МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Центр заочної, дистанційної та вечірньої форм навчання

Кафедра комп'ютерних наук

«До захисту допущено»

В. о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

13 грудня 2023 року

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі с спеціальності 122 Комп'ютерні науки

освітньо-професійної програми «Інформатика»

на тему: Інформаційна технологія аналізу даних відкритих Інтернет-джерел

Здобувачки гр. ІН.мдн-21 Базиль О.О.

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

_____ Олена БАЗИЛЬ
(підпис)

Керівник,
старша викладачка,
канд. фіз.-мат. Оксана ШОВКОПЛЯС _____
(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»
В.о. завідувача кафедри
_____ Ігор ШЕЛЕХОВ
(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувачки групи ІН.мдн-21 Базиль Олени Олександрівни

1. Тема роботи: «Інформаційна технологія аналізу даних відкритих Інтернет-джерел» затверджена наказом по СумДУ від «20» листопада 2023 р. № 1308-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 13 грудня 2023 року
3. Вихідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *1) Аналіз проблеми предметної області, постановка й формування завдань дослідження. 2) Огляд технологій, що використовуються для створення бази даних випускників. 3) Розробка бази даних випускників. 4) Аналіз результатів.*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)_____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «____» _____ 2023 р.

Завдання прийняв до виконання _____ Керівник _____
(підпис) (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	*Аналіз проблеми предметної області, постановка й формування завдань дослідження*		
2	*Огляд технологій, що використовуються для створення бази даних випускників.*		
3	*Розробка бази даних випускників*		
4	*Аналіз отриманих результатів*		
5	*Оформлення пояснювальної записки до кваліфікаційної роботи*		

Здобувач вищої освіти _____ Керівник _____
(підпис) (підпис)

АНОТАЦІЯ

Записка: 53 стор., 16 рис., 2 додатки, 6 таблиць, 31 джерело.

Обґрунтування актуальності теми роботи – Актуальність теми кваліфікаційної роботи пов’язана з розв’язанням важливої задачі: створення бази даних випускників шляхом аналізу отриманих відкритих даних з Інтернет-джерел.

Об’єкт дослідження — процес отримання даних та їх аналіз.

Мета роботи — розроблення інформаційної системи для отримання та аналізу даних з відкритих Інтернет-джерел.

Методи дослідження — алгоритми отримання та аналізу відкритих даних.

Результати — розроблено інформаційну систему, яка збирає відкриті дані з Інтернет-джерел про випускників, обробляє їх, збирає в базу даних, дозволяє переглянути інформацію про місце роботи, посаду, профілі в соціальних мережах.

ВІДКРИТІ ДАНІ, RUBY, REDIS, SIDEQIK, SQLITE.

ЗМІСТ

ВСТУП 5
1 АНАЛІТИЧНИЙ ОГЛЯД 7
1.1 Відкриті дані 7
1.2 Системи Web Mining 9
1.3 Постановка задачі 18
2 ВИБІР МЕТОДУ РОЗВ'ЯЗУВАННЯ ЗАВДАННЯ 19
2.1 Інформаційна модель 19
2.2 Структурно-функціональне моделювання 27
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ 34
3.1 Вибір засобів програмної реалізації 34
3.2 Опис роботи програмного додатку 35
ВИСНОВКИ 45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46
ДОДАТОК А 49
ДОДАТОК Б 51

ВСТУП

Актуальність. Інформація однозначно є одним із самих цінних ресурсів в людському житті.

Аналіз сучасного суспільства, яке характеризується постійними технологічними, економічними, політичними, соціальними та культурними змінами, став більш складним у зв'язку з швидким розвитком Інтернету та інформаційно-комунікаційних технологій, які забезпечують величезне та все більш вагоме джерело інформації, знань та даних. У цьому контексті значну роль відіграють так звані відкриті дані. Модель відкритого управління слугує основою для оприлюднення такого виду інформації [1]. Згідно з Законом України «Про доступ до публічної інформації» (див. статтю 10^1) відкритими даними є «публічна інформація у форматі, придатному для автоматизованої обробки електронними засобами» [2].

Тому аналіз даних, отриманих з Інтернету, є важливим та перспективним напрямком веб-технологій.

Створення бази даних про випускників на основі аналізу даних відкритих Інтернет-джерел дозволить простежити тенденції у побудові кар'єри колишніми здобувачами освіти, забезпечить постійну взаємодію випускників із ЗВО та між собою.

Об'єкт дослідження. Процес отримання даних та їх аналіз.

Предмет дослідження. Методологія отримання інформації з відкритих Інтернет-джерел та керування ними.

Гіпотеза. Отримання даних з відкритих Інтернет-джерел та керування ними можна досягнути шляхом застосування інформаційної технології, що реалізує сервіс по збиранню інформації з соціальної мережі Linkedln, ресурсу дослідників Orcid та найбільшої веб-платформи для хостингу IT-проєктів GitHub.

Новизна. Описане у даній роботі програмне рішення дозволить отримувати інформацію з відкритих Інтернет-джерел і використовувати її для

підвищення іміджу як спеціальності, так і Сумського державного університету в цілому.

Апробація матеріалів роботи. Основні результати роботи оприлюднені та обговорені на XII Міжнародній науково-практичній конференції молодих учених та студенті «Актуальні задачі сучасних технологій», яка проводилася 6 – 7 грудня 2023 року в місті Тернопіль.

Структура. Дана робота складається зі вступу, аналітичного огляду, постановки задачі, вибору методу розв’язування поставленої задачі, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

Зв'язок роботи з науковою темою. Кваліфікаційна робота виконана на кафедрі комп’ютерних наук та пов’язана з виконанням науково-дослідної роботи № 0120U103407 «Застосування технологій games learning, blended learning, віртуальної та доповненої реальності в навчальному процесі» (2020-2025).

1 АНАЛІТИЧНИЙ ОГЛЯД

Швидкий розвиток інформаційно-комунікаційних технологій привів до того, що одним з основних ресурсів розвитку людської спільноти стала інформація, джерелом якої люди все частіше обирають всесвітню мережу. Сучасний Інтернет містить величезні обсяги різноманітних відомостей. Користувачі на різних умовах можуть переглядати та завантажувати файли, аудіо- та відео-документи. Однак це різноманіття даних приховує у собі проблеми, які можуть виникнути не тільки під час аналізу, але й при пошуку необхідної інформації в Інтернеті.

1. Пошук необхідної інформації часто буває часозатратним, оскільки людина не завжди швидко може знайти потрібні йому електронні ресурси. Лише невеликий відсоток посилань серед запропонованих пошуковими системами призводить до необхідних документів. Також важким є пошук неіндексованої інформації такими засобами [3].

2. Проблема виявлення нових знань. Навіть якщо знайдено безліч інформації, для користувача отримання корисних знань є досить трудомістким і непростим завданням. Також існують складнощі, пов'язані з осмисленням відомостей, поняттям тих ідей, вкладених авторами.

3. Проблема вивчення користувачів сайтів пов'язана з наданням споживачеві інформації, яка б йому цікава.

1.1 Відкриті дані

Особливу зацікавленість для відвідувачів Інтернет-ресурсів мають відкриті дані.

Термін «відкриті дані» вперше з’явився в 1995 році, пов’язаний із обміном геофізичними та екологічними даними.

Згідно [4] відкриті дані представляють собою інформацію, до якої користувач має вільний доступ, може її використовувати та розповсюджувати з будь-якою метою. Це сприяє прийняттю виважених ефективних рішень в різних сферах людської діяльності (наприклад, придбання товарів, інформація про найближчий будівельний магазин, замовлення та доставка їжі тощо) [3].

Відкриті дані публікуються в Інтернеті, опираючись на такі принципи [5]:

• відкритості за замовчуванням. Вважається, що вся інформація є відкритою, якщо відсутні причини для її засекречення (наприклад, авторське право, захист персональних даних, державна таємниця тощо);

• оперативності та повноти. Вага та важливість відкритих даних залежить від вчасності опублікування та повного обсягу;

• доступності та готовності до використання. Дані представляються в універсальних форматах даних, що дає можливість користувачам відкривати їх на будь-якому пристрої [4].

До відкритих даних застосовуються такі міжнародні стандарти [3]:

- наявний онлайн доступ 24 години /7 днів на тиждень
- відсутність паролів, рівнів доступу та обмежень, з безкоштовним та анонімним користуванням;
- інформація є машиночитальною з можливістю завантаження у зручних для використання форматах, таких як .doc, .pdf, .jpeg;
- офіційні джерела отримання інформації.

Співробітники офісу ефективного регулювання BRDO проаналізували роботу та вміст 32 сайтів і в своєму дослідженні [3] визначили тематику Інтернет-ресурсів, заснованих на відкритих даних (рис. 1.1).

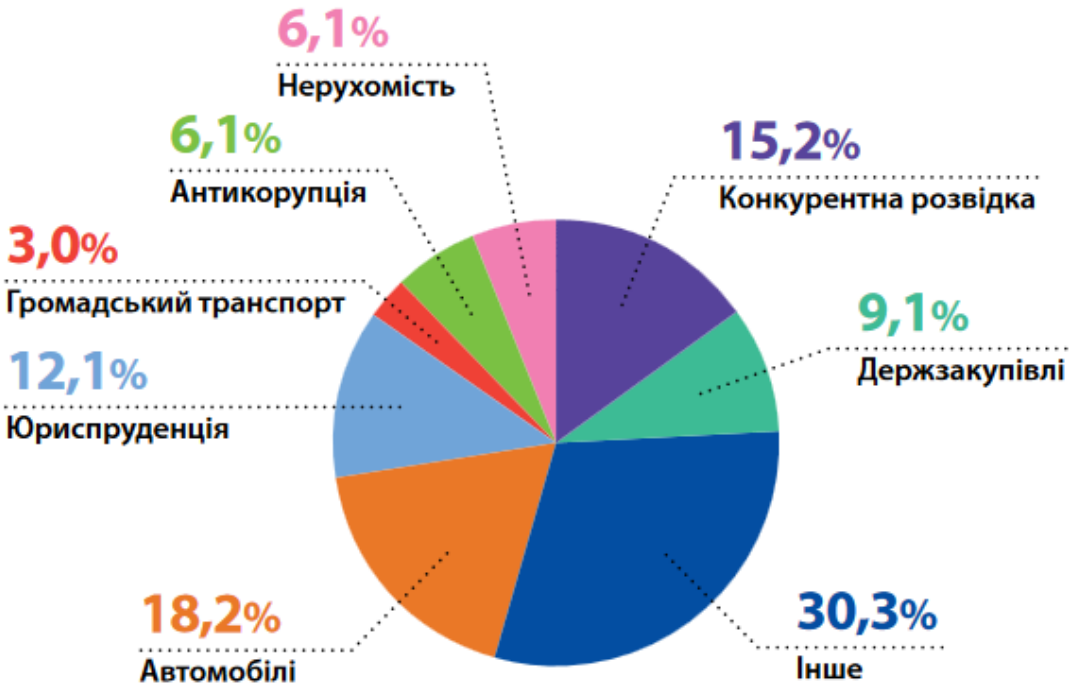

Рисунок 1.1 - Тематика сервісів на основі відкритих даних [3]

1.2 Системи Web Mining

Власників сайтів Інтернет-магазинів, фірм, організацій тощо цікавить інформація про користувачів їх Інтернет-ресурсів.

Здатність визначати інтереси та переваги кожного відвідувача, спостерігаючи за його поведінкою, є серйозною та критичною перевагою конкурентної боротьби на ринку Інтернет-послуг. Тому використання систем Web Mining можуть відповісти на багато питань, наприклад, які інтереси певного відвідувача чи групи користувачів.

Закладам вищої освіти необхідна інформація про випускників для забезпечення постійного зв'язку з ними, створення бази даних з інформацією про здобувачів, які завершили навчання. Це надасть можливість для організації співпраці випускників з кафедрами, факультетами та іншими структурними одиницями ЗВО з метою консультаційно-методичної підтримки.

Крім отримання даних важливим є аналіз даних.

Аналіз даних — це процес систематичного застосування статистичних та/або логічних методів для опису та ілюстрації, узагальнення та повторення та оцінки даних. Відповідно до Shamoo та Resnik (2003), різні аналітичні процедури «забезпечують спосіб отримання індуктивних висновків із даних і розрізнення сигналу (явища, яке цікавить) від шуму (статистичних коливань), присутніх у даних» [6].

Важливим компонентом забезпечення цілісності даних різних типів є точний і відповідний аналіз результатів досліджень. Неналежне дослідження спотворює наукові висновки.

При аналізі необхідно враховувати ряд проблем та вміти їх вирішувати [7]:

- Наявність необхідних навичок аналізу
- Одночасний вибір методів збору даних та відповідного аналізу
- Складання неупередженого висновку
- Невідповідний аналіз підгрупи
- Дотримання допустимих норм з дисциплін
- Визначення статистичної значущості
- Відсутність чітко визначених і об'єктивних вимірювань результатів
- Надання чесного та точного аналізу
- Спосіб подання даних
- Екологічні/контекстуальні проблеми
- Спосіб запису даних
- Поділ «тексту» під час аналізу якісних даних
- Навчання персоналу, який проводить аналізи
- Надійність і валідність
- Обсяг аналізу.

Технологія Web Mining охоплює методи, які допомагають на основі отриманої інформації сприяти отриманню нових знань, робити прогнози, які надалі можна буде використовувати на практиці [8].

