес встановлення вважається завершеним тоді, коли буде оброблено всі записи в базі даних і необхідні з них потраплять до черги. На даному етапі модуль вже можна буде використовувати, щоправда не всі дані будуть доступні до пошуку. В залежності від розміру системи, буде необхідний час для обробки записів, але цей процес не блокує роботу системи. Детальніше про процес обробки черги і підтримку буде описано в наступному розділі.

3.3.3 Алгоритм підтримки актуальності

Після завершення первинного заповнення важливою частиною роботи модуля стає постійна синхронізація пошукового архіву зі станом даних у системі ODOO. Цю задачу реалізовано через комплексний метод перехоплення змін, формування завдань у черзі та їх асинхронної обробки.

Детальний процес підтримки актуальності в таблиці JSON-архіві показано на рисунку 3.3.

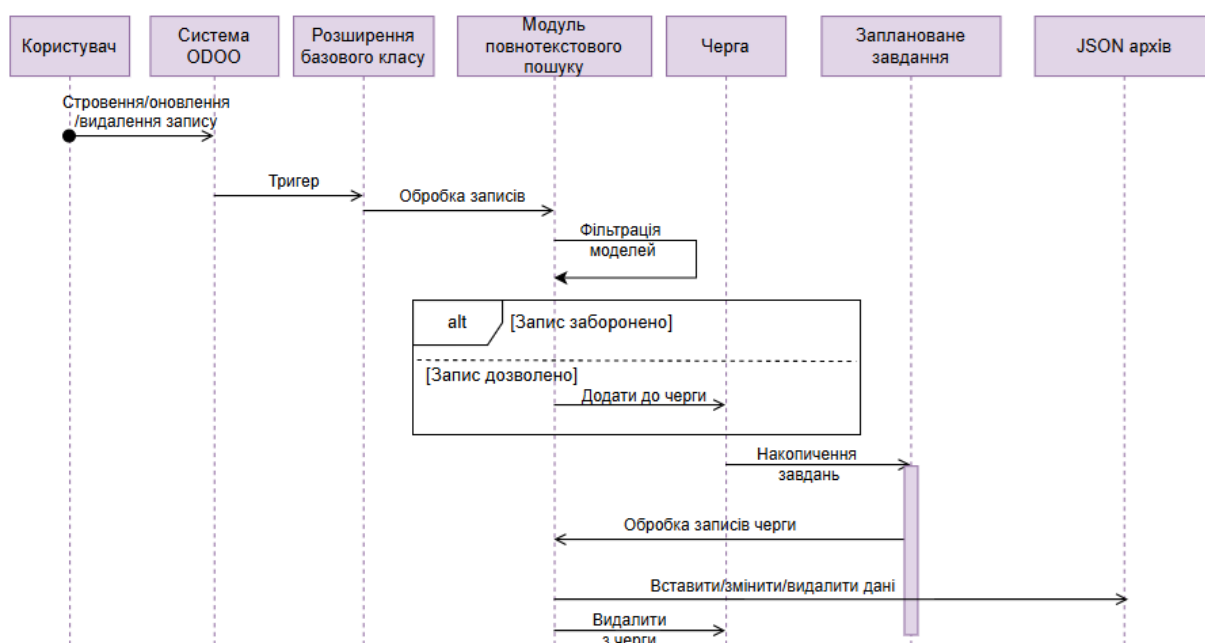


Рисунок 3.3 – Діаграма послідовності процесу підтримки актуальності

Щойно користувач створює, змінює чи видаляє будь-який запис, ODOO викликає відповідний метод моделі. Модуль розширює поведінку цих методів базового класу base: подія фіксується лише після того, як основна транзакція успішно

завершена. Це означає, що у випадку виникнення винятку, невдалої валідації чи порушення правил доступу зміни не потраплять до черги. Таким чином забезпечується жорстка відповідність між фактичним станом системи та тим, що зберігається у пошуковому індексі.

На етапі перехоплення виконується додаткова фільтрація: технічні або службові моделі виключаються, а для решти формується компактне повідомлення, яке потрапляє до тієї ж таблиці `queue`, що використовувалася під час первинного індексування. Таким чином модуль використовує єдиний процес обробки як для початкового заповнення, так і для синхронізації даних у реальному часі.

Заплановане завдання, що працює з інтервалом у кілька хвилин, обробляє накопичені зміни пакетно, що забезпечує стабільне навантаження та відсутність надмірної кількості транзакцій [30], [31]. У ході обробки формується оновлене JSON-представлення запису або, у разі видалення, додатково перевіряється, чи був запис дійсно видалений з бази. Така перевірка запобігає ситуаціям, коли тимчасові помилки або конфлікти транзакцій могли призвести до видалення інформації з індексу без фактичного видалення з моделі.

Після успішної обробки зміни потрапляють до таблиці `json_archive`, а відповідні елементи видаляються з черги. Завдяки такій логіці затримка між зміною даних у системі та їх появою в пошуковому індексі не перевищує кількох хвилин, що є оптимальним компромісом між актуальністю та продуктивністю.

Усі ці процеси працюють у повністю асинхронному режимі та не впливають на взаємодію користувачів із системою. Завдяки такій архітектурі модуль забезпечує не лише ефективний початковий збір інформації, а й гарантовану актуальність індексу при будь-яких змінах у даних. Це створює стабільну основу для якісного повнотекстового пошуку, що постійно відображає реальний стан бізнес-інформації в ODOO.

3.3.4 Обробка запиту і здійснення пошуку

Алгоритм перетворення текстового запиту на формат, зрозумілий PostgreSQL, зображено на рисунку 3.4.

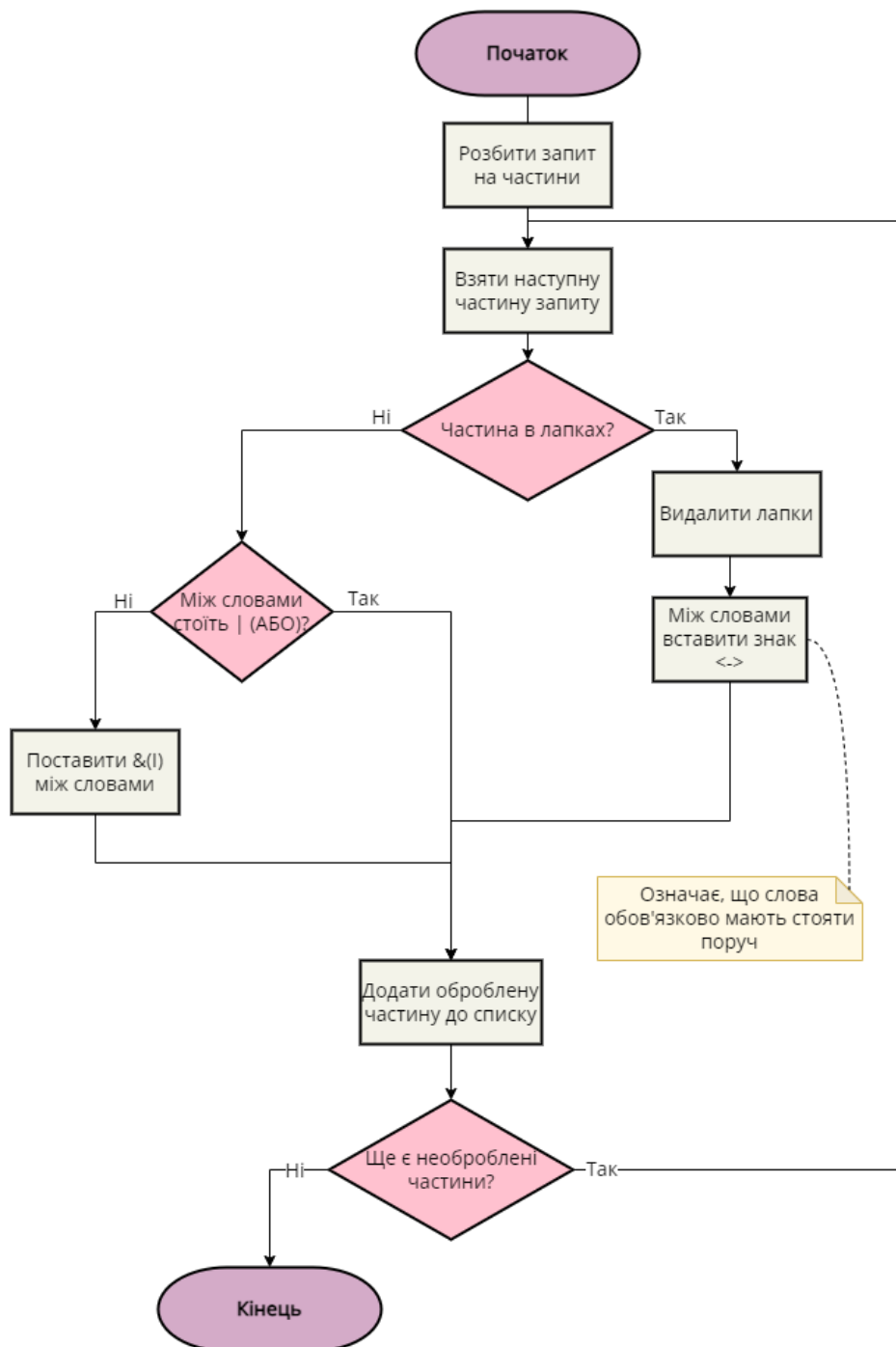


Рисунок 3.4 – Процес обробки запиту користувача

Процес здійснення пошуку у модулі починається з обробки текстового рядка, який вводить користувач. На першому етапі запит розбивається на окремі частини та аналізується на наявність спеціальних символів, що дозволяють формувати гнучкі пошукові конструкції, а саме:

— лапки («») інтерпретуються як позначення точної фрази, між словами якої в запиті має бути збережено порядок;

- знак логічного АБО (|) визначає альтернативні варіанти слів, що можуть з'являтися в записі;
- знак оклику (!) використовується для виключення певних термів;
- дужки (()) дозволяють групувати фрагменти виразу;
- двокрапка зірочка (:*) застосовується для пошуку за префіксом.

Якщо ж слова подані без спеціальних операторів і розділені пробілами, між ними автоматично додається логічний оператор «І», що забезпечує пошук записів, у яких присутні всі зазначені слова, незалежно від їхнього порядку в тексті.

Після завершення синтаксичного аналізу кожна частина запиту трансформується у відповідний елемент `tsquery`-виразу, який може містити логічні оператори, позиційні обмеження та конструкції для пошуку фраз. У такому вигляді отриманий запит передається у SQL-виклик, де здійснюється пошук по індексованому JSON-архіву, сформованому на основі текстових полів вибраних моделей.

Отримані результати на цьому етапі ще не можуть бути безпосередньо показані користувачеві, оскільки модуль працює переважно на рівні бази даних і не враховує права доступу під час виконання SQL-запиту. Тому кожен знайдений запис додатково проходить перевірку на можливість перегляду відповідним користувачем. Якщо користувач не має дозволу на доступ до певного об'єкта, такий запис автоматично відкидається і не потрапляє до кінцевої видачі. Це гарантує відповідність результатів пошуку політикам безпеки ODOO та унеможливорює розкриття інформації між користувачами.

Додатково враховується мультикомпанійність системи. У випадках, коли користувач належить до кількох компаній, результати пошуку фільтруються відповідно до активної компанії, встановленої в його сеансі. Це означає, що у видачі відображаються лише ті записи, які відповідають контексту поточної компанії та доступні в її межах. Такий підхід дозволяє забезпечити коректність даних у багатокомпанійних середовищах і підтримує характерну для ODOO модель ізольованості даних.

Після завершення перевірок сформований список результатів зберігається у спеціальну модель, де фіксується текст запиту, знайдені записи, фрагменти тексту зі збігами та посилання на відповідні форми у системі. Це дає можливість повторно

Таблиця `json_archive` є центральним сховищем індексованих текстових даних. До неї потрапляє структурований JSON-вміст записів різних моделей ODOO, який використовується під час виконання повнотекстового пошуку. Таблиця містить службові поля, ідентифікатор моделі, текстовий архів та метадані про створення та зміну запису. Через зовнішні ключі вона пов'язана з таблицями `full_text_search_result` та `full_text_search_queue`, які використовують її дані під час пошуку й оновлення індексу.

Основна таблиця `full_text_search` зберігає інформацію про пошукові запити користувачів, а також пов'язані параметри інтерфейсу, наприклад, увімкнення фільтрації або показ підказок. Запис у цій таблиці створюється кожного разу, коли користувач ініціює новий пошук. Через поле `search_id` вона пов'язана з таблицею результатів `full_text_search_result` і зі службовою таблицею зв'язків `full_text_search_table_rel`.

Таблиця `full_text_search_result` містить результати пошукового запиту, включно з ідентифікатором знайденого запису, його моделлю, фрагментами тексту зі збігами та URL-посиланням на форму перегляду. Результати пошуку прив'язуються до одного запиту та можуть бути додатково фільтровані або використовуватись повторно без повторного запуску SQL-пошуку.

Таблиця `full_text_search_table` зберігає перелік моделей ODOO, у яких були знайдені збіги під час пошуку. Це дозволяє користувачеві фільтрувати результати за конкретними таблицями. Вона реалізує зв'язок «багато до багатьох» із запитамі через таблицю пов'язань `full_text_search_table_rel`, яка містить пари значень `search_id` і `table_id`.

Таблиця `full_text_search_queue` використовується для накопичення змін у моделях ODOO. У неї додаються записи після створення, редагування або видалення об'єктів. Заплановане завдання періодично обробляє дані з цієї таблиці, оновлюючи `json_archive` та підтримуючи його актуальний стан. Завдяки цьому індекс не блокує бізнес-транзакції і рівномірно оновлюється у фоновому режимі.

Таблиця `res_company`, представлена на діаграмі, використовується для забезпечення мультикомпанійності пошуку. Модуль враховує активну компанію

користувача, тому зв'язок між компаніями та результатами дозволяє обмежувати видимість записів відповідно до контексту роботи [32].

3.3.6 Функціональні можливості

Для відображення взаємодії користувача з модулем та демонстрації його основних функціональних можливостей було побудовано діаграму прецедентів, зображену на рисунку 3.6. Вона узагальнює ключові сценарії роботи системи та допомагає структурувати процес виконання пошуку й пов'язаних із ним операцій.

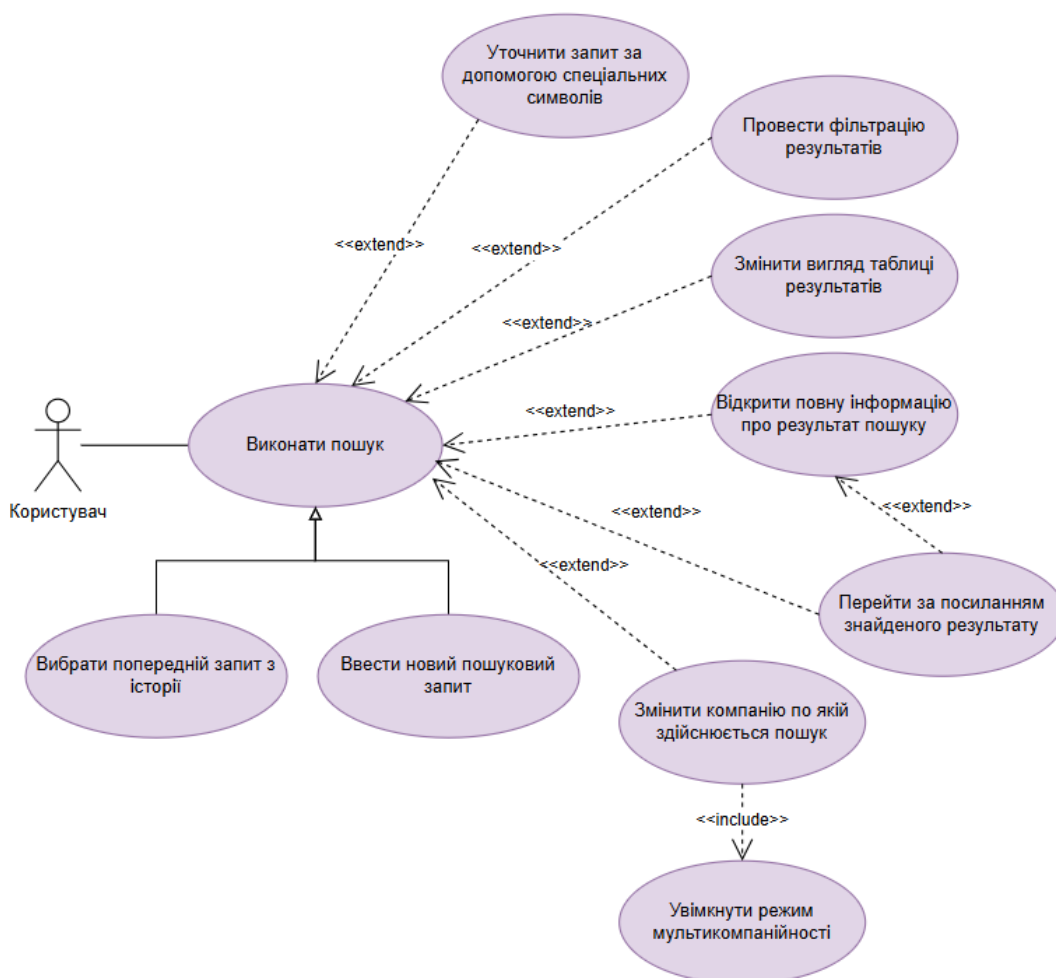


Рисунок 3.6 – Діаграма прецедентів модуля повнотекстового пошуку

На діаграмі прецедентів головним і єдиним актором є користувач, який ініціює операцію пошуку. Базовим варіантом використання є «Виконати пошук», що охоплює процес формування запиту та отримання результатів. Користувач

може почати пошук двома способами, а саме ввести новий пошуковий запит чи вибрати попередній запит з історії для повторного використання. До основного прецеденту додаються додаткові сценарії, які розширюють можливості пошуку:

- уточнити запит за допомогою спеціальних символів для формування гнучких запитів;
- провести фільтрацію результатів за моделями в яких вони були знайдені;
- змінити вигляд таблиці результатів через налаштування відображень колонок;
- відкрити повну інформацію про результат пошуку для детального перегляду знайденого запису;
- перейти за посиланням на знайдений запис в системі;
- змінити компанію, по якій здійснюється пошук.

Останній прецедент включає в себе необхідність активованого режиму мультикомпанійності, щоб дозволити користувачу мати доступ до записів з двох та більше різних компаній.

Висновки до розділу 3

У третьому розділі наведено повний огляд процесу розроблення програмного забезпечення для повнотекстового пошуку в ODOO – від вибору інструментів до практичного застосування створеного модуля. Визначено й обґрунтовано використання Python, PostgreSQL та засобів інтерфейсу ODOO, що забезпечило узгодженість рішення з архітектурою ERP-системи та можливість роботи з великими обсягами даних.

Описано структуру модуля та взаємодію його складових: підготовку даних і формування JSON-архіву, організацію черги для первинного та подальшого оновлення інформації, а також логіку обробки запитів, що підтримує складні пошукові конструкції, враховує права доступу користувачів і контекст активної компанії. Розглянуто структуру бази даних, яка забезпечує впорядковане збереження запитів, результатів і службової інформації, необхідної для стабільної роботи модуля.

4 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Інтерфейс користувача модуля повнотекстового пошуку, інтегрованого в систему ODOO, розроблений таким чином, щоб забезпечити максимально просту та інтуїтивно зрозумілу взаємодію. Усі елементи інтерфейсу організовані логічно та доступно, що дозволяє користувачам швидко освоїти роботу з модулем без необхідності додаткового навчання.

4.1 Встановлення модуля

Для встановлення модуля повнотекстового пошуку користувачу необхідно перейти у меню додатків, як показано на рисунку 4.1.

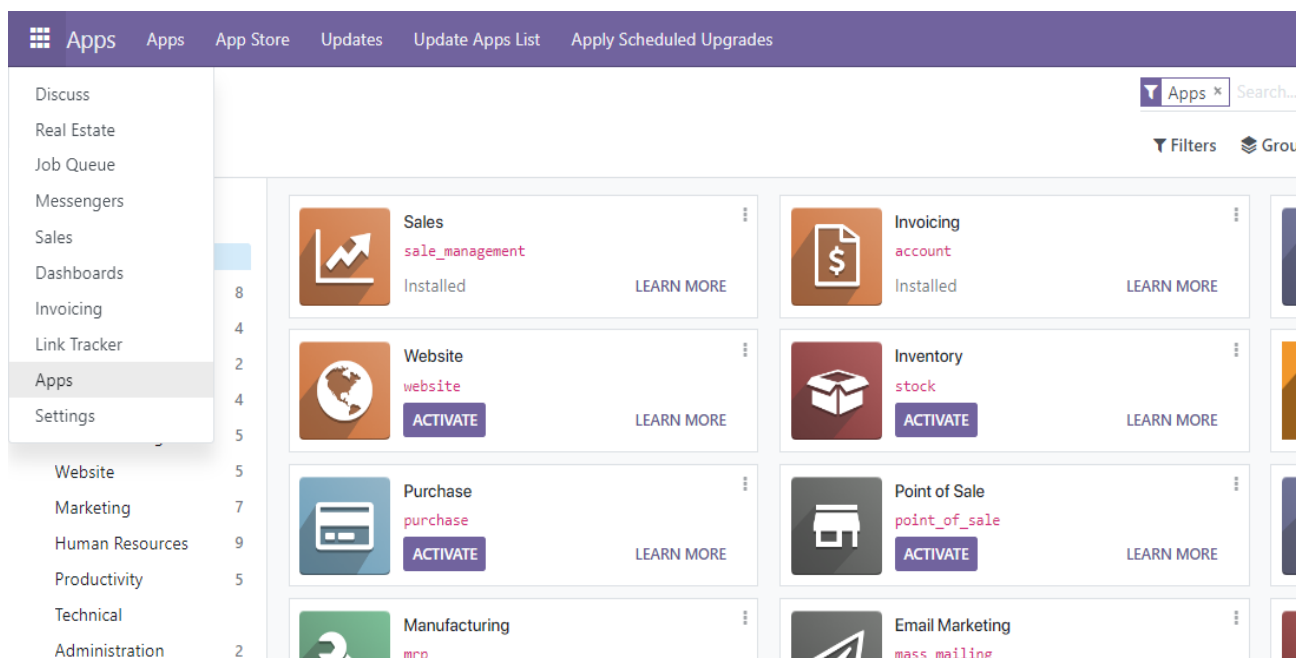


Рисунок 4.1 – Меню додатків

У полі пошуку слід ввести назву Full-text search і підтвердити пошук клавішею Enter, після цього на екрані відобразиться потрібний модуль, що продемонстровано на рисунку 4.2.

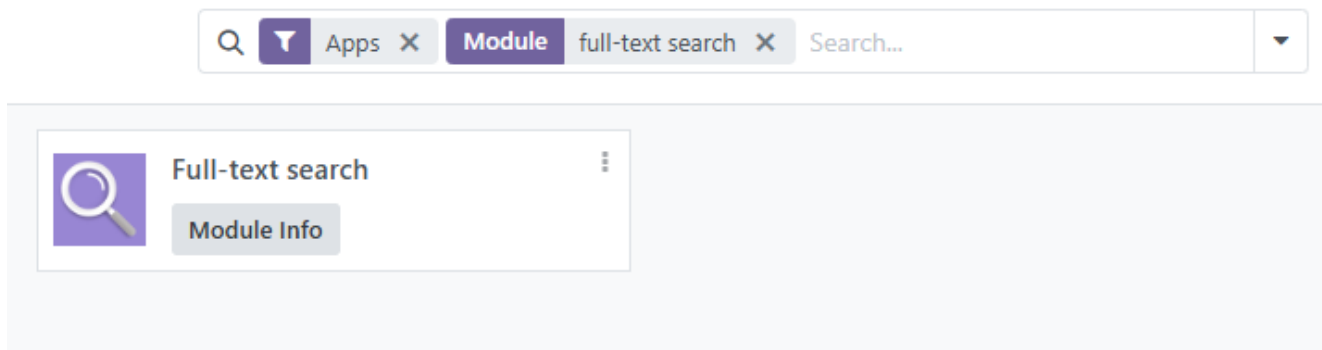


Рисунок 4.2 – Модуль повнотекстового пошуку в меню додатків

Після натискання кнопки Activate модуль буде встановлено у систему, і в основному меню з'явиться новий пункт Full-text Search, що продемонстровано на рисунку 4.3.

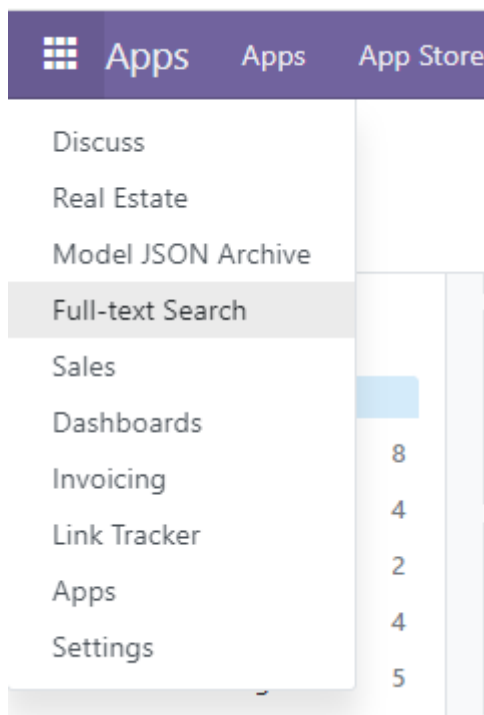


Рисунок 4.3 – Модуль повнотекстового пошуку в меню

Натиснувши на відповідний пункт користувач опиниться на формі пошуку і модуль повністю готовий до використання. Також після встановлення модуля, на головній панелі з'являється кнопка швидкого доступу до форми пошуку, що показано на рисунку 4.4.