Розглянемо основні кроки Web Mining [7]

• вхідний крок (input stage) – отримання вихідної інформації із відкритих Інтернет-джерел (логи веб-ресурсів, електронні документи різних видів);

• крок передобробки (preprocessing stage) – інформація обробляється і приймає вигляд, який використовується для успішної створення тієї чи іншої моделі;

• крок моделювання (pattern discovery stage);

• крок аналізу моделі (pattern analysis stage) – пояснення одержаних результатів.

Це основні кроки, які потрібно пройти для аналізу даних мережі Інтернет. Конкретні процедури кожного етапу залежить від поставленого завдання. У цьому сенсі виділяють різні групи Web Mining.

У Web Mining виділяють [9]:

1. Аналіз використання веб-ресурсів (Web Usage Mining).

2. Вилучення веб-структур (Web Structure Mining).

3. Вилучення веб-контенту (Web Content Mining).

Розглянемо більш детально кожну категорію.

Для аналізу використовується зібрана інформація про події, які відбувалися у програмі або системі. Файл із такими даними називається логом. Метою такого дослідження є виявлення вподобань користувачів під час подорожами між сайтами всесвітньої павутини.

Це потребує здійснення ретельного попереднього оброблення даних, видалення непотрібних записів у файлі зі службовою та статистичною інформацією тощо .

Web Usage Mining включає такі етапи [8]:

- попередня обробка;
- операційна ідентифікація;
- інструменти виявлення шаблонів;
- Інструменти аналізу шаблонів.

Індивідуальні уподобання та звички користувачів мережі Інтернет впливають на те, які саме сайти людина відвідує. Виявивши, які ресурси були відкриті та послідовність дій відвідувача, можна зробити висновок про те, що йому подобається (або йому потрібно). Аналіз результатів серед усіх користувачів веб-ресурсу дає змогу зробити висновок, наскільки ефективно працює веб-сайт, які його користуються популярністю, а які неприваблівні для відвідувачів.

На основі цього дослідження можна зробити роботу сайту більш ефективною: виявити проблеми функціонування або зовнішнього виглядувеб-ресурсу тощо [9].

Цей напрямок Web Mining також часто називають аналізом потоків кліків – упорядкована черга безліч відвідувань вкладок веб-ресурсу, які переглянув відвідувач, потрапивши на веб-сайт [10].

Інформація, яка використовується для такого аналізу, міститься у логах серверів та cookie-файлах [9]. Отримані дані потребують попередньої обробки, оскільки браузер під час відкриття веб-ресурсу створює запити також до всіх складових веб-сторінки, наприклад рисунків чи таблиць. Після того, як відокремлено окремі перегляди ресурсів відвідувачем, їх об'єднують у сесію.

Як тільки дані були очищені та підготовлені для аналізу, необхідно поставити такі питання [7]:

Яка сторінка є спільною точкою для відвідувачів?

• Чи користувачі потрапляють на веб-ресурс через спеціально створену розробниками сторінку або вони відразу відкривають потрібну сторінку?

• Яка послідовність перегляду сторінок? Чи відповідає цей порядок тому, як його запланували розробники?

• З яких зовнішніх ресурсів відвідувач потрапляє на досліджуваний сайт? Які сайти є лідерами (потрапляє найбільша кількість користувачів) і які є аутсайдерами?

• Яку кількість сторінок зазвичай відвідує користувач? Чому саме така кількість? Що потрібно до збільшення кількості відвідуваних вкладок сайту? Як на це впливає дизайн ресурсу?

• Яка тривалість перебування на сайті користувачів? Чи відповідає цей час запланований власниками веб-ресурсу? Чи можливо він недостатній і що є причиною такої поведінки відвідувачів,

• Які сторінки сайту є лідерами перегляду користувачів? Чи є розподіл користувачів за географією входу? Чим це може бути спричинено або так спеціально передбачалося? Які причини є основою того, що користувачі не відвідують сайт?

Лог-файли веб-серверів

Журнал веб-сервера — це файл журналу (або кілька файлів), автоматично створений і підтримуваний сервером. Він містить інформацію про всі події, , які він виконував [11].

Виділяють такі основні типи лог-файлів [12]:

- головний системний лог
- лог завантаження системи
- логи веб-сервера
- журнал ідентифікації користувачів
- логи бази даних
- лог електронної пошти
- логи планувальника Cron

Журнал веб-сервера підтримує історію запитів сторінок. Більш свіжі записи зазвичай додаються в кінець файлу. Зазвичай додається інформація про запит, включаючи IP-адресу клієнта, дату/час запиту, запитувану сторінку, код HTTP, обслуговувані байти, агент користувача та напрямок переходу. Ці дані можна об’єднати в один файл або розділити на окремі журнали, наприклад журнал доступу, журнал помилок або журнал переходів. Однак журнали сервера зазвичай не збирають інформацію про користувача [13].

Ці файли, як правило, недоступні звичайним користувачам Інтернету, лише веб-майстрам або іншим адміністративним особам. Статистичний аналіз журналу сервера може бути використаний для вивчення моделей трафіку за часом доби, днем тижня, джерелом переходу або агентом користувача. Ефективному адмініструванню веб-сайту, достатнім ресурсам хостингу та направленому налаштуванню відвідувань може допомогти аналіз журналів веб-сервера [11].

Аналіз журналів веб-сервера виконується на основі значень, що містяться у файлі журналу, отримує показники про те, коли, як і ким відвідується веб-сервер. Звіти зазвичай генеруються негайно, але дані, отримані з файлів журналу, можуть зберігатися в базі даних, що дозволяє створювати різні звіти на вимогу.

Загальні значення, які зазвичай містяться в журналі веб-сервера [13]:

- Кількість відвідувань і кількість унікальних відвідувачів
- Тривалість відвідування та останні відвідування
- Авторизовані відвідувачі та останні авторизовані відвідування
- Дні тижня та години пік
- Домени/країни відвідувачів хосту.
- Список хостів
- Кількість переглядів сторінок
- Найпопулярніші сторінки входу та виходу
- Типи файлів
- Використовувана ОС
- Використані браузери
- Використовуються роботи
- Пошукові системи, ключові фрази та ключові слова, використані для пошуку аналізованого веб-сайту
- Помилки HTTP

Для лозі будь-якого формату спільними полями є:

- Поле "віддалений хост". [7].
- Поле "дата/час".
- Поле "HTTP запиту", в якому наявні такі можливі чотири складові:
    - метод запиту (правила, якими передається дані запиту);
    - уніфікований індикатор ресурсу (URI);
    - заголовок;
    - протокол [11].

Найчастіше використовуються такі методи запиту [13]:

- GET призначений для отримання зазначеної в параметрах конкретної інформації
- HEAD перевіряє відповідність ідентичність заголовків, указаних за допомогою метода GET
- PUT містить інформацію про стан запиту

В загальному випадку структура Web Mining представлена на рис. 1.2.

Для аналізу веб-структур використовуються різноманітні алгоритми, представлені в таблиці 1.1 [9], [15].

Таблиця 1.1. – Алгоритми для аналізу веб-структур [14], [15]

Назва алгоритму	Автор
In Degree	Marchiori
Сімейство алгоритмів оцінки важливості веб-сторінок Page Rank за допомогою розв'язання систем лінійних рівнянь.	Brin and Page
Аналіз посилань	Kleinberg
Ранжування посилань, яке залежить від запиту Hyperlink Induced Topic Search	Kleinberg
PHITS	Cohn and Chang
Алгоритм аналізу структури зв’язків SALSA	Lempel and Moran
Weighted Page Rank	Wenpu Xing and Ali Ghorbani
Сімейство алгоритмів Page Rank, яке базується на основі посилань відвідувань	Gyanendra Kumar, Neelam Duhan, A. K. Sharma
Weighted Page Rank based on visits of links (VOL)	Neelam Tyagi, Simple Sharma

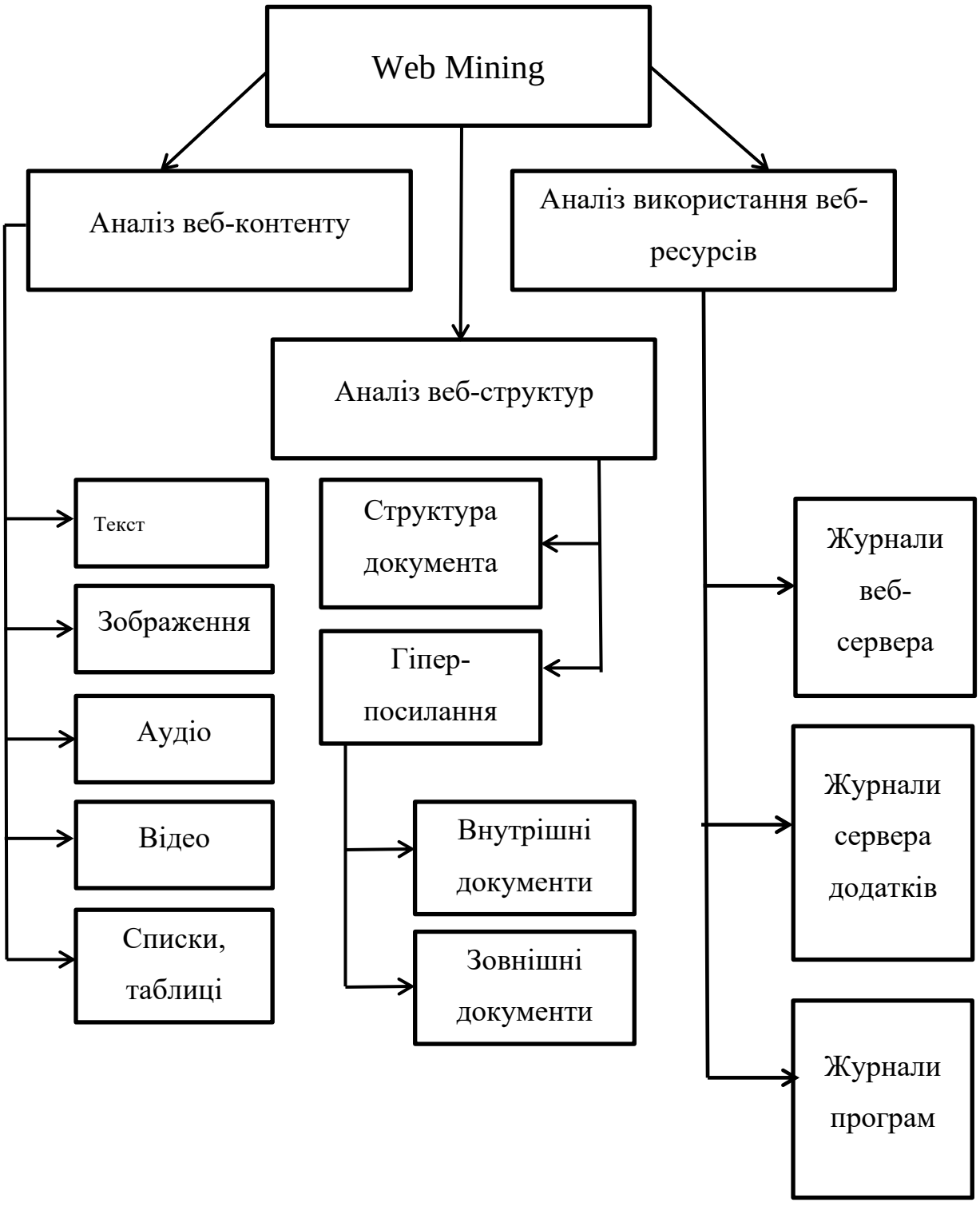

Рисунок 1.2 – Складові Web Mining [16]

1.3 Постановка задачі

Основною метою створення інформаційної технології є оптимізація відстеження та аналізу даних, отриманих з відкритих Інтернет-джерел.

Розроблений програмний продукт має задовольняти такі вимоги:

- ✓ надавати можливість додавати інформацію про нових випускників;
- ✓ слідкувати за досягненнями конкретної людини, зміні місця роботи з метою долучення випускників до профорієнтаційної роботи серед потенційних абітурієнтів, проведення занять;
- ✓ демонстрація інтеграції із зовнішніми ресурсами.

Для досягнення поставленої мети вирішувалися такі завдання:

1. аналіз предметної області;

2. обрання технології та середовища програмування проєкту;

3. створення бази даних випускників кафедри КН Сумського державного університету.

2 ВИБІР МЕТОДУ РОЗВ'ЯЗУВАННЯ ЗАВДАННЯ

2.1 Інформаційна модель

Людину в сучасному світі оточують величезні обсяги різноманітної інформації. Для оброблення та сортування даних відомості організовують у вигляді баз даних, що сприяє структуруванню та адекватному відображенню збереженої інформації. Ознаками бази даних згідно з [17] є:

- база даних містить деяку кількість даних (можливо великі обсяги), які закономірно використовуються для розв'язання визначених задач клієнтів та вирішують задані інформаційні потреби;
- подана в базі даних інформація структурована та пов'язана між собою;
- інформаційні елементи повинні подаватися на машинозчитуваних носіях. Форма подання завдяки комп'ютерно-інформаційним технологіям та використанню необхідного програмного забезпечення дає можливість використовувати дані: обробляти, перетворювати, передавати, змінювати, зберігати тощо.

Існують різні класифікації математичних моделей даних, зупинимося на систематизуванні, представленому на рисунку 2.1.