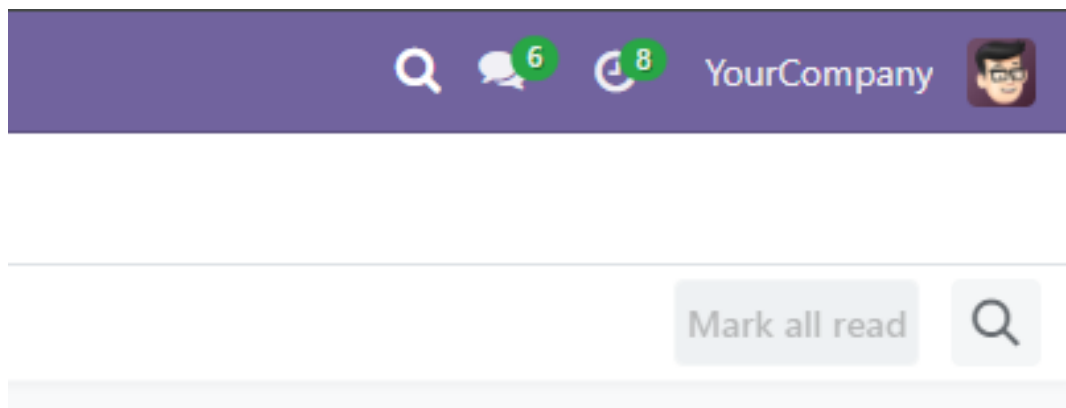


Рисунок 4.4 – Кнопка швидкого доступу

Таким чином, користувач швидко і з будь якого місця системи може перейти до модуля пошуку.

4.2 Здійснення пошуку

На головній сторінці модуля, зображений на рисунку 4.5, розміщено поле для введення пошукового запиту та кнопку запуску пошуку, за допомогою яких користувач може одразу розпочати пошук.

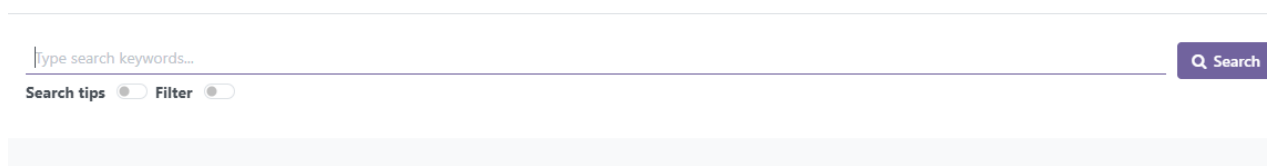


Рисунок 4.5 – Головна форма пошукового модулю

Користувач вводить ключові слова або фрази у поле пошуку. Ці запити будуть збережені в історію і надалі, при повторному пошуку фрази, її можна буде вибрати з спадного списку, як показано на рисунку 4.6.

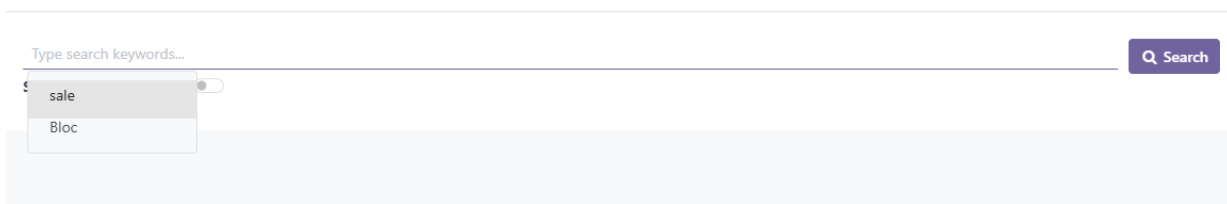


Рисунок 4.6 – Історія пошуку

Після введення запиту користувач може застосовувати різні логічні оператори, що забезпечують гнучкіший та точніший пошук. Зокрема, подвійні лапки дозволяють виконувати пошук за точною фразою, символ `|` використовується для альтернативних варіантів, а знак `!` — для виключення небажаних слів. Дужки дають можливість групувати умови, а спеціальні конструкції на кшталт `*` забезпечують префіксний пошук.

Для ознайомлення з повним переліком доступних операторів і прикладів їх використання користувач може увімкнути підказку Search tips, як показано на рисунку 4.7.

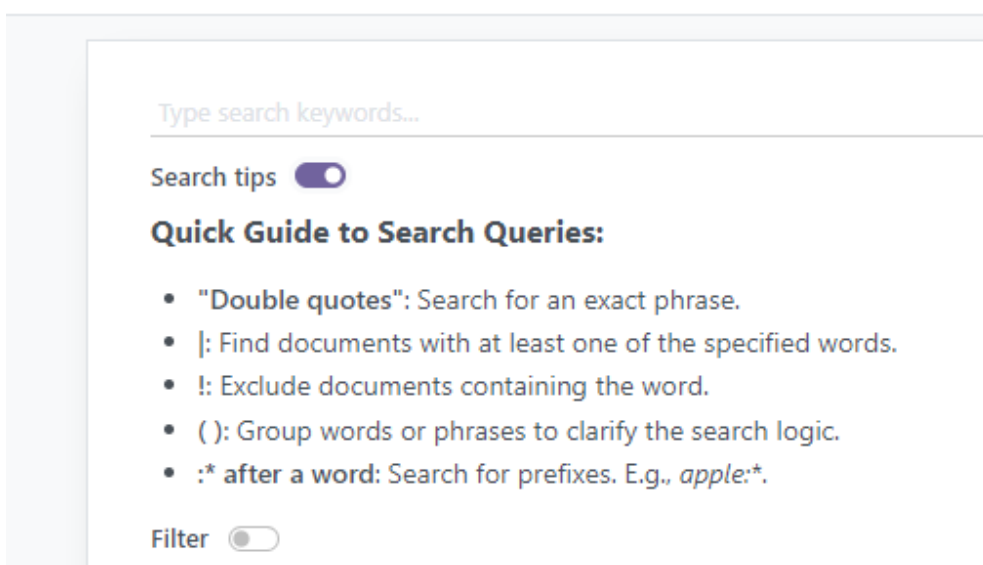
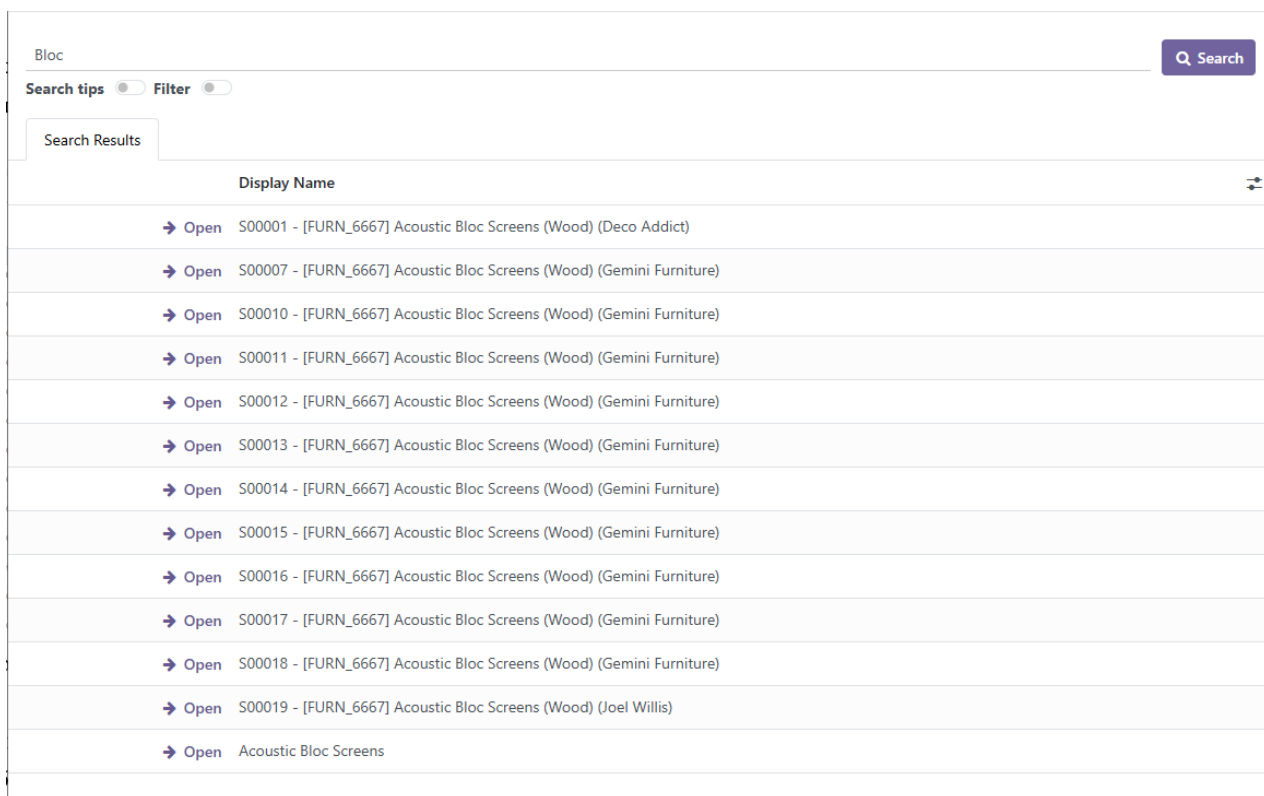


Рисунок 4.7 – Підказка з логічними символами

Після введення запиту користувач може запустити пошук, натиснувши кнопку «Пошук» або клавішу Enter. Система аналізує введений текст, виконує пошук у підготовленому індексі та повертає результати у вигляді впорядкованого списку.

Кожен елемент списку містить назву запису, де були знайдені збіги та посилання на повний запис, що зображено на рисунку 4.8.



The screenshot shows a search interface with a header 'Bloc' and a search button. Below the header, there are tabs for 'Search tips' and 'Filter'. The main section is titled 'Search Results' and displays a table of results. Each row includes a link to 'Open' the full record and the 'Display Name' of the record.

	Display Name
→ Open	S00001 - [FURN_6667] Acoustic Bloc Screens (Wood) (Deco Addict)
→ Open	S00007 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00010 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00011 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00012 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00013 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00014 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00015 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00016 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00017 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00018 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)
→ Open	S00019 - [FURN_6667] Acoustic Bloc Screens (Wood) (Joel Willis)
→ Open	Acoustic Bloc Screens

Рисунок 4.8 – Результати пошуку

Користувач може натиснути на відповідний запис, щоб відкрити детальну форму в окремій вкладці або вікні, що дає можливість оперативно переглянути повну інформацію щодо вибраного об’єкта.

Якщо необхідно відобразити додаткові дані, наприклад, назву моделі, у якій знайдено збіг, компанію, до якої належить запис, чи інші службові поля, користувач може скористатися спадним списком налаштувань відображення, розташованим у правому верхньому куті таблиці результатів (рисунок 4.9).

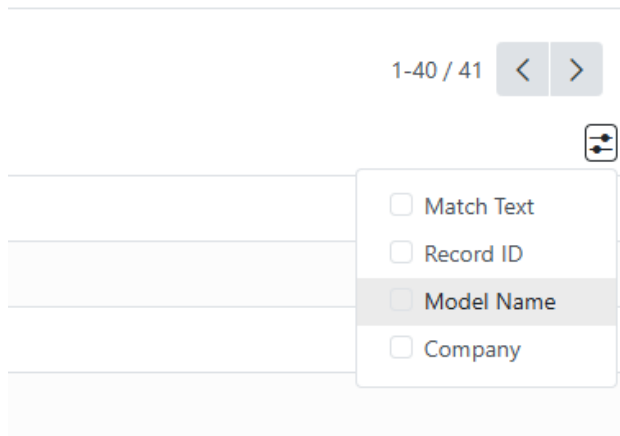


Рисунок 4.9 – Редагування вигляду результатів

На рисунку 4.10 представлена форма, на яку веде перше посилання з прикладу, продемонстрованого на одному з попередніх рисунків.

[FURN_6666] Acoustic Bloc Screens

PRINT LABELS

0 Extra Prices 16.00 Units Sold

Product Name
☆ Acoustic Bloc Screens

☒ Can be Sold ☒ Can be Purchased

General Information Sales Purchase Accounting

Product Type [?] Consumable Sales Price [?] \$ 295.00

Invoicing Policy [?] Delivered quantities Customer Taxes [?]

Consumables are physical products for which you don't manage the inventory level: they are always available.

Invoice after delivery, based on quantities delivered, not ordered.

Cost [?] \$ 287.00

Internal Reference FURN_6666

Barcode [?]

Product Category All / Saleable / Office Furniture

Product Tags

Additional Product Tag

Tag Name	Color
Add a line	

Рисунок 4.10 – Знайдена форма зі збігом

Як видно, відкрито форму картки товару, у назві якого присутнє слово Bloc.

4.3 Фільтрація результатів

Для зручності користувач може звузити результати пошуку, застосувавши фільтрацію за конкретними таблицями. У спеціальній панелі фільтрів, яку можна активувати за допомогою перемикача «Filter» (рисунок 4.11), користувач обирає моделі, за якими бажає виконувати пошук. Це дозволяє обмежити коло результатів і зосередитися саме на тих даних, що мають найбільшу значущість у контексті виконуваного запиту.

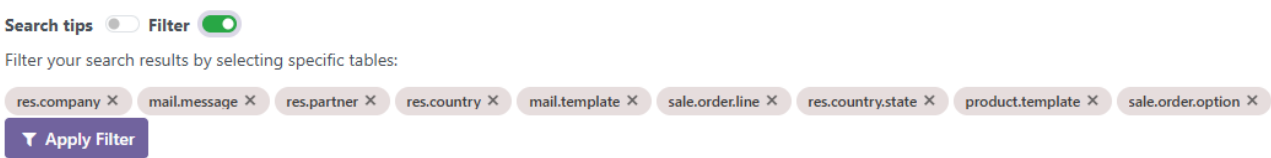


Рисунок 4.11 – Фільтр

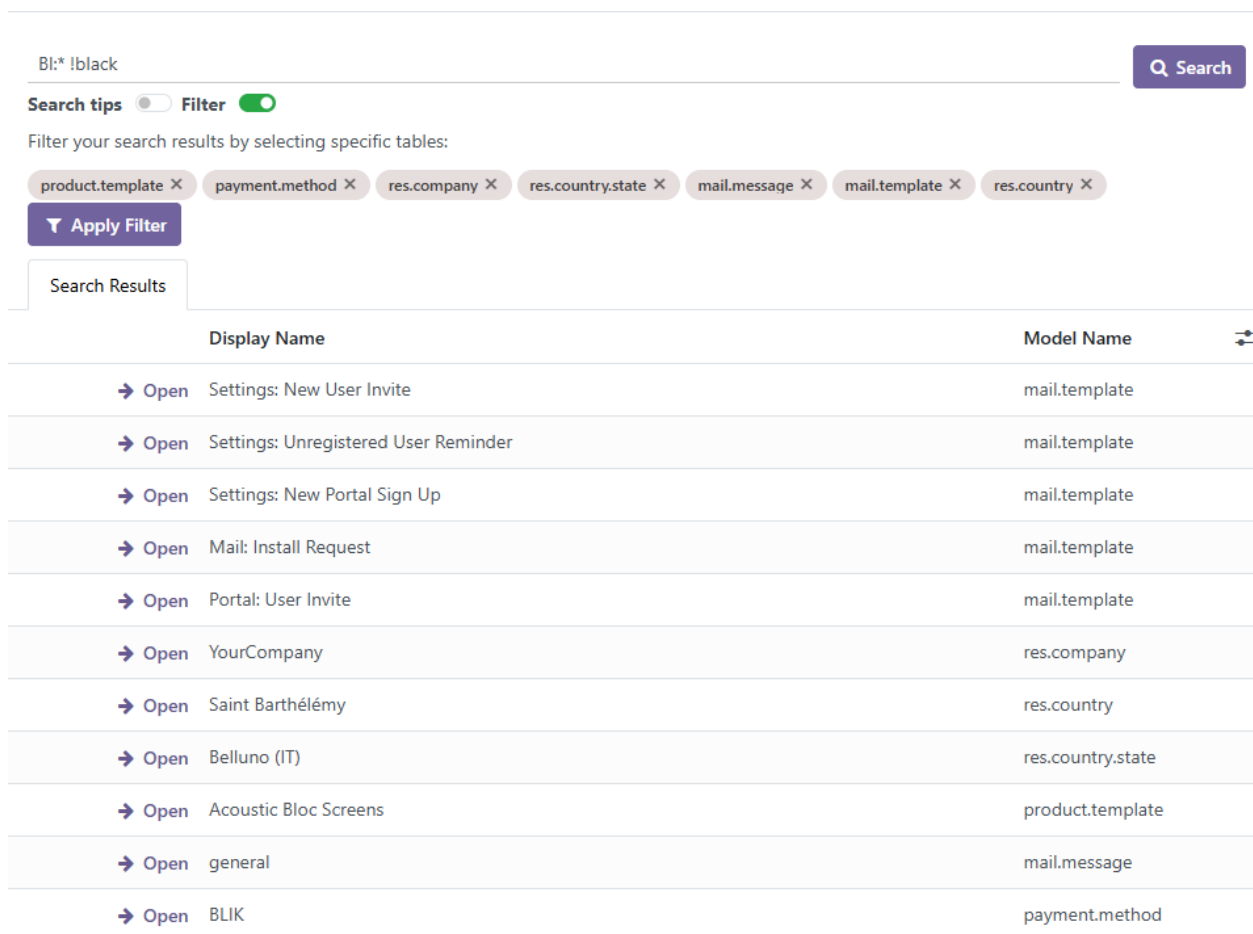
Якщо перші результати були знайдені, наприклад, у таблиці `sale.order.line`, як показано на рисунку 4.12, але користувач розуміє, що записи з цієї таблиці не є для нього релевантними, він може просто виключити її зі списку фільтрів і натиснути кнопку «Apply filter».

1-40 / 41 < >

Display Name	Model Name
→ Open S00001 - [FURN_6667] Acoustic Bloc Screens (Wood) (Deco Addict)	sale.order.line
→ Open S00007 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)	sale.order.line
→ Open S00010 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)	sale.order.line
→ Open S00011 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)	sale.order.line
→ Open S00012 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)	sale.order.line
→ Open S00013 - [FURN_6667] Acoustic Bloc Screens (Wood) (Gemini Furniture)	sale.order.line

Рисунок 4.12 – Результати до фільтрації

Після застосування фільтра система відобразить оновлений список результатів, що дає змогу користувачеві сконцентруватися на більш релевантних даних (рисунок 4.13).



Display Name	Model Name
→ Open Settings: New User Invite	mail.template
→ Open Settings: Unregistered User Reminder	mail.template
→ Open Settings: New Portal Sign Up	mail.template
→ Open Mail: Install Request	mail.template
→ Open Portal: User Invite	mail.template
→ Open YourCompany	res.company
→ Open Saint Barthélémy	res.country
→ Open Belluno (IT)	res.country.state
→ Open Acoustic Bloc Screens	product.template
→ Open general	mail.message
→ Open BLIK	payment.method

Рисунок 4.13 – Результати після фільтрації

Завдяки повній інтеграції з інтерфейсом ODOO користувачі можуть одразу користуватися новими можливостями модуля без додаткового навчання чи адаптації.

Висновки до розділу 4

У четвертому розділі розглянуто особливості роботи користувача з розробленим модулем повнотекстового пошуку та оцінено зручність його використання в умовах реальної експлуатації. Детально описано процес встановлення

модуля, переходу до форми пошуку, введення запитів та отримання результатів, що дозволяє повністю відтворити типові сценарії взаємодії користувача із системою.

Було показано, що інтерфейс модуля організований таким чином, щоб пошук можна було виконати без додаткової підготовки чи навчання. Користувач має змогу формувати запити різного рівня складності, працювати з результатами, застосовувати фільтри та переглядати детальні записи безпосередньо з видачі. Усі дії виконуються у звичному середовищі ODOO, що робить роботу з модулем природною та інтуїтивною.

Описані можливості підтверджують, що створений інструмент не лише розширює діапазон доступних засобів пошуку, а й вписується у загальну концепцію ODOO як платформи з орієнтацією на зручність користувача. Це дає підстави стверджувати, що розроблений модуль забезпечує ефективний доступ до текстових даних та суттєво покращує навігацію великими інформаційними масивами.

5 РОЗРОБКА СТАРТАП ПРОЕКТУ

У цьому розділі розглядається концепція створення стартап-проєкту, що ґрунтується на запропонованому модулі повнотекстового пошуку для платформи ODOO. Розроблений програмний продукт дозволяє значно розширити базові можливості системи, забезпечуючи швидкий, релевантний та масштабований пошук інформації у великих масивах даних. З огляду на зростаючий попит на ефективні інструменти пошуку в корпоративних інформаційних системах, створення окремого стартап-проєкту на основі розробленого модуля є перспективним напрямом для подальшої комерціалізації.

У межах цього розділу буде сформовано унікальну ідею стартапу, визначено ключові напрями застосування продукту, окреслено ціннісні переваги та потенціал його впровадження в різних галузях. Особливу увагу буде приділено аналізу ринку та конкурентного середовища, що дозволить визначити попит на інструменти корпоративного пошуку, а також окреслити позиціонування і унікальну цінність запропонованого рішення.

Окремим етапом є проведення технологічного аудиту, метою якого є оцінка доступності та готовності необхідних технологій для реалізації стартап-проєкту. Також буде сформована ринкова стратегія, орієнтована на виведення продукту на ринок рішень для екосистеми ODOO та суміжних платформ, включаючи визначення каналів комунікації, моделі монетизації та можливих сценаріїв масштабування.

5.1 Ідея стартап проєкту

Стартап-проєкт носить назву “Full-text search”. Логотип стартап-проєкту наведено на рисунку 5.1.



Рисунок 5.1 – Логотип стартап-проекту «Full-text search»

Основна ідея стартап-проекту полягає у створенні модуля повнотекстового пошуку, який інтегрується в екосистему ODOO та значно розширює її стандартні засоби пошуку даних. На відміну від базових алгоритмів фільтрації та доменних запитів, модуль Full-text search пропонує можливість швидкого та точного пошуку інформації в масштабних корпоративних базах даних, використовуючи розширені можливості PostgreSQL.

Даний продукт орієнтований на підприємства та організації, які працюють із великими масивами даних у рамках ERP-системи ODOO та потребують інструменту для оперативного доступу до інформації. Модуль дозволяє здійснювати пошук одразу по всіх необхідних моделях даних, відображати контекст збігів та переходити до результатів у зручному інтерфейсі. Це робить Full-text search універсальним рішенням для бізнесу, де якість і швидкість пошуку безпосередньо впливають на ефективність роботи.

Зміст ідеї стартап-проекту, напрями її застосування та ключові переваги для користувачів наведено в таблиці 5.1.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка універсального модуля повнотекстового пошуку для ODOO з підтримкою індексації, ранжування та пошуку за релевантністю.	Підприємства з великими обсягами даних у CRM, продажах, проєктному управлінні, документації.	Швидке знаходження потрібної інформації, скорочення часу операцій, підвищення ефективності співробітників.

Кінець таблиці 5.1.

Автоматизація пошуку по всіх вибраних моделях та динамічне відображення контекстних фрагментів збігів.	Аналітичні та сервісні компанії, які працюють із великою кількістю записів.	Підвищення точності аналізу даних, покращення якості прийняття рішень.
Інтеграція алгоритмів PostgreSQL (GIN, tsvector, pg_trgm) у зручний інтерфейс ODOO.	Впровадження в корпоративні системи документообігу та бізнес-процесів.	Зменшення навантаження на команду IT, відсутність потреби у зовнішніх пошукових сервісах.
Можливість масштабування та оптимізації під індивідуальні бізнес-вимоги.	Великі підприємства, інтегратори ODOO, мультинаціональні компанії.	Гнучкість, адаптивність, швидка модифікація під конкретні процеси організації.
Екосистемний підхід: модуль легко інтегрується з іншими модулями ODOO та дозволяє будувати додаткові пошукові сценарії.	Розширення корпоративних ERP-систем, автоматизація складних процесів.	Підвищення цінності екосистеми ODOO для конкретного бізнесу, можливість подальшого нарощування функціоналу.

На основі наведених характеристик можна зробити висновок, що модуль Full-text search має широкий спектр можливих сценаріїв застосування та здатен суттєво підвищити ефективність роботи користувачів у різних сферах діяльності. Його універсальність та можливість інтеграції з будь-якими моделями даних у складі системи ODOO роблять продукт привабливим як для малого та середнього бізнесу, так і для великих підприємств, що працюють з великою кількістю неоднорідної інформації.