У роботі використана SQLite. Це бібліотека мовою С, яка реалізує компактну, самодостатню, високонадійну, повнофункціональну систему баз даних SQL [18]. SQLite є найбільш використовуваним механізмом баз даних у світі [19]. SQLite вбудовано в усі мобільні телефони та більшість комп'ютерів і поставляється разом із незліченною кількістю інших програм, якими люди користуються щодня. Формат файлу SQLite є стабільним, кросплатформним і зворотно сумісним, і розробники обіцяють зберегти його таким до 2050 року. Файли бази даних SQLite зазвичай використовуються як контейнери для передачі багатого вмісту між системами [20] і як довготривалий архівний формат для даних . Активно використовується понад 1 трильйон (1e12) баз даних SQLite [21]. Вихідний код SQLite є загальнодоступним і може вільно

використовуватися для будь-яких цілей. Остання версія програми 3.44.2 вийшла в листопаді 2023 року.

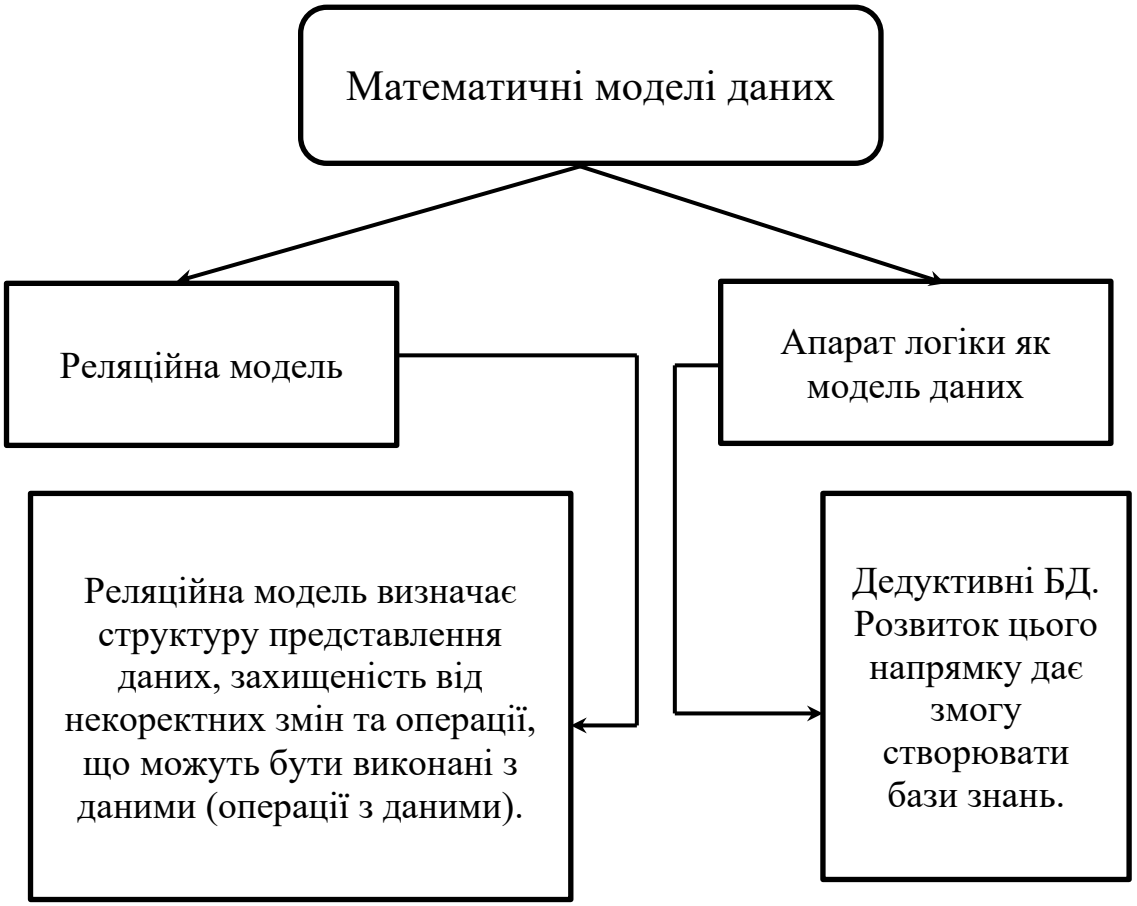

Рисунок 2.1 – Класифікація математичних моделей даних [22]

Коротко процес встановлення та налаштування SQLite описано за посиланням https://coderlessons.com/tutorials/bazy-dannykh/vyuchit-sqlite/sqlite-kratkoe-rukovodstvo.

Наша база даних буде включати такі таблиці: alumns, career_positions, link_viewers, social_links, users. Взаємозв’язок між таблицями бази даних представлений на рисунку 2.2.

В таблицях 2.1 – 2.5 подана інформація про поля таблиць бази даних.

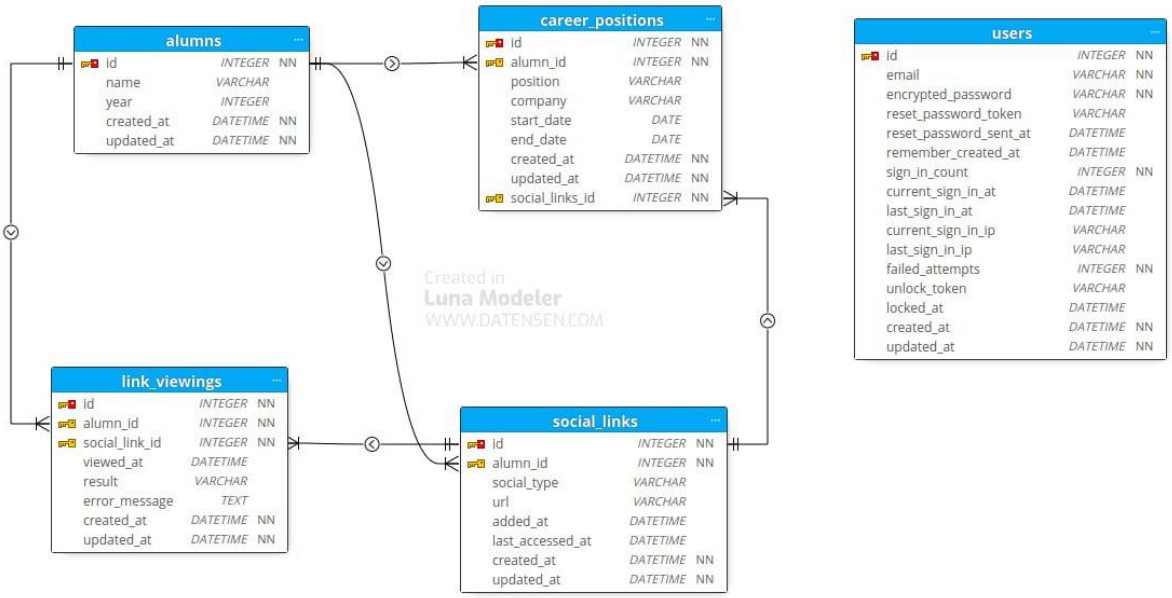

Рисунок 2.2 - Логічна модель бази даних

Таблиця 2.1 – Поля таблиці alumns

Назва поля	Тип даних	Примітка	Призначення
id	цілий	NN	Номер
name	текст		прізвище, ім’я та по-батькові випускника, інформація про якого додається до бази даних
year	цілий		рік закінчення ЗВО
created_id	дата	NN	дата додавання нового користувача
updated_id	дата	NN	дата оновлення інформації про випускника

Перелік (табл. 2.1) описує поля таблиці при додаванні інформації про

нового випускника (прізвище, ім’я, по батькові, рік закінчення Сумського державного університету).

Таблиця 2.2 містить інформацію про посади та організації, в яких працює (працював) випускник. Також тут присутні дані про дату початку роботу, дати додавання та оновлення інформації.

Таблиця 2.2 – Поля таблиці career_position

Назва поля	Тип даних	Примітка	Призначення
id	цілий	NN	номер
alumn_id	цілий	NN	номер випускника
position	текст		остання посада
company	текст		компанія (організація), де працює випускник
start_date	дата		дата початку роботи в організації
end_date	дата		дата завершення роботи в організації
created_at	дата	NN	дата додавання інформації
updated_at	дата	NN	дата оновлення інформації
social_links_id	цілий	NN	id посилання на соціальну мережу

Таблиця 2.3 містить інформацію про посилання на всесвітні мережі, в яких виявлена інформація про випускників нашого ЗВО.

Таблиця 2.3 – Поля таблиці link_viewings

Назва поля	Тип даних	Примітка	Призначення
id	цілий	NN	номер
alumn_id	цілий	NN	номер випускника
social_links_id	цілий	NN	id посилання на соціальну мережу
viewed_at	дата		дата запиту
result	текст		результат пошуку
error_message	текст		повідомлення про помилку
created_at	дата	NN	дата додавання інформації
updated_at	дата	NN	дата оновлення інформації

Таблиця 2.4 містить інформацію сторінки випускників в мережах Linkedln, GitHub, ORCID.

Таблиця 2.4 – Поля таблиці social_links

Назва поля	Тип даних	Примітка	Призначення
id	цілий	NN	номер
alumn_id	цілий	NN	номер випускника
social_type	текст		тип посилання (впливає на скрапер, що буде застосовуватись)
url	текст		посилання на акаунт у соціальній мережі
addet_at	дата		додавання інформації користувача про себе
last_accessed_at	дата		останнє оновлення інформації користувача про себе
created_at	дата	NN	дата додавання інформації
updated_at	дата	NN	дата оновлення інформації

Таблиця 2.5 містить загальну інформацію про користувачів бази даних.

Таблиця 2.5 – Поля таблиці users

Назва поля	Тип даних	Примітка	Призначення
id	цілий	NN	номер
email	текст	NN	електронна пошта
encrypted_password	текст	NN	пароль користувача у зашифрованому вигляді
reset_password_token	текст		токен для відновлення паролю
reset_password_sent_at	дата		дата генерації токену
remember_created_at	дата		дата входу із опцією «запам’ятай мене» (не виходити)
sign_in_count	цілий		кількість входів
current_sign_in_at	дата		дата поточного відвідування
last_sign_in_at	дата		дата останнього відвідування

Продовження таблиці 2.5

Назва поля	Тип даних	Примітка	Призначення
current_sign_in_ip	текст		IP поточного входу в систему
last_sign_in_ip	текст		IP останнього входу в систему
falled_attempts	цілий		кількість невдалих спроб для входу поспіль (захист від підбору пароля)
unlock_token	текст		токен для розблокування
locked_at	дата		дата блокування облікового запису
created_at	дата		дата створення користувача
updated_at	дата		дата оновлення даних про користувача

2.2 Структурно-функціональне моделювання

Завдання на створення кінцевого продукту, а саме проєктування та наповнення бази даних про випускників кафедри комп'ютерних наук Сумського державного університету представимо у вигляді UML – діаграми.

Згідно з [23] UML (Unified Modeling Language) — представляє собою уніфіковану мову моделювання, що застосовується для представлення результатів роботи процесів та систем у наочному та простому вигляді для полегшення сприйняття поданої інформації, тобто забезпечує візуалізацію процесу програмного забезпечення за допомогою набору діаграм.

Переваги використання UML – діаграм [24]:

- ✓ дозволяє швидко долучити нових членів команди.
- ✓ легка навігація вихідним кодом.
- ✓ дозволяє спланувати нові види робіт до початку етапу програмування.
- ✓ полегшує спілкування з фахівцями та замовниками.

Основними типами елементів, які використовуються в UML – діаграмах є:

- ✓ лінії;
- ✓ написи;
- ✓ значки;
- ✓ фігури.

Для побудови UML – діаграм використовують різні програми, серед яких виділяються такі як: Dbdiagram.io, xmind.net, Microsoft Visio, Google Drawings, Diagrams.net.

На рисунку 2.3 представлена принципова схема. Sikekiq представляє собою ефективний, простий в інтеграції і кращий планувальник завдань з відкритим вихідним кодом. Зрозумілий інтерфейс дозволяє користувачу налаштувати програмний засіб для виконання багатьох завдань одночасно в

тому самому процесі з кількома потоками. Він дозволяє одночасно виконувати такі потоки завдань: заплановані роботи, обробка помилок, Ruby API, обробка численних потоків тощо.