Для формування повної картини конкурентного середовища було проведено порівняння стартап-проєкту Full-text search із двома найбільш поширеними

рішеннями, що реалізують глобальний пошук у системі ODOO, а саме: модулями Bloomup Fulltext Search та SH Global Search. Обидва модулі належать до сучасного напряму інтегрованих рішень на основі PostgreSQL і забезпечують глобальний пошук без використання зовнішніх сервісів. Разом з тим вони мають низку особливостей, що впливають на зручність використання, масштабованість та продуктивність.

Результати порівняння наведено в таблиці 5.2.

Таблиця 5.2 — Визначення характеристик ідеї проєкту

	Full-text search	Bloomup Fulltext Search	SH Global Search
W - слабка сторона	Потребує коректного налаштування індексування в PostgreSQL для максимальної продуктивності	Ручне додавання полів для кожної моделі; можливі дублікати ідентифікаторів;	Ручний вибір полів для кожної моделі; відсутність складних операторів PostgreSQL; базовий синтаксис пошуку
N - нейтральна сторона	Працює локально в межах PostgreSQL без зовнішніх сервісів	Зручний інтерфейс налаштувань; природний синтаксис запитів; проста інтеграція без зовнішніх залежностей	Зручний глобальний рядок пошуку; мультикомпанійність; автоматичне урахування прав доступу
S - сильна сторона	Автоматичний глобальний пошук по багатьох моделях; релевантність; можливість розширення; мінімальна ручна конфігурація	Гнучкість у виборі полів; хороше ранжування; можливість пошуку в повідомленнях; повна відсутність зовнішніх сервісів	Єдиний глобальний інтерфейс пошуку; миттєвий доступ до будь-яких моделей; сучасний UI; виділення результатів за компаніями

Аналіз даних таблиці показує, що найпоширенішими недоліками конкурентних рішень є висока залежність від ручного налаштування, необхідність окремого вибору полів для кожної моделі, а також обмежений синтаксис пошукових запитів. Модуль Full-text search завдяки автоматизації індексації, підтримці розширених методів PostgreSQL та можливості виконувати пошук одночасно по багатьох моделях дозволяє усунути більшість цих проблем.

5.2 Технологічний аудит стартап-проєкту

Технологічний аудит стартап-проєкту Full-text search спрямований на оцінювання можливості його реалізації з огляду на доступність, придатність і технологічну зрілість необхідних програмних інструментів. Оскільки розроблений модуль інтегрується в екосистему ODOO, яка базується на Python та використовує PostgreSQL як основну систему керування базами даних, вибраний технологічний стек органічно вписується в архітектуру платформи та не потребує залучення зовнішніх інфраструктурних компонентів.

Основою реалізації повнотекстового пошуку є нативні можливості PostgreSQL, зокрема tsvector та tsquery, індекси типу GIN, а також розширення pg_trgm, що забезпечує швидкий пошук за триграмами та підтримку часткових збігів. Завдяки цим інструментам модуль здатен ефективно опрацьовувати великі обсяги даних і виконувати релевантний пошук за мінімальний час. Використання розширених операторів PostgreSQL та можливість оптимізації індексації роблять продукт технічно стійким і придатним для масштабування.

Застосування ORM ODOO забезпечує зручну взаємодію модуля з моделями даних, спрощує доступ до інформації та гарантує коректну роботу з правами користувачів. Інтерфейсна частина модуля побудована за допомогою XML-представлень і технологій Owl, що використовуються в останніх версіях ODOO. Це дозволяє забезпечити сучасний користувацький досвід та підтримувати інтерактивність при роботі з результатами пошуку.

Окрему роль відіграють технічні рішення, пов'язані з автоматизацією індексації даних, підтримкою актуальності текстових представлень та створенням додаткових службових структур (наприклад, JSON-архіву для збереження змін). Ці принципи узгоджуються з архітектурними принципами ODOO та не вимагають модифікації ядра системи, що робить їх придатними для комерційного використання та подальшої підтримки.

Загальна оцінка доступності та придатності технологій, що необхідні для реалізації ідеї стартап-проєкту, наведена у таблиці 5.3.

Таблиця 5.3 – Технологічна здійсненність ідеї проєкту

№ п/п	Компонент проєкту	Технології реалізації	Наявність технології	Доступність
1	Пошукове ядро	PostgreSQL (tsvector, tsquery, GIN, pg_trgm)	Наявна	Доступна безкоштовно
2	Серверна логіка	Python, ORM ODOO, серверні методи	Наявна	Доступна безкоштовно
3	Інтерфейс користувача	XML views, Owl (JavaScript), ODOO WebClient	Наявна	Доступна безкоштовно
4	Автоматизація індексації	Python (automated recompute, batch jobs), тригери PostgreSQL	Наявна	Доступна безкоштовно
5	База даних та структура моделей	PostgreSQL, моделі ODOO	Наявна	Доступна безкоштовно
6	Інфраструктура розгортання	ODOO.sh, локальні сервери (On-premise), Docker	Наявна	Частково безкоштовно
7	Інструменти аналізу та тестування	PyCharm, ODOO shell, PostgreSQL tools (EXPLAIN, ANALYZE)	Наявні	Доступні безкоштовно

Проведений технологічний аудит свідчить, що всі необхідні для реалізації стартап-проєкту Full-text search технології є доступними, зрілими та повністю сумісними з архітектурою ODOO. Використання нативних можливостей PostgreSQL дозволяє забезпечити високу продуктивність і масштабованість пошуку без додаткових інфраструктурних витрат, а модульна структура ODOO забезпечує легку інтеграцію модуля до існуючих бізнес-процесів підприємств. Таким чином, реалізувати стартап-проєкт технічно можливо, а вибраний технологічний стек є оптимальним для досягнення поставленої мети.

5.3 Аналіз ринкових можливостей стартап-проєкту

Для успішного запуску стартап-проєкту важливо проаналізувати ринок, визначити потенційних користувачів та оцінити конкурентне середовище, в якому функціонуватиме продукт. Аналіз ринкових можливостей дає змогу визначити актуальність запропонованого рішення, оцінити його потенціал для розвитку та окреслити ключові переваги, що дозволяють сформувати конкурентну позицію. Важливо також встановити чинники, що впливають на попит у відповідній галузі, виявити обмеження для входу на ринок та зрозуміти специфічні вимоги користувачів, від яких залежатиме ефективність впровадження модуля повнотекстового пошуку.

У межах цього підрозділу буде розглянуто загальні характеристики ринку рішень для системи ODOO, проведено аналіз цільових сегментів та їхніх потреб, а також здійснено порівняння із наявними конкурентами. Це дасть змогу сформувати комплексне уявлення про можливості подальшого розвитку стартап-проєкту Full-text search та встановити перспективність його комерціалізації.

5.3.1 Загальна характеристика ринку корпоративного пошуку в системах ODOO

Сучасні підприємства працюють із великими обсягами структурованої та неструктурованої інформації, що зростають разом із цифровізацією бізнес-процесів. ERP-системи, зокрема ODOO, дедалі частіше стають центральною платформою для

управління операційною діяльністю, що зумовлює потребу у швидкому та зручному доступі до даних. Ефективні методи пошуку відіграють ключову роль у забезпеченні продуктивності користувачів, оскільки вони дозволяють оперативно знаходити потрібні записи, документи, повідомлення або елементи бізнес-процесів без значних часових витрат.

Популярність платформи ODOO зростає завдяки її модульності, гнучкості та здатності масштабуватися під потреби різних категорій бізнесу. За останні роки екосистема ODOO значно розширилася, включаючи впровадження тисяч сторонніх модулів, зростання кількості інтеграторів та активний розвиток спільноти. У таких умовах ринок рішень для розширеного пошуку стає актуальним, оскільки стандартні можливості системи не завжди відповідають вимогам компаній, що працюють із великими обсягами різномірних даних.

Ключовою передумовою появи спеціалізованих модулів повнотекстового пошуку є обмеженість базового пошуку ODOO, який використовує прості фільтри та логіку пошуку лише в окремих полях. У великих інсталяціях це ускладнює доступ до інформації та збільшує час на виконання користувацьких операцій. Крім того, попит на розширені можливості пошуку формується з боку організацій, які використовують ODOO як основну систему документообігу, CRM, управління проектами чи інтеграції внутрішніх комунікацій.

Таким чином, ринок інструментів корпоративного пошуку для системи ODOO характеризується зростанням потреб у швидкому доступі до інформації, високою цінністю релевантності результатів та прагненням компаній до автоматизації внутрішніх процесів. Ці чинники створюють сприятливі умови для розвитку спеціалізованих пошукових модулів, таких як Full-text search, які здатні забезпечити користувачам зручніший, точніший і швидший доступ до корпоративних даних.

5.3.2 Сегментація цільової аудиторії

Цільова аудиторія стартап-проекту Full-text search охоплює широке коло користувачів системи ODOO, для яких важливо забезпечити швидкий і точний доступ до даних у межах корпоративних процесів. Характеристика можливих споживачів демонструє, що попит на функціональні інструменти пошуку формується у різних

сегментах ринку, які відрізняються масштабом діяльності, рівнем цифрової зрілості та обсягами інформації, з якою працюють щодня.

Сегменти цільової аудиторії наведено а таблиці 5.3.

Таблиця 5.3 – Сегменти цільової аудиторії

№	Сегмент	Характеристика сегменту	Основні потреби	Вигоди від використання модуля
1	Малі та середні підприємства	Використовують ODOO як базову ERP	Простий і швидкий пошук	Пошук по моделях, висока релевантність
2	Великі компанії	Великий обсяг даних, складні процеси	Масштабованість і швидкість	Інтеграція з PostgreSQL, швидка індексація
3	Інтегратори ODOO	Обслуговують багато клієнтів	Універсальні інструменти	Готовий модуль для різних бізнесів
4	Організації з великими текстовими даними	Пошук у повідомленнях, документах	Пошук у коментарях і логах	Модуль підтримує такі сценарії
5	Користувачі ODOO Community	Відсутній Enterprise Search	Потужний пошук без Enterprise	Full-text search = альтернатива

Перший сегмент - малі та середні підприємства, що використовують ODOO як універсальну ERP-систему. Для цих організацій особливо важливо мати простий та ефективний інструмент пошуку, який дозволяє швидко знаходити необхідні записи серед зростаючих обсягів даних. Завдяки відсутності зовнішніх залежностей та простоті інтеграції модуль Full-text search є привабливим рішенням для компаній цього сегменту.

Другий сегментом є великі підприємства та корпорації, що опрацьовують великі масиви інформації, включно з документами, листуванням, повідомленнями та записами кількох бізнес-процесів. У таких організаціях пошук відіграє критичну роль у забезпеченні ефективної роботи співробітників. Для них важливо мати гнучкий інструмент, який підтримує релевантність, швидке ранжування результатів та можливість пошуку одразу по декількох моделях. Використання методів PostgreSQL робить запропонований модуль ефективним навіть у великих інсталяціях.

Третій сегмент це компанії-інтегратори та розробники рішень для ODOO, які працюють із різними проєктами клієнтів і потребують універсальних та надійних інструментів для автоматизації пошукових процесів. Для таких компаній модуль Full-text search є можливістю швидко адаптувати рішення під потреби замовника, не вдаючись до створення складних або зовнішніх інфраструктур.

Четвертий сегмент - організації, що працюють із великою кількістю неструктурованої інформації, наприклад, сервісні та консалтингові компанії, відділи підтримки, освітні установи чи медіа-сфери. Для них важливими є можливість повнотекстового пошуку у коментарях, повідомленнях, описах та документах, а також інтуїтивний інтерфейс перегляду результатів.

Останнім, п'ятим сегментом є користувачі ODOO Community, які не мають доступу до інструментів Enterprise Search, але потребують потужних засобів пошуку. Для цієї групи модуль Full-text search заповнює технологічну порожнечу, пропонуючи доступне рішення для глобального пошуку без переходу на комерційну версію системи.

Загалом, потенційний ринок для стартап-проєкту є достатньо широким і охоплює як малий бізнес, так і великі корпоративні структури. Це підтверджує доцільність розробки універсального рішення, яке може адаптуватися під різні сценарії використання в екосистемі ODOO.

5.3.3 Ринкова потреба у модулі Full-text search

Зростання складності бізнес-процесів та обсягів інформації, що зберігається в системі ODOO, формує стабільний попит на інструменти, здатні забезпечити швидкий, точний і релевантний пошук даних. Більшість підприємств, які

використовують ODOO як основну ERP-систему, стикаються з проблемою обмеженості стандартних пошукових методів. Базовий пошук системи, як правило, охоплює лише окремі поля і не забезпечує можливості повнотекстової індексації даних, а також не підтримує розширених сценаріїв пошуку, необхідних для роботи з великими масивами записів та неструктурованої інформації.