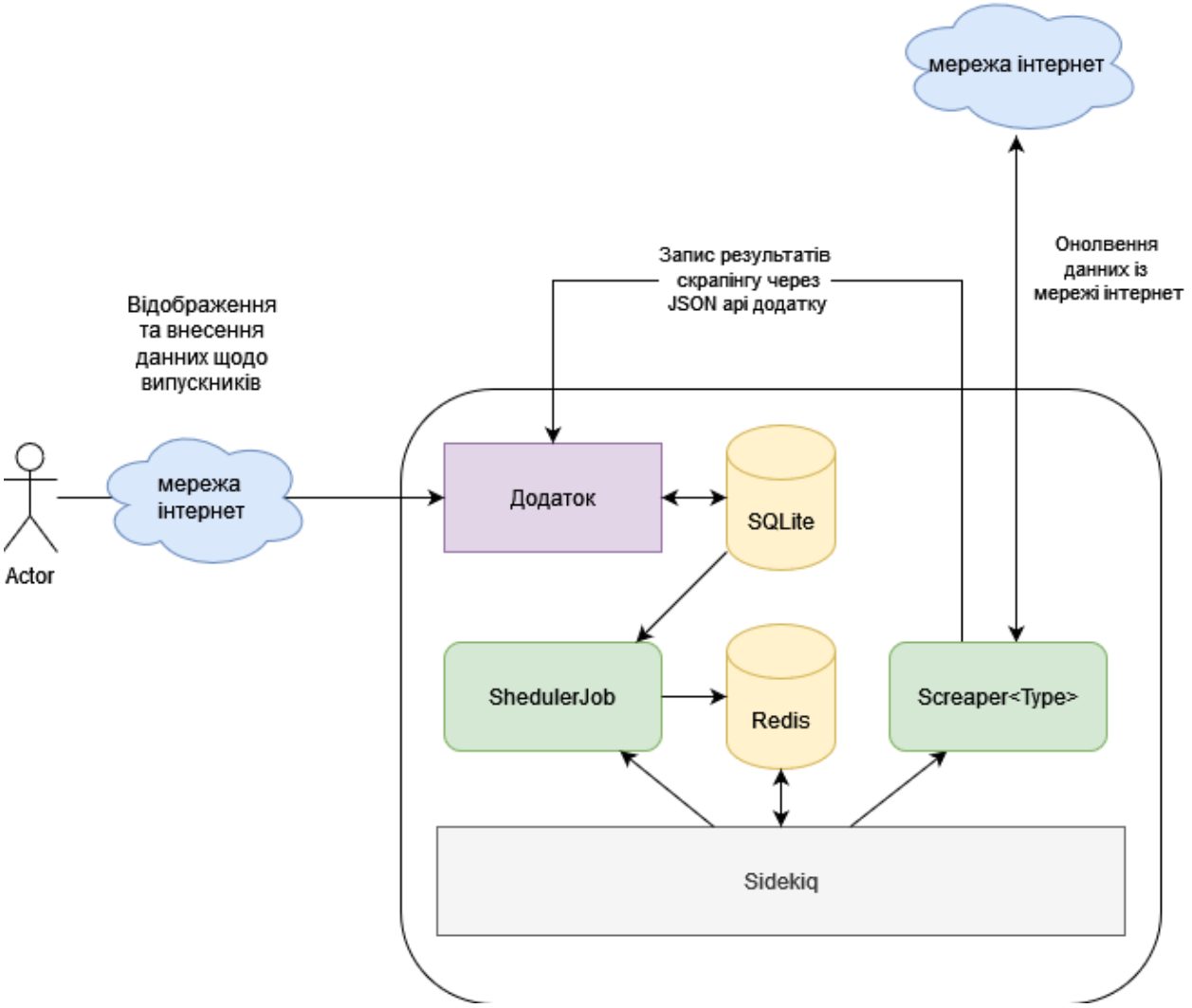

Рисунок 2.3 – Принципова схема роботи програми

Даний планувальник створений на мові програмування Ruby.

Основна перевага Sikekiq – це можливість виконання асинхронних завдань в фоновому режимі. Цей програмний засіб дуже зручно використовувати для обробки великих обсягів інформації (наприклад, зчитування даних з великого документа та занесення їх до бази даних), щоб користувач отримав відповідь не чекаючи, поки виконається вся робота.

Для використання Sikekiq необхідно мати встановлений фреймворк Ruby on Rails [25]. Він написаний мовою програмування Ruby, реалізує шаблон MVC для веб-застосунків, а також забезпечує їх інтеграцію з веб-сервером і сервером баз даних. Ruby on Rails є відкритим програмним забезпеченням та поширюється під ліцензією MIT.

У даній роботі був використаний Ruby on Rails версії 7.1.1.

MVC (або Model-View-Controller) представляє собою архітектурний шаблон [26]. Він поділяє дані програми та керуючої логіки на три окремих компоненти: модель, вигляд та контролер - таким чином, що модифікація кожного компонента може здійснюватися незалежно. Взаємодія компонентів MVC подана на рис. 2.4.

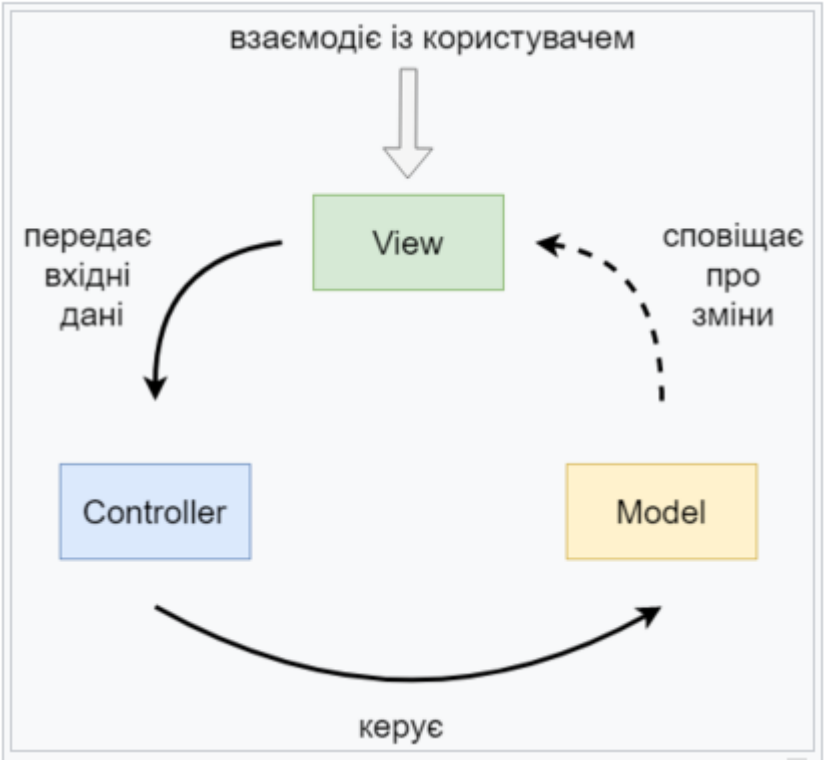

Рисунок 2.4 - Діаграма взаємодії між компонентами шаблону MVC [26]

В основі gem'a для Rails знаходиться модель акторів. Для його використання необхідно додати в Gemfile команду:

gem 'sidekiq'

Для коректної роботи Sidekiq також мати інстальоване та запущене сховище структури даних Redis. Це програмний додаток із відкритим вихідним кодом (ліцензія BSD). Його можна використовувати як базу даних у пам’яті, яка зберігається на диску, кеш-пам’ять, брокер повідомлень і механізм потокового передавання.

Модель даних, що використовується в Redis, дозволяє підтримувати багато різних видів значень: рядки, списки, набори, відсортовані набори, хеші, потоки, HyperLogLogs, растрові зображення.

Redis підтримує значну кількість мов програмування, для яких існують прив'язки бібліотек, найвідоміші з них такі:

- C
- C++
- C#
- Python
- Ruby
- ActionScript
- Java
- PHP
- Node.js.

Перша версія Redis, яка була написана Сальваторе Санфіліпо, побачила світ в 2009 році.

Scheduler – це розширення для Sidekiq, яке дозволяє запланувати або запускати завдання у визначений час або з певними інтервалами. Додаток SchedulerJob запускається один раз на добу. Він приводить в дію Straubing for Linkedln, і за день знаходить та обробляє інформацію трьох профілей випускників, дані яких розміщені на платформі соціальної мережі Linkedln.

API (прикладний програмний інтерфейс) – це засіб для обміну даними між програмами чітко визначених способами.

База даних SQLite описана в п. 2.1.

Схема обміну даними API представлена на рис. 2.5.

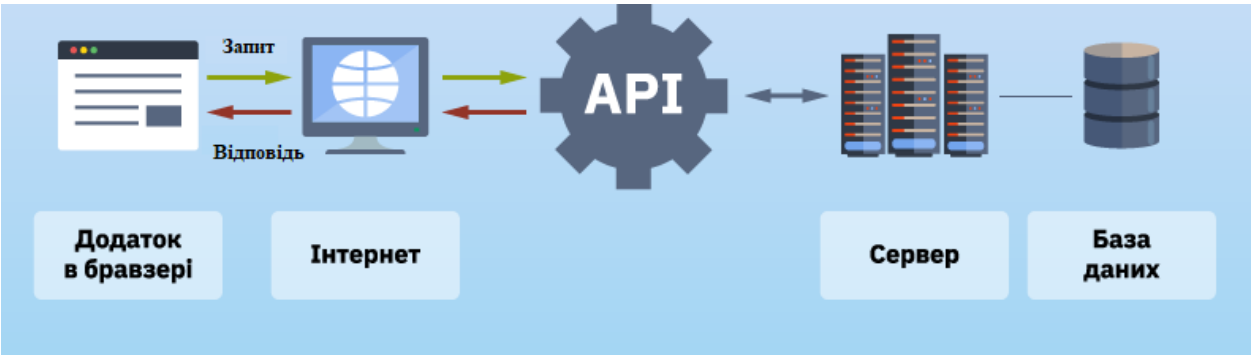

Рисунок 2.5 – Обмін даними API [27]

JSON (JavaScript Object Notation) - це текстовий формат для впорядкованого зберігання та транспортування структурованих даних із синтаксисом об’єктів JavaScript, який був створений американським програмістом Дугласом Крокфордом. JSON легко поєднується з будь-яким сучасним середовищем програмування, зокрема, код для введення та обробки даних у цьому форматі присутній у мовах Python, PHP, Java та Ruby.

JSON помітно спрощує та прискорює взаємну передачу даних у веб-додатках (наприклад, надсилання деяких даних із сервера до клієнта, щоб їх можна було відобразити на веб-сторінці, або навпаки). Більше того, завдяки текстовому вигляду рядка дані JSON можна легко передавати через будь-які інші канали обміну інформацією у всесвітній павутині.

JSON існує як рядок — це корисно, коли ви хочете передати дані через мережу. Його потрібно перетворити на рідний об’єкт JavaScript, коли ви хочете отримати доступ до даних. Це не велика проблема — JavaScript надає глобальний об’єкт JSON, який має методи, доступні для перетворення між ними.

Рядок JSON можна зберігати у власному файлі, який, по суті, є лише

текстовим файлом із розширенням .json і типом MIME application/json. також цей формат може бути представлений в інших типах файлів (наприклад, .html), відображаючись у вигляді рядка JSON або об'єкта. Важливою особливістю стандарту є те, що рядок JSON виглядає як звичайний текст, який легко читається людиною – як і у випадку з будь-якими іншими текстовими форматами.

Файл JSON можна включити ті самі основні типи даних, що й у стандартний об’єкт JavaScript — рядки, числа, масиви, логічні значення та інші об’єктні літерали. Це дозволяє побудувати ієрархію даних.

JSON оптимально взаємодіє з AJAX (асинхронний JS та XML), разом вони забезпечують асинхронне завантаження даних у фоновому режимі. Така функція дозволяє сайтам та веб-програми оновлювати інформацію без обов'язкового перезавантаження сторінок. Крім того, за допомогою JSON користувачам доступний запит даних із стороннього домену. Зробити це можна через тег <script>, а сам метод називається JSONP - це єдиний допустимий спосіб обміну даними між доменами.

Формат JSON має декілька видів структури, а саме [28]:

Пара "ключ-значення" ("key" : "value"), в якій ключі є рядками, а значення - допустимим типом даних JSON.

Ключі та значення JSON у різних мовах програмування називаються по-різному: структура, запис, словник, асоціативний масив, послідовність, вектор, список тощо. Така універсальність дозволяє легко обмінюватись даними між програмними середовищами через JSON.

Використання хуків, коли користувач задає URL та GitHub, то формується http request, в якому ми вказуємо запит щодо сторінок випускників та події, які відображаються на цих сторінках. Отримані дані наповнюють нашу базу даних.

Організація People Data Labs заснована в 2015 році Генрі Нев’ю та Шоном Торном. Сайт компанії розміщений за посиланням https://www.peopledatalabs.com/.

Платформа People Data Labs дозволяє отримати відкриті, правдиві, персональні дані про людей із відкритих джерел, тобто сприяє підбору висококваліфікованого персоналу для компаній, які це потребують. яке мало допомогти компаніям краще зрозуміти та знайти кандидатів.

Збирається в одному місці інформація про людину, а саме, прізвище та ім’я, рік народження, робоча електронна пошта, профілі із найбільшої у світі онлайн-мережі професійних контактів LinkedIn, GitHub, Facebook, Twitter, дані про місця роботи, посади здобувачів.

Понад 3,2 мільярда профілів, зібраними People Data Labs, використовуються провідними компаніями для вдосконалення платформ рекрутингу, посилення моделей штучного інтелекту, створення власних аудиторій тощо.

Однак перегляд та використання інформації, отриманої від People Data Labs, є платною послугою. На безкоштовній версії можна переглядати та копіювати дані із профілей 100 людей на місяць.

Для автоматизації роботи з веб-браузером використовується інструмент Selenium, який призначений для тестування та роботи з програмними додатками. Він складається з таким компонентів, як Selenium IDE, Selenium Client API, Selenium Remote Control, Selenium WebDriver, Selenium Grid. В роботі був використаний Selenium WebDriver для емуляції роботи користувача на сайті orcid.org. На жаль, сайт вимагає використання JavaScript для перенаправлення при першому вході користувача, тож отримати данні безпосередньо http запитом неможливо.

Натомість данні на github.com публічно доступні, тож для отримання інформації використовується http запит та бібліотека Nokogiri для аналізу отриманого документа.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Вибір засобів програмної реалізації

Для розроблення додатку була обрана мова програмування Ruby версії 3.2.2. Вона представляє собою стабільну, просту та потужну об’єктно-орієнтовану мову програмування. Вказана мова була створена Юкіхіро Мацумото. Роботу над Ruby програміст розпочав 24 лютого 1993 року і в 1995 році вийшла загальнодоступна версія цього продукту.

Ruby об’єднала в собі найкращі переваги таких мов програмування як Smalltalk, Perl, Java, Smalltalk, Python, Eiffel, Ada і Lisp.

Розробники програмного забезпечення зазвичай використовують Ruby для написання серверів, експериментів із прототипами та для вирішення повсякденних завдань програмування. Оскільки Ruby повністю інтегрована об’єктно-орієнтована мова, це дозволяє їй добре масштабуватися.

Особливості Ruby:

- Простий синтаксис,
- Основні функції об’єктно-орієнтованого програмування (класи, методи, об’єкти тощо),
- Спеціальні функції орієнтованого орієнтування (одиночні методи, перейменування тощо),
- Перевантаження оператора,
- Обробка винятків,
- Ітератори та закриття,
- Великий набір готових рішень
- Динамічне завантаження (в залежності від архітектури),
- працює на різних операційних системах: Unix, Windows, DOS, macOS, OS/2, Amiga тощо.
- реалізовано багато шаблонів програмування,
- інтерпретатор мови Ruby поширюється як вільне програмне забезпечення.

Для авторизації в Ruby on Rails 7.1.1 використовувася devise. Він представляє собою гем, написаний мовою програмування Ruby. Основні особливості Devise:

- заснований на Rack;
- є закінченим MVC-рішенням, заснованим на Rails;
- дозволяє вхід до системи за кількома моделями одночасно;
- заснований на модульності: використовує тільки те, що вам дійсно потрібно.

Devise надає багато корисних функцій, таких як обробка сеансів користувачів і додавання підтримки стороннього входу за допомогою OAuth за допомогою gem OmniAuth. Devise також має вбудовані модулі для таких функцій, як скидання забутих паролів, відстеження кількості входів і часових позначок, визначення тайм-аутів, блокування облікових записів тощо.

Devise робить автентифікацію користувача такою ж простою, як ініціалізація gem і створення моделі користувача з необхідними функціями. Якби ви створювали автентифікацію користувача з нуля, вам довелося б написати код і тести для всіх потрібних вам функцій, а також обробляти всі крайні випадки обробки сеансів, зберігання файлів cookie та захисту даних [29].

3.2 Опис роботи програмного додатку

Принцип роботи програмного додатку представлений на рис. 3.1.

Програмний додаток виконує такі функції:

- управляє даними;
- презентує отримані дані;
- управляє чергами завдань;
- має сховище для збереження даних;
- запускає на SchedulerJob та Scrapper виконання періодичних завдань;
- здійснює http запит до соціальних мереж LinkedLn, Orcid, GitHub.

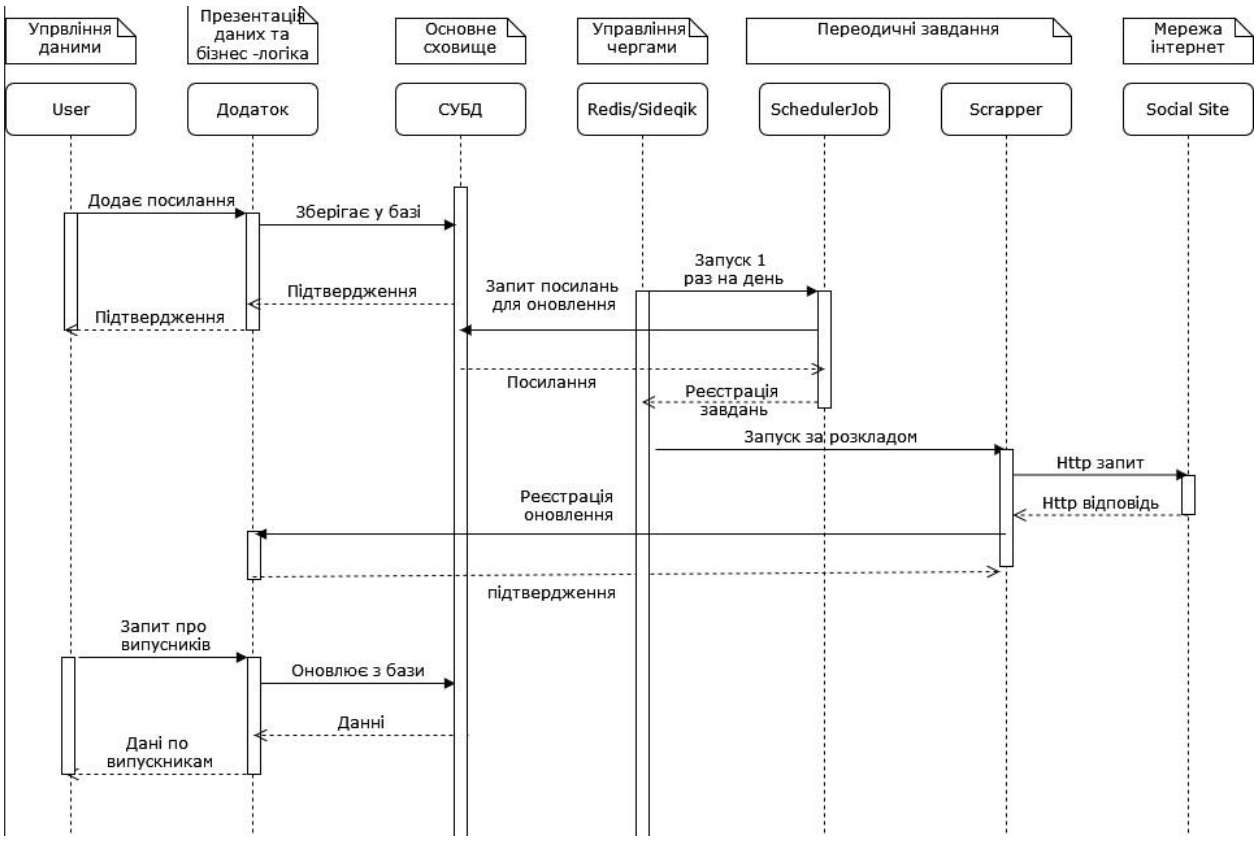

Рисунок 3.1 - Принцип роботи програмного додатку

LinkedlnScrapper дозволяє отримувати інформацію із профілів користувачів соціальної мережі Linkedln. На жаль, Linkedln значно обмежує можливості автоматичного перегляду сторінок. Тож для отримання даних використовується JSON API від PeopleDataLabs. Безкоштовно можна переглянути до 100 профілів на місяць.

Завдання SchedulerJob запускається один раз на добу та планує запуск оновлення профілів користувачів із на поточну добу із урахуванням поточних обмежень. Наприклад – не більше 3 профілів LinkedIn, 100 профілів ORCID, тощо. При цьому автоматично обираються профілі із найдавнішою датою оновлення чи ті, що небули відвідані жодного разу.

Детальна інформація про СУБД SQLite, Redis, Sideqik, JSON API наведена в п. 2.2.

Статистика роботи Sideqik представлена на рис. 3.2. Як видно з рис. 3.2

опрацьовано 754 акаунти, із 42 спроби були невдалими. Інтервал опитування складає 20 секунд.

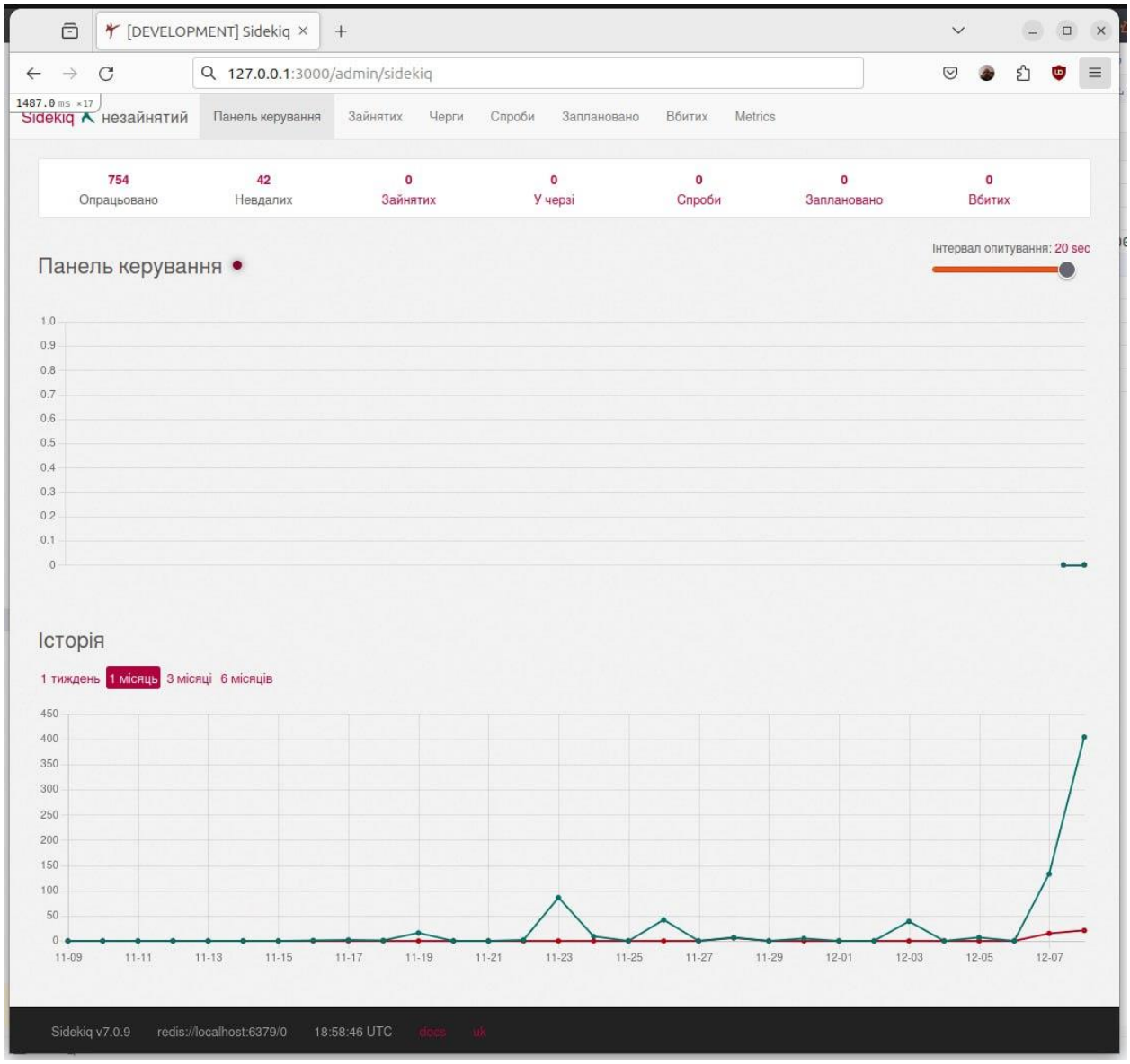

Рисунок 3.2 – Панель керування та історія використання Sideqik

Вміст Scrapper'а, який звертається до мережі Github, наведена нижче. Він аналізує 100 сторінок користувачів на день. Зібрані дані повертаються в основну програму у форматі JSON.

```
# frozen_string_literal: true
class GithubScraper < LinkScraper
  def initialize(url)
```

```
    @url = url
  end

  def self.match_url?(url)
    url.match?('github.com')
  end

  def run_per_day
    100
  end

  def scrape_data
    # Логіка для скрапінгу даних зі сторінки Github
    # Повернення зібраних даних у форматі JSON
  end
end
```

Детальну інформацію про веб-скрапінг та його практичну реалізацію можна отримати в книзі [30].

Вміст файлу orcid_scrapper.rb наведений в додатку А.

Користувач використовує свій логін (електронну пошту) та пароль для входу в базу даних. Якщо акаунту немає – існує реєстрація. Якщо користувач – його можна відновити, відправивши дані про відновлення на електронну пошту.

На рис. 3.3 представлена сторінка входу.

Рисунок 3.4 ілюструє процес створення нового користувача. Додаємо інформацію: прізвище, ім'я, по батькові, рік закінчення Сумського державного університету, посилання.