Користувачі, які активно працюють із великою кількістю документів, внутрішніх повідомлень або описової інформації, часто витрачають значний час на пошук потрібних записів за допомогою стандартних фільтрів та ручної навігації. Особливо гостро ця проблема проявляється у компаніях, що ведуть інтенсивну комунікацію всередині системи або використовують ODOO для управління проєктами, продажами, договорами чи іншими процесами, що потребують доступу до великої кількості змістовних записів.

Важливою потребою ринку є також пошук, що охоплює відразу кілька моделей даних. У багатьох бізнес-сценаріях інформація розподілена між різними підсистемами: повідомленнями, документами, замовленнями, контактами, завданнями тощо. Відсутність єдиного глобального пошуку ускладнює оперативний доступ до потрібних відомостей, що призводить до зниження ефективності роботи та збільшення навантаження на користувачів.

Окремо варто відзначити сегмент компаній, які використовують ODOO Community. У цій версії системи відсутній комерційний продукт Enterprise Search, тому саме модуль Full-text search заповнює критичну функціональну прогалину, пропонуючи можливість реалізувати повнотекстовий пошук без переходу на Enterprise-ліцензію. Це створює значну додану цінність для цього сегмента ринку.

Потреба у релевантності також є ключовою. Користувачам важливо не просто знайти записи, що містять певні слова, а отримати результати, які найбільше відповідають змісту пошукового запиту. Методом PostgreSQL, на яких базується модуль Full-text search, забезпечують ранжування результатів, підсвітку фрагментів та підтримку складних логічних операторів, що відповідає потребам бізнесу у якісному аналізі даних.

Таким чином, ринкова потреба у модулі Full-text search зумовлена сукупністю факторів: зростанням інформаційного навантаження на користувачів, обмеженістю

стандартних інструментів пошуку в ODOO, відсутністю доступних альтернатив у Community-версії та потребою компаній у релевантному та масштабованому пошуковому інструменті. Це підтверджує актуальність запропонованого рішення та формує основу для його подальшого успішного впровадження.

5.3.4 Аналіз конкурентного середовища

Ринок рішень для повнотекстового пошуку в системі ODOO є досить динамічним та сформованим навколо кількох підходів до реалізації функціоналу пошуку. На відміну від великих ERP-платформ, які мають інтегровані інструменти інтелектуального пошуку або спеціалізовані хмарні сервіси (як-от Dynamics 365 із Azure Cognitive Search чи Oracle ERP Cloud), екосистема ODOO пропонує більш різноманітний спектр рішень. Він включає як базові вбудовані інструменти, так і незалежні модулі від сторонніх розробників, що реалізують різні варіанти повнотекстового пошуку.

У межах цієї роботи ключовими конкурентами стартап-проєкту Full-text search виступають два сучасні модулі глобального пошуку, які були проаналізовані в розділі 2.3, а саме Bloomip Fulltext Search та SH Global Search, а також стандартний пошук, що постачається із системою ODOO. Їхній функціонал, популярність та спектр застосування формують основне конкурентне середовище для запропонованого стартап-проєкту.

Порівнюючи три розглянуті рішення, можна зазначити, що вони орієнтовані на різні категорії користувачів і відрізняються підходами до реалізації пошуку. Стандартний пошук є доступним та простим, але має суттєві функціональні обмеження. Bloomip Fulltext Search пропонує гнучкість і хорошу релевантність, але потребує значної ручної конфігурації. SH Global Search пропонує інтегрований досвід роботи з пошуком, проте його можливості щодо індексації та синтаксису запитів є обмеженими.

У свою чергу модуль Full-text search поєднує ключові переваги конкурентних рішень: простоту інтеграції, релевантність результатів і глобальний пошук, водночас зменшуючи їхні недоліки за рахунок автоматизації індексації, мінімізації ручного налаштування та підтримки розширених можливостей PostgreSQL. Це формує

унікальну ринкову позицію та надає продукту конкурентних переваг у сегменті рішень для ODOO Community та великих інсталяцій.

Порівняння розробленого модуля з конкурентами представлено в таблиці 5.4.

Таблиця 5.4 - Порівняння конкурентних рішень для глобального пошуку в ODOO

Критерій	Full-text search (власний проєкт)	Bloomup Fulltext Search	SH Global Search
Простота впровадження	Висока (автоматизація, мінімальна кількість налаштувань)	Середня (необхідність ручного вибору полів)	Середня (ручний вибір моделей глобальний рядок пошуку)
Масштабованість	Висока (підтримка великих БД, оптимізація через PostgreSQL)	Середня (зниження продуктивності на великих базах)	Середня (залежність від налаштувань)
Глибина пошуку	Висока (релевантність, контекст, підтримка складних запитів)	Середня (добра релевантність, але обмежений контроль над індексацією)	Базова (простий синтаксис, без складних операторів)
Автоматизація процесів	Висока (автоматичне індексування та підтримка актуальності даних)	Низька (створення індексів вручну, конфігурація кожної моделі окремо)	Середня (автоматичне врахування прав доступу, але ручне налаштування полів)
Цільовий сегмент	Малі, великі підприємства, Community-користувачі	Середні компанії без великих БД	Середні інсталяції, компанії з потребою у глобальному рядку пошуку

Порівняння конкурентних рішень демонструє, що найбільші недоліки сторонніх модулів полягають у залежності від ручної конфігурації, обмеженій масштабованості та недостатній підтримці складних пошукових запитів. Модуль Full-text search поєднує високий рівень автоматизації з гнучкими можливостями пошуку та ефективністю при роботі з великими обсягами даних, що дозволяє позиціонувати його як конкурентоспроможний інструмент у сегменті рішень для системи ODOO.

5.3.5 Аналіз тенденцій ринку та перспектив розвитку

Ринок корпоративних систем управління ресурсами підприємства демонструє стабільну тенденцію до розширення функціональних можливостей, орієнтованих на автоматизацію пошуку та обробки великих обсягів інформації. Збільшення кількості даних у сучасних організаціях зумовлює потребу у високоефективних інструментах пошуку, здатних забезпечити швидкий доступ до записів незалежно від їх походження чи структури. У багатьох ERP-системах ця тенденція проявляється у вигляді впровадження розширених когнітивних та семантичних інструментів, що поєднують аналіз текстів, машинне навчання та можливості інтеграції з хмарними сервісами.

У контексті екосистеми ODOO спостерігається зростання попиту на модулі, що дозволяють користувачам працювати з даними у більш гнучкий та ефективний спосіб. Це особливо важливо для компаній, які обирають ODOO як центральну платформу для зберігання інформації про клієнтів, документи, замовлення, внутрішні повідомлення та інші бізнес-процеси. З поширенням практики використання ODOO у сферах продажу, обліку, маркетингу, логістики та сервісної діяльності зростає кількість організацій, що потребують швидкого доступу до релевантної інформації та мінімізації часу на пошук.

Важливою тенденцією є перехід користувачів від локальних рішень до інтегрованих модулів, які працюють у межах бази даних PostgreSQL без необхідності впровадження сторонніх сервісів. Це пов'язано зі зменшенням витрат на інфраструктуру, потребою у підвищенні безпеки та бажанням компаній зберігати контроль над власними даними. Саме тому внутрішні пошукові методи, засновані на

нативних можливостях PostgreSQL, набувають дедалі більшої популярності, що сприяє зростанню ринку рішень, подібних до Full-text search.

Крім того, у спільноті розробників та користувачів ODOO спостерігається сталий інтерес до універсальних модулів пошуку, які здатні працювати з різними моделями системи та підтримувати зручний інтерфейс для взаємодії з результатами. У перспективі ринок ODOO може розвиватися у напрямі інтеграції елементів семантичного пошуку, автоматичної класифікації текстів та більш глибокої аналітичної обробки інформації, що створює додаткові можливості для вдосконалення існуючих рішень.

З урахуванням наведених тенденцій стартап-проект Full-text search має значний потенціал розвитку, оскільки відповідає попиту на універсальні, масштабовані та інтегровані рішення для пошуку інформації в системі ODOO. Зростання популярності платформи, збільшення кількості користувачів ODOO Community та потреба у функціональній альтернативі комерційним продуктам створюють сприятливі умови для комерціалізації модуля та подальшого розширення його функціональності.

5.4 Розроблення ринкової стратегії стартап-проекту

Розроблення ринкової стратегії стартап-проекту Full-text search ґрунтується на визначенні ключових переваг продукту та формуванні підходу, який забезпечить ефективне його виведення на ринок рішень для платформи ODOO. З огляду на проведений аналіз ринку та конкурентного середовища, найбільш перспективною є стратегія, орієнтована на забезпечення простоти впровадження та доступності для широкого кола користувачів. У сучасних умовах саме легкість встановлення та використання програмного модуля є вагомим чинником, що впливає на рішення компаній щодо вибору конкретного інструменту для автоматизації бізнес-процесів.

Основою ціннісної пропозиції продукту є можливість отримати функціональність глобального повнотекстового пошуку без складної конфігурації, додаткової інфраструктури чи зовнішніх сервісів. Модуль працює “з коробки”, автоматично інтегрується в екосистему ODOO та забезпечує релевантний пошук по

кількох моделях одночасно. Такий підхід дозволяє значно зменшити поріг входу для користувачів, тоді як більшість конкурентних рішень потребують ручного налаштування полів, багатокрокової індексації або глибокого розуміння методів PostgreSQL.

З огляду на це, ринкова стратегія стартапу базується на позиціонуванні продукту як доступного та зручного інструменту, орієнтованого передусім на сегмент користувачів ODOO Community, малих і середніх підприємств, а також інтеграторів, які потребують швидкого і простого у впровадженні рішення для своїх клієнтів. Простота встановлення, відсутність потреби в додаткових ресурсах та підтримка глобального пошуку роблять модуль привабливим для компаній, які не мають можливості чи необхідності впроваджувати складні інструменти на кшталт Elasticsearch або комерційного Enterprise Search.

Важливим елементом ринкової стратегії є вибір каналу дистрибуції. Оскільки ODOO Apps Store є основним майданчиком для поширення модулів екосистеми, саме цей канал стає центральним інструментом для просування стартап-проєкту. Такий підхід забезпечує широкий доступ до продукту, спрощує процес придбання та дозволяє охопити глобальну аудиторію користувачів ODOO. Додатково ефективність стратегії може бути підсилена за рахунок надання документації, демонстраційних матеріалів, прикладів запитів та покрокових інструкцій для швидкого старту.

Застосування стратегії простоти та доступності дозволяє сформуванню позитивне сприйняття продукту серед цільових груп користувачів, що зумовлює підвищення конкурентоспроможності стартап-проєкту на ринку рішень для ODOO. Очікуваним результатом реалізації такої стратегії є стале зростання кількості користувачів, розширення партнерських зв'язків з інтеграторами та формування бази для подальшого вдосконалення функціональності модуля.

5.5 Розроблення маркетингової програми стартап-проєкту

Маркетингова програма стартап-проєкту Full-text search спрямована на формування попиту на програмний модуль та забезпечення його ефективного

просування серед цільових сегментів користувачів системи ODOO. З огляду на особливості ринку та конкуренції, ключовим завданням маркетингової програми є інформування потенційних клієнтів про переваги продукту, демонстрація його практичної цінності та створення максимально простого шляху від ознайомлення з модулем до його фактичного використання у бізнес-процесах.

Основним каналом дистрибуції є платформа ODOO Apps Store, яка забезпечує глобальне охоплення аудиторії та є стандартним способом поширення модулів в екосистемі ODOO. Розміщення продукту на цьому майданчику дозволяє користувачам швидко знаходити модуль, ознайомлюватися з описом, переглядати технічні характеристики та безпосередньо завантажувати його. Одним із важливих елементів маркетингової програми є створення якісної презентаційної сторінки продукту, яка міститиме структурований опис функціональності, переваги над конкурентами, короткі приклади використання та скриншоти інтерфейсу. Зрозумілий та інформативний опис дозволяє зменшити бар'єр входу для потенційних клієнтів і підвищити ймовірність встановлення модуля.

Важливою складовою є розроблення демонстраційних матеріалів: коротких відео, які показують процес роботи модуля, приклади реальних пошукових сценаріїв і порівняння часу отримання результатів у різних умовах. Такі матеріали значно підвищують довіру потенційних користувачів, оскільки дозволяють швидко оцінити можливості модуля без необхідності встановлення. Крім того, надання зрозумілої технічної документації, включаючи покрокові інструкції зі встановлення та рекомендації щодо оптимального використання, допомагає підсилити лояльність користувачів і зменшити навантаження на технічну підтримку.