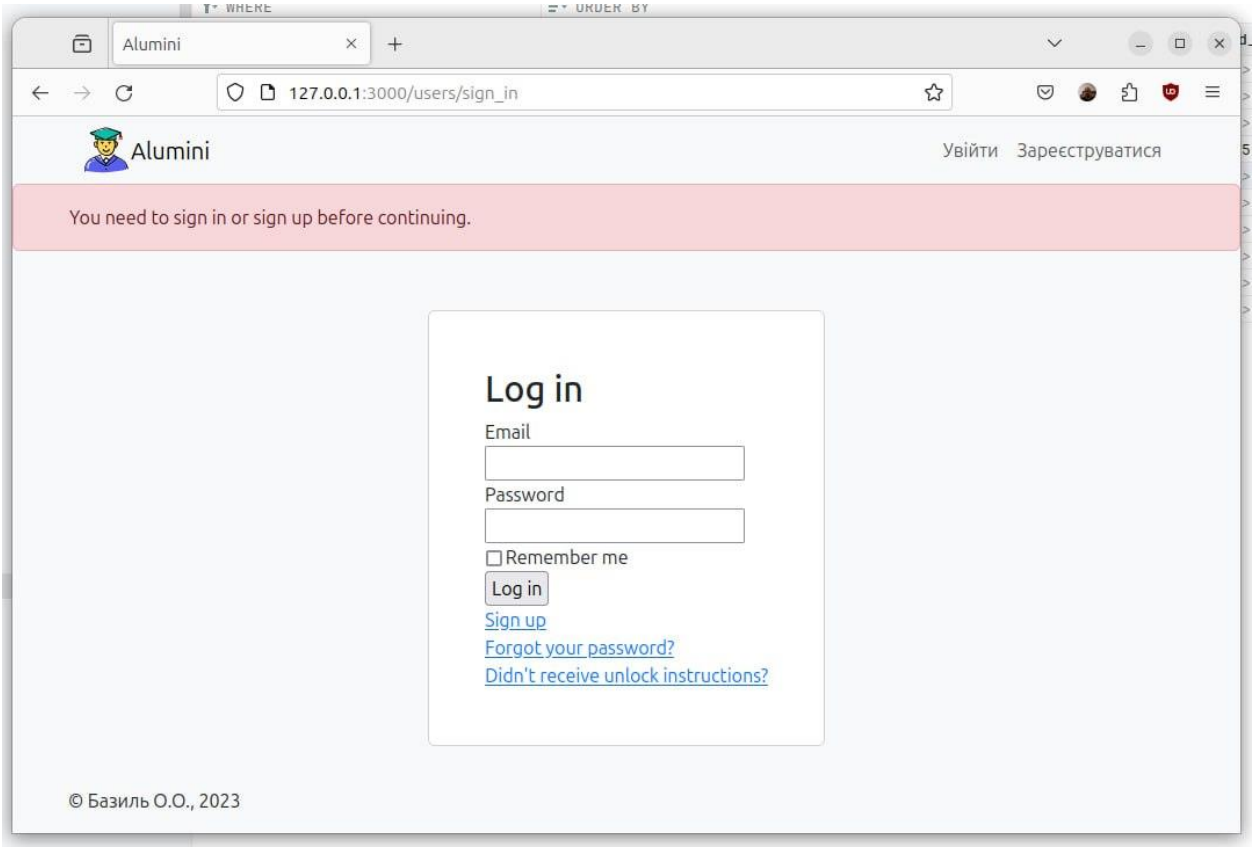

Рисунок 3.3 – Сторінка авторизації в програмний додаток

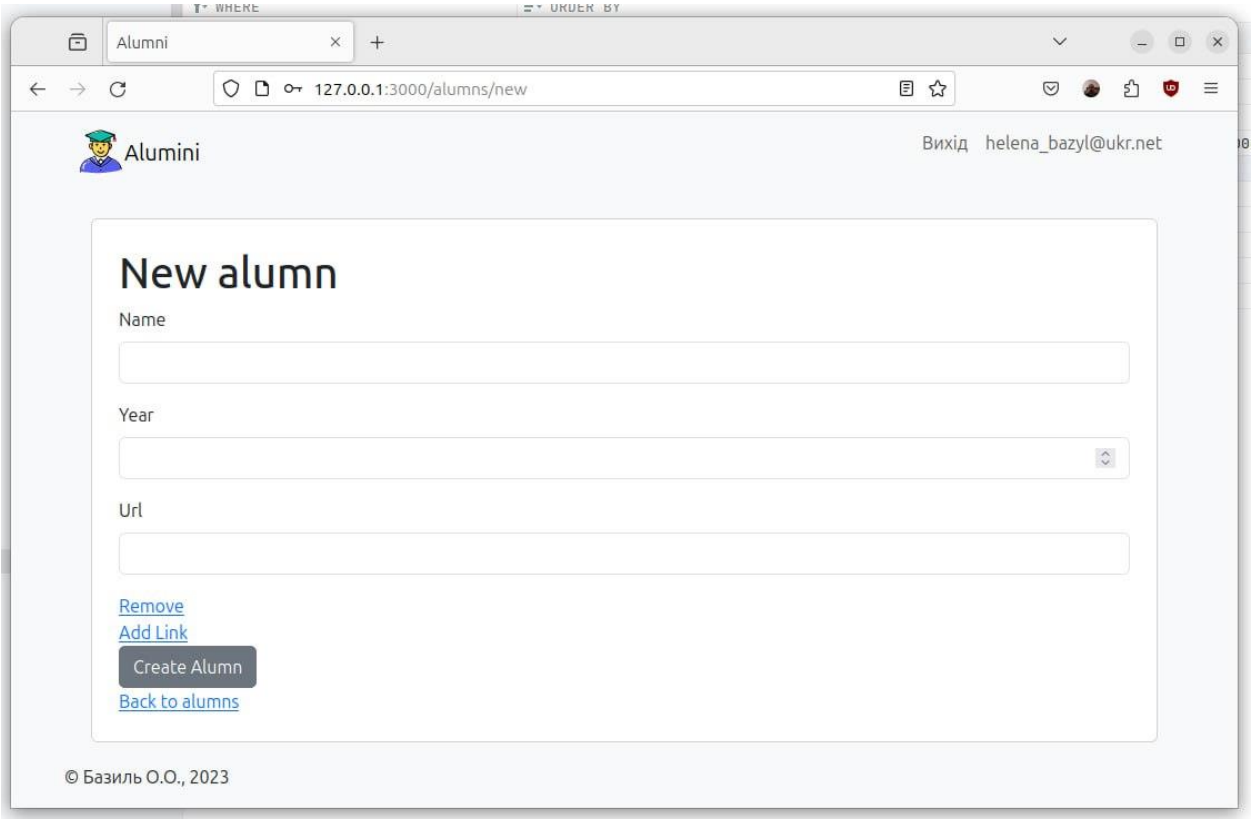

Рисунок 3.4 – Додавання інформації про нового випускника

На рисунку 3.5 представлений фрагмент бази даних випускників кафедри комп'ютерних наук Сумського державного університету. Як видно із рис. 3.5, на першій сторінці виводиться інформація про прізвище, ім'я, по-батькові випускника, рік закінчення навчання в ЗВО, остання посада, яка витягнена із LinkedLn, останнє оновлення інформації. Для отримання розгорнутої детальної інформації про випускника необхідно натиснути на кнопку «Show» навпроти його прізвища, ім'я, по-батькові.

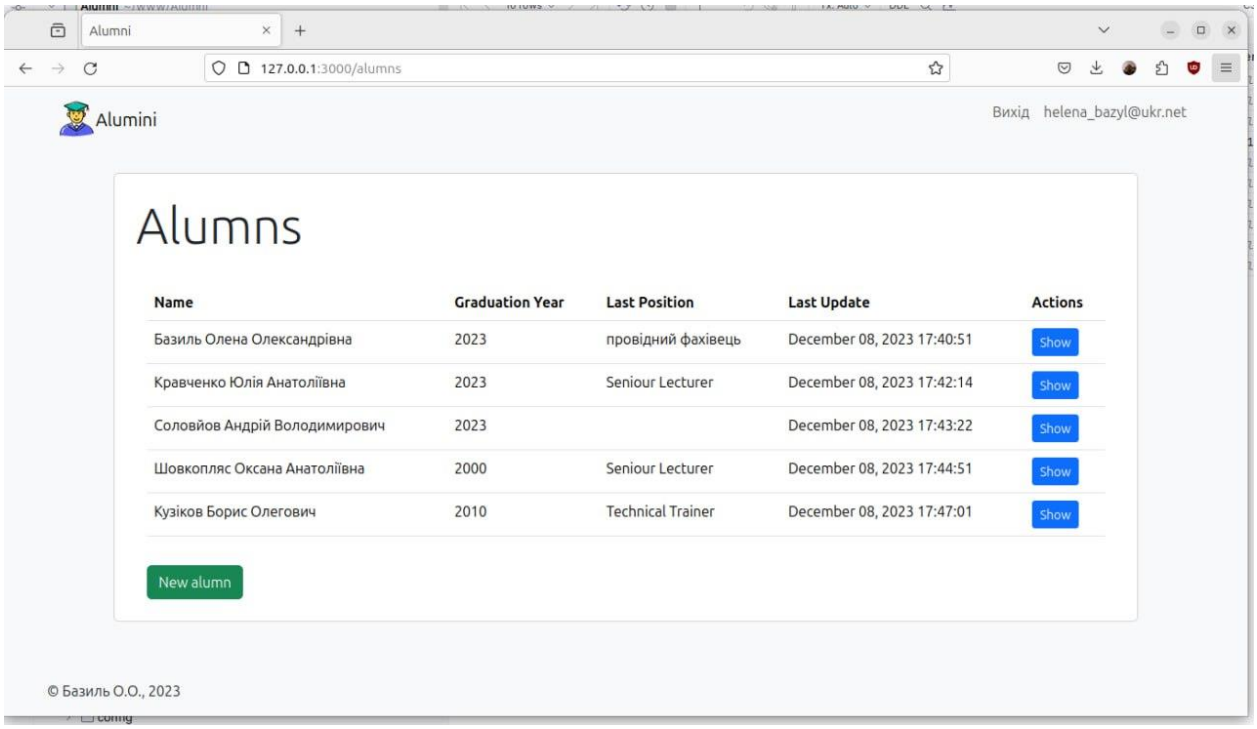

Рисунок 3.5 – Фрагмент бази даних випускників кафедри КН СумДУ

Для отримання інформації про випусника Сумського державного університету 2010 року Кузікова Бориса Олеговича натискаємо на кнопку «Show» навпроти його прізвища. Отримана інформація про особу, яка нас цікавить, представлена на рис. 3.6 – 3.9.

В додатку Б наведений лістинг програми, який описує звернення до відкритих джерел із запитом щодо потрібного користувача.

Рисунок 3.6 подає інформацію про Кузікова Бориса Олеговича, а саме рік закінчення ним навчання в СумДУ, посилання на соціальні мережі, які

містять інформацію про користувача.

https://orcid.org/0000-0002-9511-5665 - сторінка на ORCSD

https://github.com/potapuff/ - сторінка на Github

https://www.linkedin.com/in/borys-kuzikov-002172105/ - акаунт користувача на LinkedIn

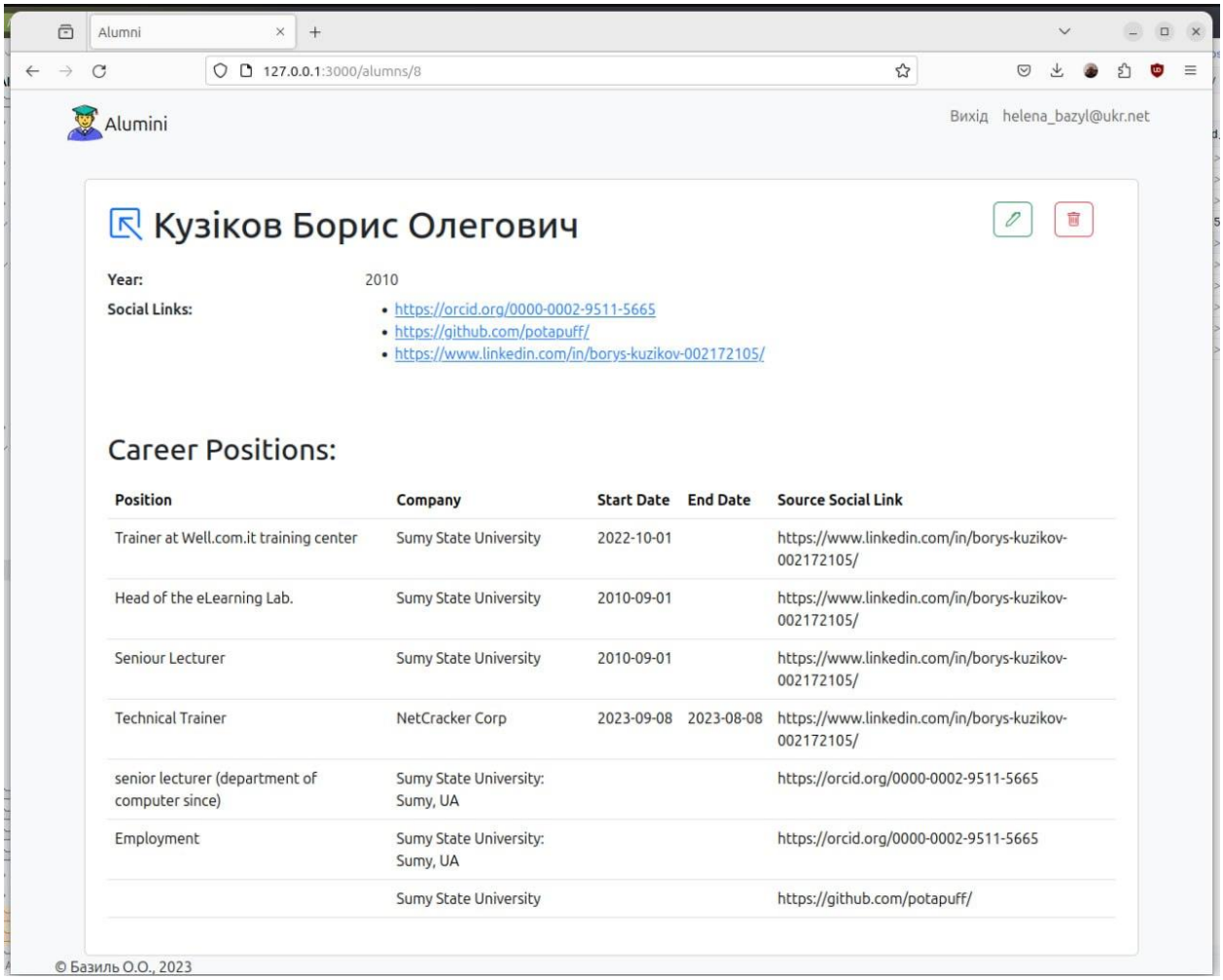

Рисунок 3.6 – Інформація про посади та місця роботи Кузікова Бориса Олеговича

Сторінка містить дані про посаду, місце роботи, дату початку та завершення роботи на вказаній посаді, посилання на сторінки на соціальних мережах, звідки ця інформація була отримана.