Окрему увагу приділено роботі з інтеграторами та партнерами, які впроваджують ODOO у компаніях різних розмірів. Їхня роль є важливою, оскільки саме інтегратори часто приймають рішення про вибір сторонніх модулів у межах проєктів своїх клієнтів. У межах маркетингової програми доцільно забезпечити цю групу матеріалами для швидкого ознайомлення, наприклад технічними характеристиками модуля, прикладами успішного впровадження та інформаційними буклетами. Побудова партнерських відносин дозволить суттєво розширити

охоплення ринку та інтегрувати модуль у комплексні рішення для корпоративних клієнтів.

Комунікаційна стратегія стартап-проєкту включає також активну присутність у професійних спільнотах ODOO, тематичних форумах, репозиторіях GitHub та каналах соціальних мереж, де користувачі обмінюються досвідом щодо впровадження модулів та оптимізації роботи системи. Розміщення інформації про нові версії, оновлення та покращення сприятиме формуванню іміджу надійного та постійно розвиваного продукту. Додатково важливим є збір зворотного зв'язку від користувачів, який дозволяє оперативно удосконалювати модуль та підтримувати його конкурентоздатність на ринку.

Реалізація маркетингової програми базується на стратегії доступності та простоти, що узгоджується з-поміж ключових характеристик самого продукту. Простий процес встановлення, відсутність необхідності у додатковій конфігурації та підтримка глобального пошуку створюють позитивне враження у користувачів та сприяють формуванню лояльності. Очікуваним результатом упровадження маркетингової програми є підвищення рівня впізнаваності модуля, зростання його популярності на ODOO Apps Store та формування стабільної бази користувачів, що забезпечить подальший розвиток стартап-проєкту.

Висновки до розділу 5

У цьому розділі було проведено комплексний аналіз стартап-проєкту Full-text search із позиції ринкової доцільності, технологічної здійсненності та стратегічних перспектив його розвитку. На основі дослідження ринку корпоративних систем, особливостей екосистеми ODOO та зростання потреби у швидкому доступі до інформації встановлено, що рішення, орієнтовані на глобальний повнотекстовий пошук, мають стійкий попит серед різних категорій користувачів — від малих підприємств до великих організацій, а також серед інтеграторів, що впроваджують ODOO у бізнес-процеси своїх клієнтів.

Сегментація цільової аудиторії дозволила визначити ключові групи користувачів та їхні потреби, що формують основу ціннісної пропозиції стартапу. Проведений аналіз конкурентного середовища показав, що існуючі рішення мають низку обмежень, пов'язаних із ручною конфігурацією, недостатньою релевантністю результатів або обмеженнями в масштабованості. У свою чергу розроблений модуль поєднує простоту впровадження із можливістю виконання ефективного глобального пошуку без необхідності додаткових налаштувань, що надає йому конкурентних переваг.

Розроблена ринкова стратегія базується на позиціонуванні продукту як доступного та зручного інструменту, який працює “з коробки” та не вимагає глибокої технічної експертизи для початку використання. Маркетингова програма, що охоплює канали дистрибуції, засоби комунікації та співпрацю з інтеграторами, спрямована на забезпечення ефективного просування продукту та формування довгострокових відносин із користувачами.

Узагальнюючи отримані результати, можна зробити висновок, що стартап-проект Full-text search має значний ринковий потенціал, відповідає актуальним тенденціям розвитку інструментів пошуку в системах ODOO та демонструє перспективи подальшого вдосконалення і комерційного успіху.

ВИСНОВКИ

1. У результаті проведеного дослідження проаналізовано існуючі підходи до реалізації повнотекстового пошуку в ERP-системах та визначено їхні переваги й обмеження. Порівняння тематичних рішень, зовнішніх пошукових рушіїв та вбудованих можливостей PostgreSQL дозволило обґрунтувати вибір концепції, що не потребує додаткової інфраструктури і забезпечує повну сумісність з архітектурою ODOO.
2. На основі виявлених особливостей платформи ODOO сформовано вимоги до модуля пошуку, які враховують структуру даних, принципи доступу, модульну будову середовища та високу варіативність інформації. Вимоги визначили основні функції програмного забезпечення та стали основою для побудови його архітектури.
3. Розроблено архітектуру модуля, що складається з підсистеми управління даними та підсистеми обробки запитів, які спільно працюють із централізованим JSON-архівом. Така організація забезпечила поділ відповідальностей, узгоджену роботу з великими інформаційними масивами та можливість швидкого пошуку без зміни структури базових моделей ODOO.
4. Реалізовано повний цикл підготовки даних, який включає первинний збір текстової інформації, формування черги та пакетне перенесення записів до архіву. Це дало змогу уникнути перевантаження системи під час індексування та забезпечити стабільність роботи навіть для великих баз даних. У роботі також враховано можливість використання розширення `pg_trgm`, що додатково підвищує ефективність пошуку.
5. Створено й описано процес підтримання актуальності архіву, який передбачає перехоплення змін у моделях ODOO, реєстрацію оновлень у черзі та регулярне перенесення даних у JSON-архів. Такий підхід гарантує узгодженість між фактичним станом системи та індексованими даними, а також виключає ризик їхнього розходження.
6. Розроблено логіку обробки користувацьких запитів, що враховує спеціальні оператори, префіксний пошук, фразові конструкції, а також складні комбінації

логічних умов. Після виконання SQL-пошуку результати проходять додаткову перевірку прав доступу та фільтрацію відповідно до активної компанії користувача, що забезпечує коректність і безпечність видачі.

7. Створено структуру бази даних, що підтримує зберігання запитів, результатів, фільтрів, службової інформації та черги оновлень. Така організація гарантує впорядкованість даних, гнучкість розширення та можливість ефективної взаємодії між усіма частинами модуля.

8. Проведено аналіз інтерфейсу та сценаріїв роботи користувача з модулем. Показано, що новий функціонал органічно інтегрується в середовище ODOO, забезпечує інтуїтивну взаємодію, підтримує історію запитів, фільтрацію результатів, доступ до детальних форм та можливість динамічного налаштування видимих полів. Це підтверджує зручність експлуатації та практичну цінність створеного рішення.

9. У ході тестування модуль продемонстрував стабільну роботу, високу швидкість обробки запитів, коректність індексування та точність відображення результатів у різних сценаріях. Це свідчить про відповідність розробленого програмного забезпечення поставленим вимогам і його готовність до використання у реальних умовах.

10. Результати виконаної роботи демонструють можливість створення ефективного інструменту пошуку в рамках стандартної інфраструктури ODOO без залучення зовнішніх сервісів. Перспективи подальшого розвитку включають вдосконалення логіки ранжування результатів, розширення підходів до морфологічної обробки тексту, підтримку багатомовних конфігурацій користувача та інтеграцію інтелектуальних підказок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hexamethylenediamine Market Size, Share & Trends Analysis Report by Application (Nylon Synthesis, Biocides), by End-Use (Automotive, Textile), by Region, and Segment Forecasts, 2025–2030. – San Francisco : Grand View Research, 2025. – ID: GVR-4-68040-114-8. – URL: <https://www.grandviewresearch.com/industry-analysis/hexamethylenediamine-market>
2. The State of Odoo in 2025 – URL: <https://storeleads.app/reports/odoo>
3. Офіційна документація Odoo – URL: <https://www.odoo.com/documentation/user>.
4. Reis, D. Odoo 15 Development Essentials. - Daniel Reis. - Packt Publishing. - 2021
5. Khodiakova H. V., Khodiakova N. V., Pozdeev V. A. Full text search setup on a website. Control systems and computers. 2021. No. 5-6 (295-296). P. 55–60. URL: <https://doi.org/10.15407/csc.2021.05-06.055>.
6. Лазоренко А. Р. Повнотекстовий пошук інформації в системі ODOO. XI Всеукраїнська науково-практична конференція здобувачів вищої освіти тамолодих вчених з автоматичного управління, Херсон-Хмельницький, 12 квітня 2024 року.
7. Лазоренко А. Р. Методи повнотекстового пошуку інформації в системі ODOO. XXII Міжнародна науково-практична конференція молодих вчених і студентів м. Київ, 22–25 квітня 2025 року.
8. Manning C. D. An Introduction to Information Retrieval / Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. – Cambridge University Press, 2009. – P. 22–36
9. Інтеграція ElasticSearch – URL: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>.
10. Elasticsearch: офіційна документація – 2025. URL: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>.
11. Introduction to Solr. Apache Solr Reference Guide. – 2025. – URL: <https://solr.apache.org/guide/solr/latest/getting-started/introduction.html>
12. Apache Solr Tutorial: What Is, How It Works & What Is It Used For [Електронний ресурс] / Sematext. – 2025. – URL: <https://sematext.com/guides/solr/>

- 13.Venkatapuram S. S. Architecting an apache solr custom implementation. *International journal of research publication and reviews*. 2024. Vol. 5, no. 4. P. 6508–6811. URL: <https://doi.org/10.55248/gengpi.5.0424.1088> .
- 14.Deniz A., Elömer M. M., Aydin A. A. A comparison of Apache Solr and Elasticsearch technologies in support of large-scale data analysis. *Gümüşhane üniversitesi fen bilimleri enstitüsü dergisi*. 2023. URL: <https://doi.org/10.17714/gumusfenbil.1213317> .
- 15.Dynamics 365 documentation. Microsoft Learn. – 2025. – URL: <https://learn.microsoft.com/en-us/azure/architecture/solutions/dynamics-365-scenarios>.
- 16.Step By Step Guide to Dynamics 365 & Azure Integration / IESGP. – 2025. – URL: <https://www.iesgp.com/blog/dynamics-365-azure-integration->.
- 17.Oracle Fusion Cloud ERP: Ultimate Features Guide for 2025 / Zoftware. – 2025. – URL: <https://blogs.zoftwarehub.com/oracle-fusion-cloud-erp-ultimate-features-guide-for-2025/>.
- 18.Elastic Stack Connector – URL: https://apps.odoo.com/apps/modules/18.0/elastick_stack_connector.
- 19.Bloomup Fulltext Search – URL: https://apps.odoo.com/apps/modules/17.0/bloomup_fulltext_search.
- 20.Global Search by Softhealer Technologies – URL: https://apps.odoo.com/apps/modules/19.0/sh_global_search.
21. Python - Офіційна документація – URL: <https://docs.python.org/3/>.
- 22.McKinney, W. Python for Data Analysis. - Wes McKinney. - O'Reilly Media. - 2017. - 544 p.
- 23.Full Text Search. PostgreSQL 16 Documentation – URL: <https://www.postgresql.org/docs/current/textsearch.html>
- 24.Big data full-text search index minimization using text summarization / W. Iqbal et al. Information technology and control. 2021. Vol. 50, no. 2. P. 375–389. URL: <https://doi.org/10.5755/j01.itc.50.2.25470>.
25. Вступ до pg_trgm розширення в PostgreSQL – URL: <https://www.postgresql.org/docs/current/pgtrgm.html>.

26. Ultimate Guide to OWL JS Hooks for Odoo Development / Surekha Technologies. – 2024. – URL:: <https://www.surekhatech.com/blog/owl-js-hooks-for-odoo-development> .
27. OWL: A web framework for structured, dynamic and maintainable applications – Odoo. – 2025. – URL: <https://github.com/odoo/owl>.
28. Leo G. Dynamic inverted index maintenance. 2007. URL: <https://doi.org/10.5281/zenodo.1055164> .
29. PostgreSQL Full-Text Search: Using Ranking Algorithms Effectively – URL: <https://www.slingacademy.com/article/postgresql-full-text-search-using-ranking-algorithms-effectively/>.
30. Büttcher, S., Clarke, C. L. A., Cormack, G. V. A Survey on Indexing Techniques for Large-Scale Text Retrieval. – Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval.
31. Big data processing-beyond batch processing. International journal of recent technology and engineering. 2019. Vol. 8, no. 4. P. 2221–2224. URL: <https://doi.org/10.35940/ijrte.d7903.118419>
32. Сегеда І.В. Проектування та використання баз даних -1. Комп'ютерний практикум: навч. посібник. Київ: КПІ ім. Ігоря Сікорського, 2021. 12 с.

ДОДАТОК А

Тези XXII Міжнародній науково-практичній конференції молодих вчених і студентів, м. Київ, 22-25 квітня 2025 року “СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ”

УКР.НТУУ”КПІ ім. Ігоря Сікорського”_ІАТЕ_ЦТЕ_ТР-41мп

Аркушів 3

Київ-2025

УДК 004.65.2:658.5

¹ Магістрант 1 курсу Лазоренко А.Р.¹ Проф., д.т.н. Шушура О.М.<https://scholar.google.com.ua/citations?user=beUGko0AAAAJ&hl=uk&oi=sra>¹ КПІ ім. Ігоря Сікорського

МЕТОДИ ПОВНОТЕКСТОВОГО ПОШУКУ ІНФОРМАЦІЇ В СИСТЕМІ ODOO

Постановка проблеми та її актуальність. З розвитком цифрових технологій і зростанням кількості інформації, що мають обробляти бізнеси, питання швидкого і зручного доступу до необхідних даних стає все більш актуальним. ERP-системи, що об'єднують записи з різних джерел, відіграють ключову роль у цифровій трансформації бізнесу, тож ефективний пошук у них перетворюється з корисної додаткової функції на необхідність.