Рисунок 3.7 ілюструє наукову сторону діяльності Кузікова Бориса Олеговича, оскільки наукова діяльність є визначною в роботі НПП [31]. Ми бачимо його Orcid. Як видно з рис. 3.7, сторінка користувача пов’язана з

ORCID аккаунтом Сумського державного університету. Сторінка містить посилання на Scopus Author ID (55653809800) та науковий профіль (2814254). Також ми бачимо 40 робіт автора, які індексуються в базі даних Scopus.

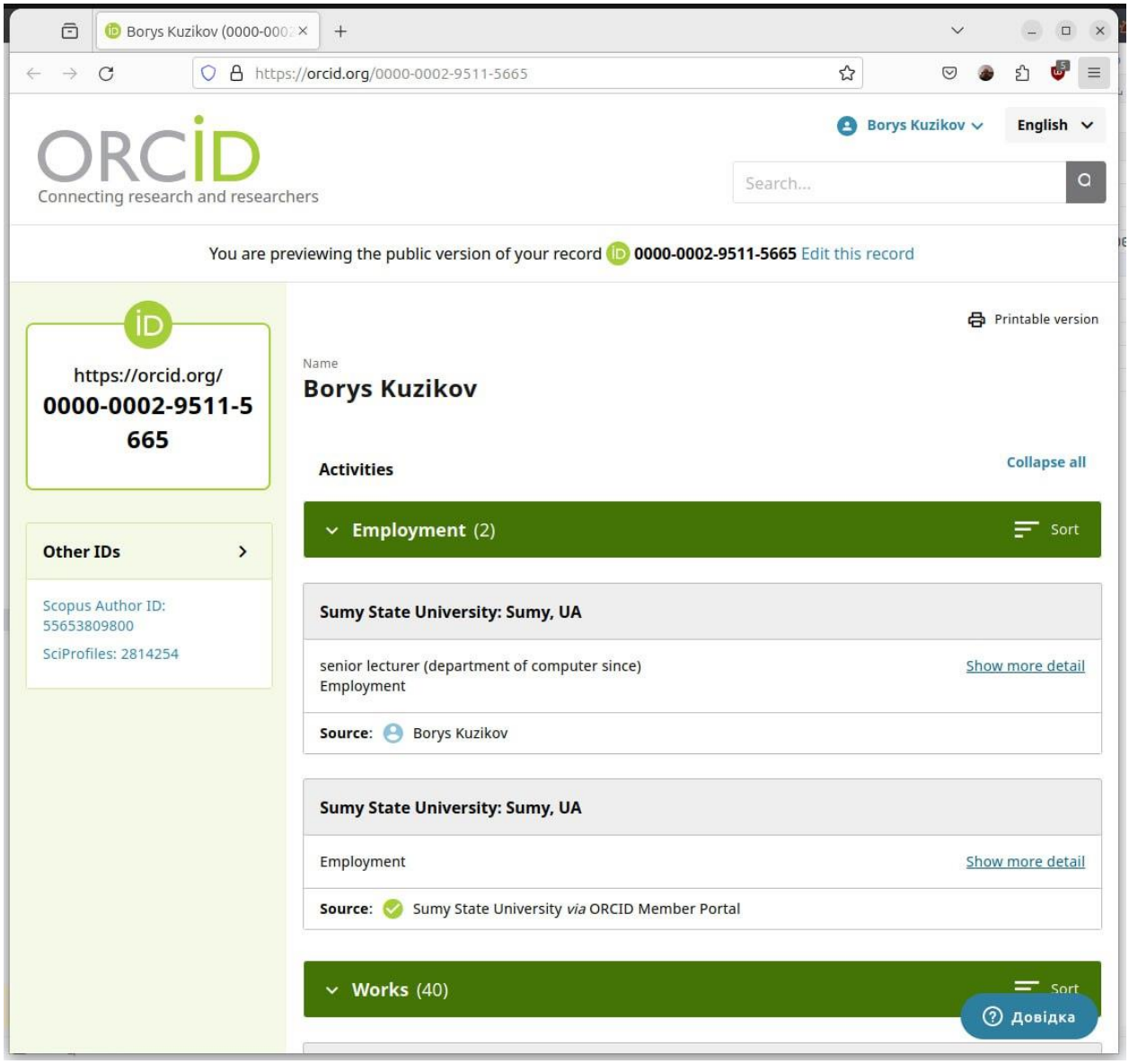

Рисунок 3.7 – Профіль Кузікова Бориса Олеговича на платформі ORCID

Рисунок 3.8 ілюструє інформацію про проєкти автора як програміста та його репозіторіс. Ми бачимо активність Бориса Олеговича на Github за останній рік. 63 активності виявлено в 2023 році, значна кількість яких припадає на грудень цього року.

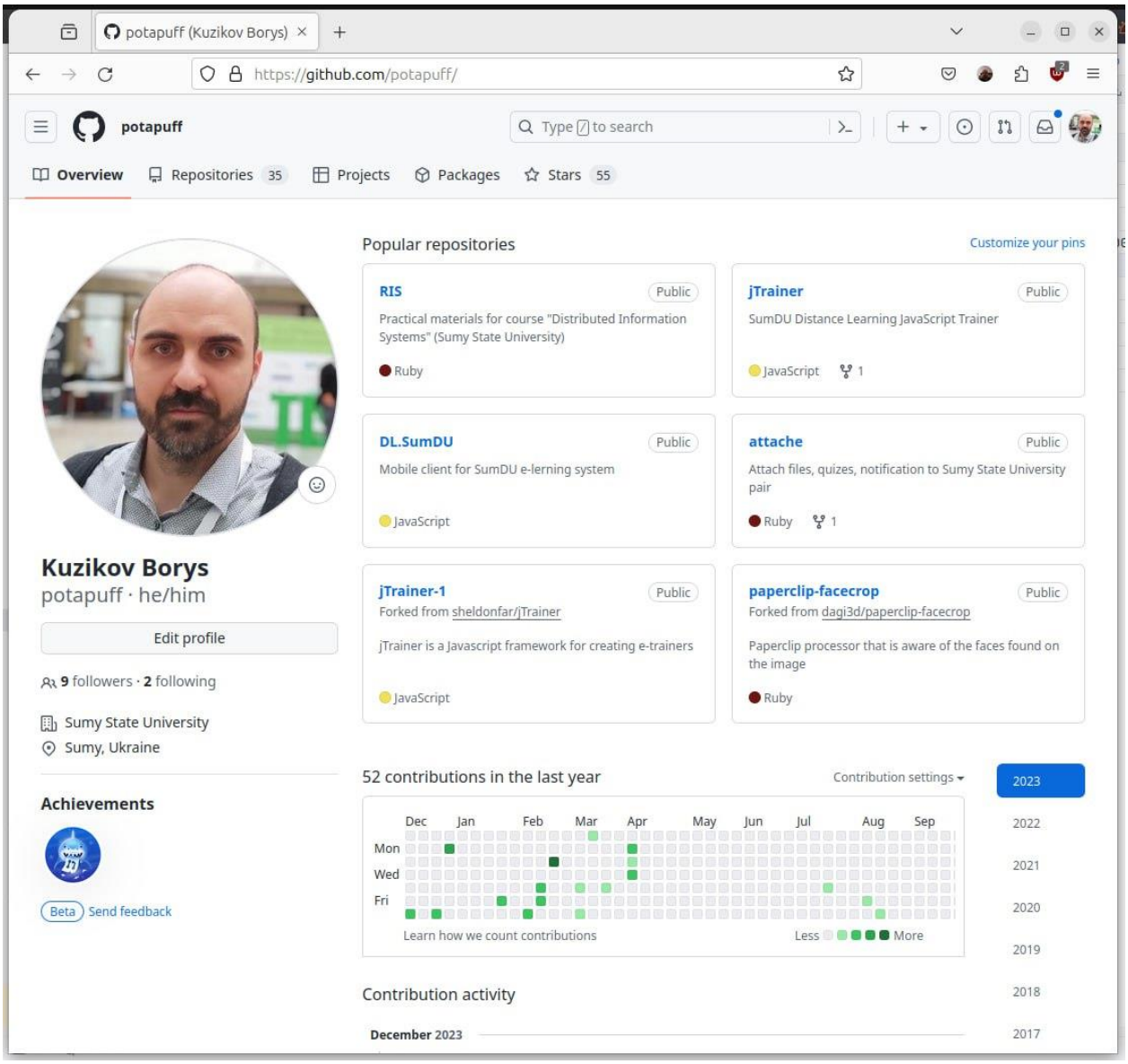

Рисунок 3.8 - Профіль Кузікова Бориса Олеговича на платформі Github

Рисунок 3.9 подає інформацію, яка витягнута програмним додатком із мережі LinkedIn. Ми бачимо, що значна частина професійної діяльності Кузікова Бориса Олеговича пов’язана із Сумським державним університетом. Ми бачимо, що з вересня 2010 року і по теперішній він працює старшим викладачем. Діяльність Бориса Олеговича як науково-педагогічного працівника пов’язана з викладанням дисциплін:

- Бази даних та інформаційні системи
- Автоматизація тестування програмного забезпечення
- Захищені інформаційні системи та бази даних

- Забезпечення якості програмних продуктів тощо.

Також тут відображена його діяльність з неповною зайнятістю в Netcracker, elearning Lab, Well.com.

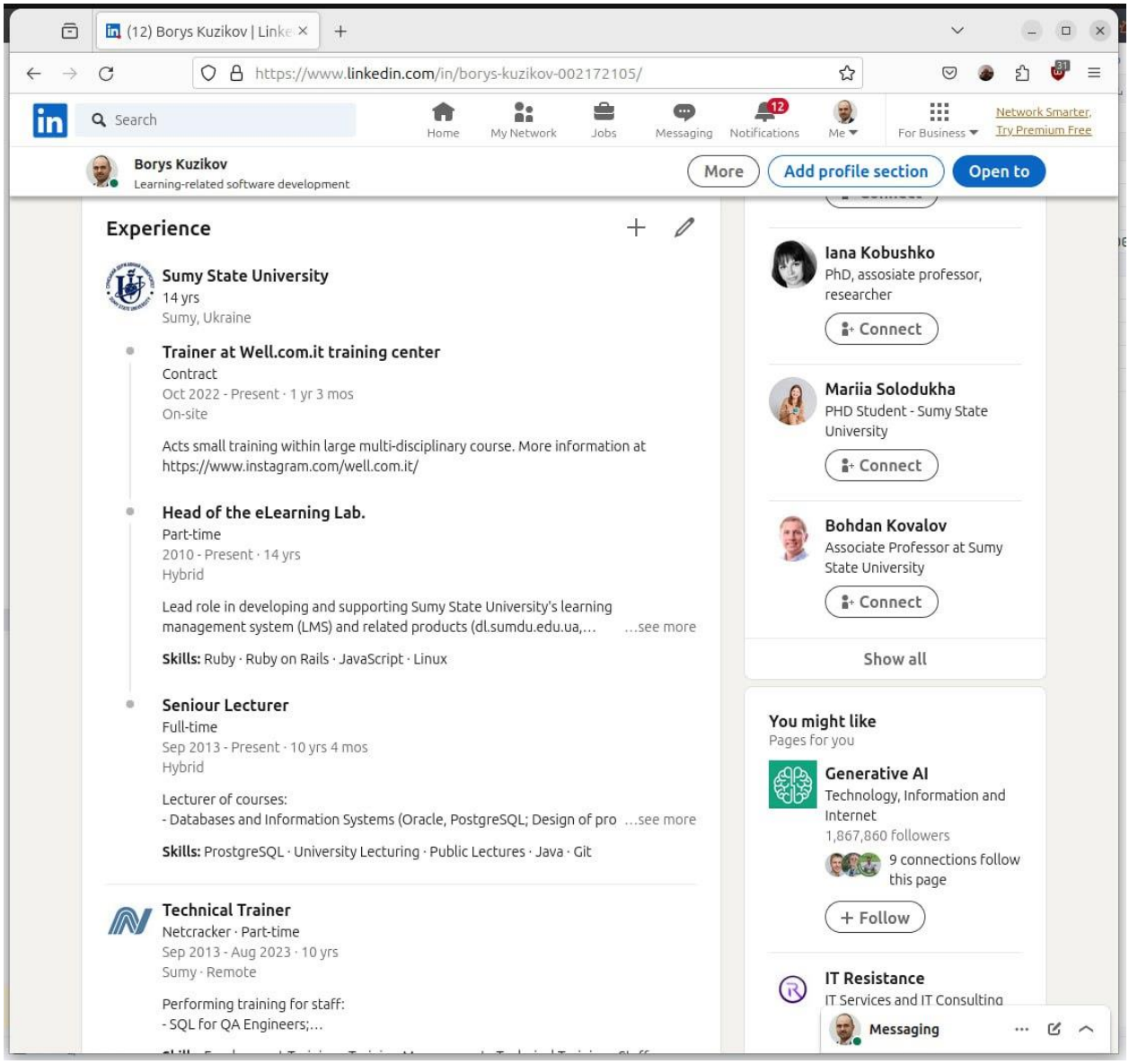

Рисунок 3.9 Профіль Кузікова Бориса Олеговича на платформі LinkedIn

Отже, створена база даних дозволяє з різних боків оглянути професійну, наукову та практичну діяльність випускника.