Odoo[1] є гнучкою ERP-системою з модульною архітектурою, що дозволяє адаптувати її під потреби компаній різного масштабу. Проте, пошуковий механізм у ній реалізований на рівні окремих моделей, що ускладнює роботу з великими масивами даних та не забезпечує глобального пошуку.

Для забезпечення зручної навігації в масиві даних можуть використовуватися різні підходи до пошуку — від простого фільтрування за заданими параметрами до складних аналітичних механізмів. Однак, при роботі з великими масивами даних, коли запити потребують гнучкості та точності, саме повнотекстовий пошук стає найефективнішим рішенням. Його методи дозволяють не лише знаходити релевантні результати, а й адаптувати алгоритми до специфіки бізнес-процесів.

Аналіз останніх досліджень. Повнотекстовий пошук поступово став невід'ємною частиною сучасних інформаційних систем, забезпечуючи швидкий і зручний доступ до великих обсягів даних. Спочатку ця технологія використовувалася переважно у пошукових системах, таких як Apache Lucene[2]. У 2010 році на його основі був створений Elasticsearch[3] — платформа, що забезпечує швидкий та масштабований пошук у великих обсягах текстових даних. Завдяки високій продуктивності та можливості роботи з розподіленими системами Elasticsearch швидко став популярним серед компаній, яким потрібна ефективна обробка текстової інформації.

Згодом повнотекстовий пошук почали впроваджувати й ERP-системи для спрощення доступу до інформації. Наприклад, у BAS ERP[4] він допомагає швидко знаходити потрібні дані, що підвищує ефективність роботи. Подібні рішення використовуються і в системах електронного документообігу, де повнотекстовий пошук значно скорочує час пошуку файлів та спрощує управління великими обсягами документації.

Крім бізнес-додатків, ця технологія активно застосовується в бібліотеках та архівах, забезпечуючи доступ до великої кількості наукових і історичних матеріалів.

Формулювання мети. Метою є розробка модуля повнотекстового пошуку для системи Odoo, оптимізованого для великих організацій із великими базами даних. Модуль має забезпечувати швидкий доступ до необхідної інформації, з урахуванням її актуальності та ефективного використання ресурсів.

Основна частина. У системі Odoo присутній стандартний точний пошук даних по окремих модулям і застосовується безпосередньо до записів конкретних таблиць. Такий підхід працює ефективно для невеликих обсягів даних, але у великих системах зростає навантаження на базу даних, що може призводити до значних затримок у виконанні запитів. Крім того, він не дозволяє здійснювати глобальний пошук по всіх даних системи, що обмежує можливості користувачів у швидкому доступі до необхідної інформації.

Розроблений модуль повнотекстового пошуку обходить ці обмеження, забезпечуючи централізовану індексацію даних із різних модулів у єдиній таблиці, оптимізованій для швидкого пошуку. Основна перевага модуля — оптимізований механізм обробки та збереження даних, який дозволяє ефективно працювати навіть із великими базами даних. При встановленні модуля основні підготовчі процеси виконуються у фоновому режимі, не блокуючи роботу системи.

Оновлення пошукового індексу також відбувається поступово, за допомогою пакетної обробки даних (batch processing)[5]. Нові, змінені та видалені записи додаються до спеціальної черги оновлень, яка обробляється через заплановані дії кожні три хвилини. Такий підхід зменшує навантаження на систему і гарантує, що результати пошуку завжди актуальні без негативного впливу на продуктивність основних бізнес-процесів.

Крім того, модуль адаптується до можливостей бази даних: якщо в PostgreSQL доступне розширення pg_trgm[6], використовується триграмний індекс, що покращує точність і швидкість пошуку. В іншому випадку застосовується стандартний інвертований індекс, що гарантує коректну роботу навіть у системах із обмеженими можливостями конфігурації. Для кращого розуміння процесу на Рисунку 1 зображено ключові етапи встановлення модуля.

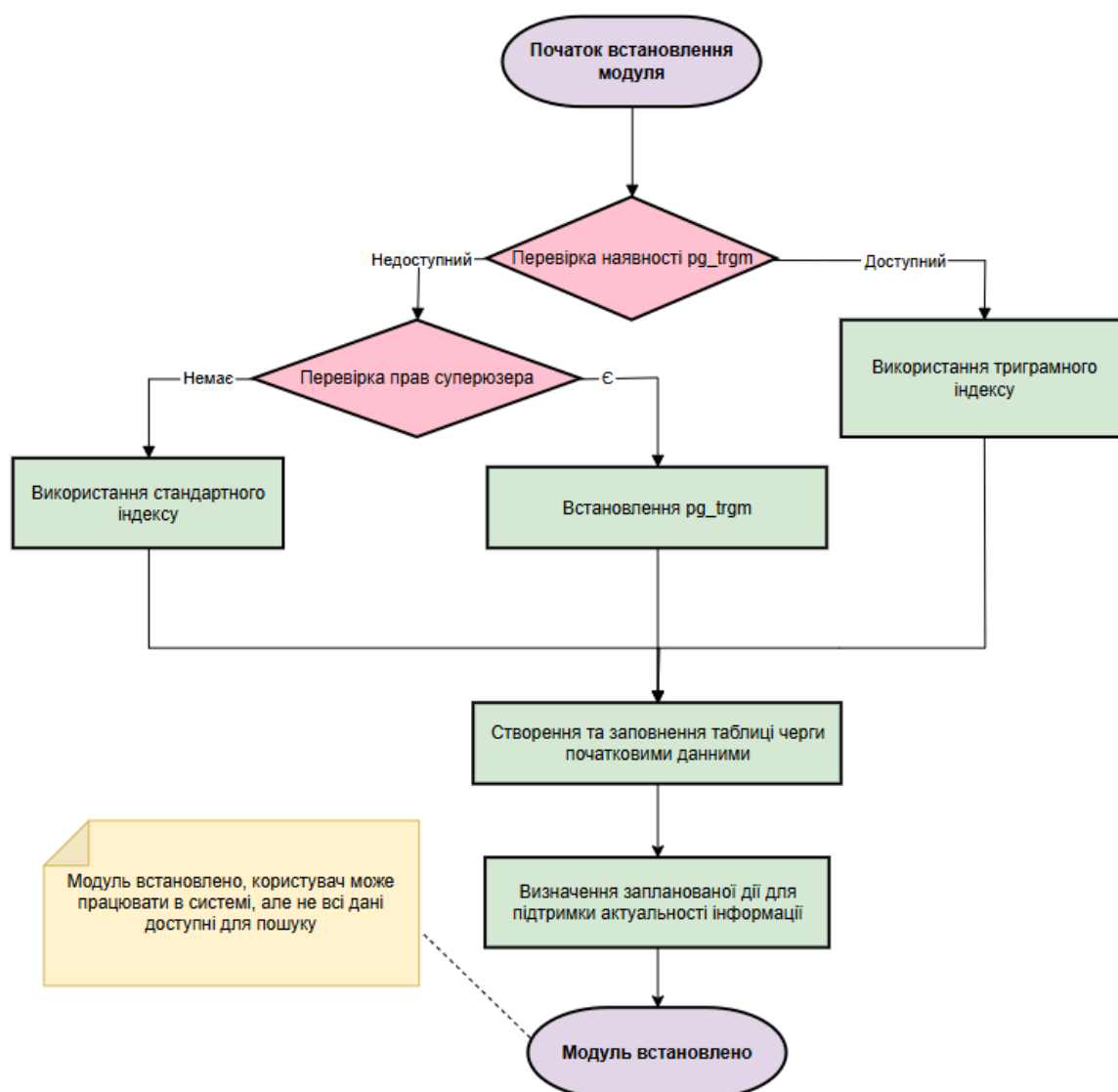


Рисунок 1 – Процес встановлення модуля

Спочатку перевіряється доступність розширення pg_trgm, і залежно від результату система або буде використовувати триграмний індекс, або стандартний. Після цього створюється таблиця черги, що містить початковий набір записів для пошуку, а також налаштовується механізм оновлення даних.

Для забезпечення актуальності даних налаштовані тригери на оновлення, додавання та видалення записів. Це дозволяє системі автоматично оновлювати дані та підтримувати актуальність у режимі реального часу. Процес оновлення пошукового індексу зображено на Рисунку 2.

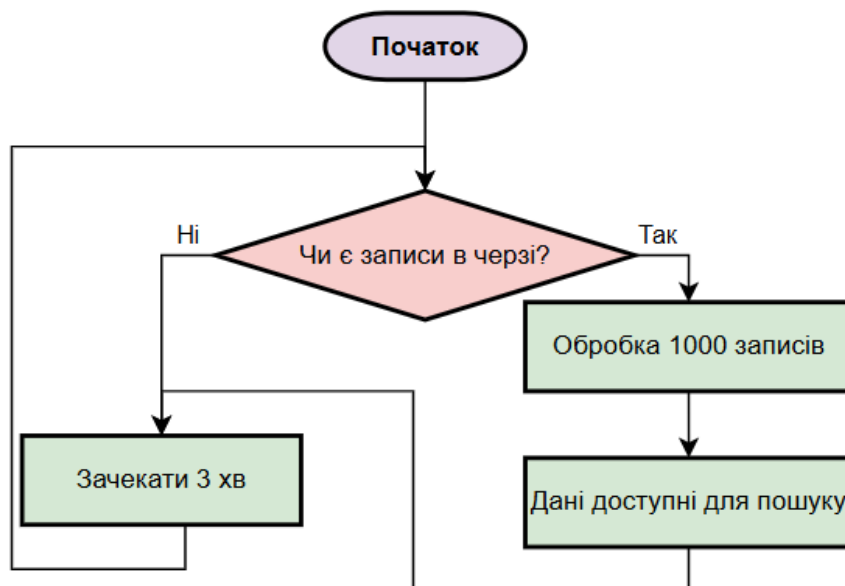


Рисунок 2 – Алгоритм оновлення пошукового індексу

Система регулярно перевіряє чергу і, якщо там є записи на додання змінення чи видалення, опрацьовує їх у пакетному режимі. Це дозволяє підтримувати актуальність даних, не створюючи значного навантаження на систему.

Висновки. Розроблений модуль повнотекстового пошуку для системи Odoo забезпечує швидкий доступ до інформації навіть у великих базах даних, усуваючи проблеми тривалого встановлення та блокування системи під час ініціалізації. Оптимізована архітектура дозволяє виконувати індексацію поступово, без негативного впливу на робочі процеси.

Ключовою перевагою є механізм фонові пакетної обробки даних, завдяки якому оновлення пошукового індексу виконується регулярно і не створює надмірного навантаження на систему. Це гарантує, що дані, доступні для пошуку, залишаються актуальними, а сам процес не заважає користувачам працювати у системі.

Перелік посилань:

1. Odoo documentation. URL: <https://www.odoo.com/documentation/16.0/index.html> (дата звернення 24.20.2025).
2. Apache Lucene. High-performance text search engine library. URL: <https://lucene.apache.org/core/index.html> (дата звернення: 26.02.2025).
3. ElasticSearch Documentation. URL: <https://www.elastic.co/guide/index.html> (дата звернення: 26.02.2025).
4. BAS ERP Documentation. URL: https://its.bas-soft.eu/section/doc_ua (дата звернення: 26.02.2025).
5. Dean J., Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 2008. V. 51, № 1. P. 107–113.
6. Full Text Search PostgreSQL 16 Documentation URL: <https://www.postgresql.org/docs/current/textsearch.html> (дата звернення: 26.02.2025).

ДОДАТОК Б

Довідка щодо прийому до друку статті у науковому журналі «Вісник Кременчуцького національного університету імені Михайла Остроградського» (№6/2025), включеному до переліку фахових видань України категорії Б

УКР.НТУУ"КПІ ім. Ігоря Сікорського" _ІАТЕ_ЦТЕ_ТР-41мп

Аркушів 1

Київ-2025

ДОВІДКА

Видана в тому, що:

1. Стаття авторів **Олексія Шушура, Анни Лазоренко, Лариси Левченко** (Національний технічний університет «Київський політехнічний інститут імені Ігоря Сікорського»), **Сергія Соломахи, Людмили Асєєвої** (Державний університет інформаційно-комунікаційних технологій) **«Методи повнотекстового пошуку інформації в ERP-системах на прикладі ODOO»** отримала позитивну рецензію та буде опублікована у випуску 6/2025 (155) наукового журналу «Вісник Кременчуцького національного університету імені Михайла Остроградського».

Заступник

головного редактора

Виконавчий редактор



Володимир НИКИФОРОВ

Вікторія КОВАЛЬЧУК