ВИСНОВКИ

У ході виконання дослідження за темою «Інформаційна технологія аналізу даних відкритих Інтернет-джерел», нами було встановлено, що сучасному університету потрібна інформація про його випускників з метою об’єднання зусиль колишніх здобувачів різних поколінь для розвитку ЗВО, збереження та примноження його звичаїв та моральних цінностей, піднесення рейтингу вишу як в Україні, так і за кордоном.

У ході виконання кваліфікаційної роботи було виконано такі завдання:

1. проаналізована предметна область;

2. обрана технологія та середовища програмування проєкту;

3. створена бази даних випускників кафедри КН Сумського державного університету на основі отриманих відкритих даних.

При створенні програмного продукту були використані мова програмування Ruby 3.2.2, фреймворк Ruby on Rails 7.1.1, сховище даних Redis.

При створенні бази даних випускників робилися запити до мереж Linkedln, Orcid, GitHub, отримувалися звідти відкриті дані, перевірялися, аналізувалися і заносилися до бази даних.

У подальшому планується вдосконалювати створену базу, розширюючи перелік ресурсів, з яких можна отримувати інформацію.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Catone M.C. The role of open data in digital society: The analysis of scientific trending topics through a bibliometric approach // Front. Sociol. Frontiers Media S.A., 2023. Vol. 8. P. 1134518.
2. Закон України “Про доступ до публічної інформації” {Із змінами, внесеними № 2849-IX від 13.12.2022} [Electronic resource]. URL: https://zakon.rada.gov.ua/laws/show/2939-17#Text (accessed: 17.12.2023).
3. Дорогань О., Лавриненко І., Кобець Р. Зелена книга “Політика відкритих даних” [Electronic resource]. 2023. URL: https://regulation.gov.ua/book/145-zelena-kniga-politika-vidkritih-danih (accessed: 17.12.2023).
4. Про затвердження Положення про набори даних, які підлягають оприлюдненню у формі відкритих даних | LIGA:ZAKON [Electronic resource]. 2023. URL: https://ips.ligazakon.net/document/view/KP150835?ed=2020_10_09 (accessed: 10.12.2023).
5. Yang R. et al. A new class of metrics for learning on real-valued and structured data // Data Min. Knowl. Discov. Springer New York LLC, 2019.
6. Дорошенко А.Ю. Розробка інформаційної технології інтелектуального аналізу фактографічної інформації // Біоніка інтелекту. 2018. Vol. 1, № 90. С. 116–121.
7. Langebein J. et al. A Data Mining Approach for Detecting Collusion in Unproctored Online Exams. 2023. P. 6–16.
8. Shimmei M., Matsuda N. Can’t Inflate Data? Let the Models Unite and Vote: Data-agnostic Method to Avoid Overfit with Small Data // Proceedings of the 16th International Conference on Educational Data Mining / ed. Feng M., Kaser T., Talukdar P. 2023. P. 286–295.
9. Chang D. et al. OCPMDM 2.0: An intelligent solution for materials data mining // Chemom. Intell. Lab. Syst. Elsevier B.V., 2023. Vol. 243. P. 234–242.
10. Benson M. DATA MINING : concepts and algorithms. 1st ed. MURPHY &

MOORE PUB, 2022. 245 p.

11. Faradzhullaev R. Analysis of web server log files and attack detection // Autom. Control Comput. Sci. 2018. Vol. 42, № 1. P. 50–54.
12. Лог — що це таке і як з ними працювати | HOSTiQ [Electronic resource]. 2023. URL: https://hostiq.ua/wiki/ukr/log/ (accessed: 16.12.2023).
13. Web Server logs | Server logs [Electronic resource]. URL: https://thirdeyedata.ai/web-server-logs/ (accessed: 12.12.2023).
14. Al-Zoubi A.M. et al. Evolving Support Vector Machines using Whale Optimization Algorithm for spam profiles detection on online social networks in different lingual contexts // Knowledge-Based Syst. Elsevier B.V., 2018. Vol. 153. P. 91–104.
15. Parisi G.I. et al. Continual Lifelong Learning with Neural Networks: A Review. 2018.
16. Hitam N.A., Ismail A.R., Saeed F. An Optimized Support Vector Machine (SVM) based on Particle Swarm Optimization (PSO) for Cryptocurrency Forecasting // Procedia Comput. Sci. Elsevier B.V., 2019. Vol. 163. P. 427–433.
17. Лосєв М.Ю., Федько В.В. Бази даних. Харків: ХНЕУ ім. С. Кузнеця, 2019. 232 с.
18. Han J. et al. Design and Implementation of Enabling SQL–Query Processing for Ethereum-Based Blockchain Systems // Electron. Multidisciplinary Digital Publishing Institute (MDPI), 2023. Vol. 12, № 20.
19. Daraghmi E. et al. Forensic Operations for Recognizing SQLite Content (FORC): An Automated Forensic Tool for Efficient SQLite Evidence Extraction on Android Devices // Appl. Sci. Multidisciplinary Digital Publishing Institute (MDPI), 2023. Vol. 13, № 19.
20. Syaifudin Y.W. et al. Implementation and Evaluation of Self-learning Topic for SQLite Integration in Flutter Programming Learning Assistance System. Institute of Electrical and Electronics Engineers (IEEE), 2023. P. 589–594.
21. Wu Y., Luo H., Liu Y. SQLite embedded database in data chain devices // Proceedings of SPIE - The International Society for Optical Engineering.

SPIE-Intl Soc Optical Eng, 2023. P. 77.

22. Доценко С.І. Організація та системи керування базами даних. Український державний університет залізничного транспорту, 2023. 117 с.
23. Sergievskiy M. V. Metaclasses in UML and in Programming Languages // Program. Comput. Softw. Pleiades Publishing, 2023. Vol. 49, № 5. P. 464–469.
24. Chen C.Y., Tai K.Y. Online ontological quality assessment of converted UML class diagrams in SRE // Autom. Softw. Eng. Springer, 2023. Vol. 30, № 2.
25. Qiu S. Research on Intelligent Computer Information Service Terminal System Under Deep Learning Framework // 2023 IEEE Int. Conf. Image Process. Comput. Appl. ICIPCA 2023. Institute of Electrical and Electronics Engineers Inc., 2023. P. 597–602.
26. Nor R.M. et al. Cloudemy: Step into the Cloud // J. Telecommun. Electron. Comput. Eng. 2018. Vol. 9 No. 3-5. P. 135–139.
27. Jain J. Learn API testing : norms, practices, and guidelines for building effective test automation. 2023. 223 p.
28. Friesen J. Java XML and JSON: Document Processing for Java SE. 2019. 538 p.
29. How To Set Up User Authentication with Devise in a Rails 7 Application | DigitalOcean [Electronic resource]. URL: https://www.digitalocean.com/community/tutorials/how-to-set-up-user-authentication-with-devise-in-a-rails-7-application (accessed: 19.12.2023).
30. Vanden Broucke S., Baesens B. Practical Web Scraping for Data Science // Pract. Web Scraping Data Sci. Apress, 2018.
31. Смірнова В.А. Дослідження відкритих цифрових інформаційних систем для аналізу результатів дослідницької діяльності науково-педагогічних працівників закладів вищої освіти // Електронне наукове фахове видання “Відкрите освітнє Е-середовище сучасного університету.” Borys Grinchenko Kyiv University, 2020. № 9. P. 134–144.

ДОДАТОК А

```
# frozen_string_literal: true

require 'nokogiri'
require 'watir'
require 'selenium-webdriver'

class OrcidScraper < LinkScraper
  def initialize(url)
    @url = url
  end

  def self.match_url?(url)
    url.match?('orcid.org')
  end

  def run_per_day
    100
  end

  def get_driver
    options =  {
      args: %w[--ignore-certificate-errors --disable-
popup-blocking --disable-translate --disable-
notifications --start-maximized --disable-gpu --
headless]
    }
    driver = Watir::Browser.new :chrome, options:
options
    driver.driver.manage.timeouts.implicit_wait = 100
    driver
  end

  def scrape_data
    begin
      browser = get_driver
      browser.goto @url
      html = browser.html
    rescue StandardError => ex
      #TODO ?
    ensure
      if browser
        browser.close rescue raise("Scraping failed")
```

```
      end
    end

    content = Nokogiri::HTML(html)
    data = []

    content.css('app-affiliation-stack app-panel').each
do |doc|
      pair = {}

      header = doc.css('div.header > div.ng-star-
inserted').first
      pair[:company]  = header.text.strip.gsub(/\s+/,'
')

      position = doc.css('div.body div.data-
content').first
      pair[:position] =
position.text.strip.gsub(/\s+/,'')

      data << pair
    end

    data
  end

end
```

ДОДАТОК Б

{"id":"Mng2OnBhRhdGRKF3QFa94A_0000",
"full_name":"borys
kuzikov","first_name":"borys","middle_initial":null,"mi
ddle_name":null,"last_initial":"k","last_name":"kuzikov
","gender":"male","birth_year":null,"birth_date":null,"
linkedin_url":"linkedin.com/in/borys-kuzikov-
002172105","linkedin_username":"borys-kuzikov-
002172105","linkedin_id":"445309058","facebook_url":nul
l,"facebook_username":null,"facebook_id":null,"twitter_
url":"twitter.com/potapuff","twitter_username":"potapuf
f","github_url":null,"github_username":null,"work_email
":null,"personal_emails":[],"recommended_personal_email
":null,"mobile_phone":null,"industry":"research","job_t
itle":"trainer at well.com.it training
center","job_title_role":"health","job_title_sub_role":
"fitness","job_title_levels":[],"job_company_id":"sumy-
state-university","job_company_name":"sumy state
university","job_company_website":"sumdu.edu.ua","job_c
ompany_size":"501-
1000","job_company_founded":1948,"job_company_industry"
:"research","job_company_linkedin_url":"linkedin.com/co
mpany/sumy-state-
university","job_company_linkedin_id":"1257361","job_co
mpany_facebook_url":null,"job_company_twitter_url":null
,"job_company_location_name":"sumy,
ukraine","job_company_location_locality":null,"job_comp
any_location_metro":null,"job_company_location_region":
"sumy","job_company_location_geo":null,"job_company_loc
ation_street_address":null,"job_company_location_addres
s_line_2":null,"job_company_location_postal_code":null,
"job_company_location_country":"ukraine","job_company_l
ocation_continent":"europe","job_last_updated":"2023-
11-06","job_start_date":"2022-10-
01","location_name":"ukraine","location_locality":null,
"location_metro":null,"location_region":null,"location_
country":"ukraine","location_continent":"europe","locat
ion_street_address":null,"location_address_line_2":null
,"location_postal_code":null,"location_geo":null,"locat
ion_last_updated":"2023-02-
06","phone_numbers":[],"emails":[],"interests":[],"skil
ls":["postgresql","software
development","linux","java","git","sql","ruby","ruby on
rails","javascript","databases","xml","software

engineering","css","html","analysis","research","jquery
"],"location_names":[],"regions":[],"countries":["ukrai
ne"],"street_addresses":[],"experience":[{"company":{"n
ame":"sumy state university","size":"501-
1000","id":"sumy-state-
university","founded":1948,"industry":"research","locat
ion":{"name":"sumy,
ukraine","locality":null,"region":"sumy","metro":null,"
country":"ukraine","continent":"europe","street_address
":null,"address_line_2":null,"postal_code":null,"geo":n
ull},"linkedin_url":"linkedin.com/company/sumy-state-
university","linkedin_id":"1257361","facebook_url":null
,"twitter_url":null,"website":"sumdu.edu.ua"},"location
_names":[],"end_date":null,"start_date":"2022-10-
01","title":{"name":"trainer at well.com.it training
center","role":"health","sub_role":"fitness","levels":[
]},"is_primary":true},{"company":{"name":"netcracker
technology","size":"10001+","id":"netcrackertech","foun
ded":1993,"industry":"telecommunications","location":{"
name":"waltham, massachusetts, united
states","locality":"waltham","region":"massachusetts","
metro":"boston, massachusetts","country":"united
states","continent":"north
america","street_address":"95 sawyer
road","address_line_2":"suite
600","postal_code":"02453","geo":"42.37,-
71.23"},"linkedin_url":"linkedin.com/company/netcracker
tech","linkedin_id":"165155","facebook_url":"facebook.c
om/netcrackertech","twitter_url":"twitter.com/netcracke
rtech","website":"netcracker.com"},"location_names":[],
"end_date":"2023-08-01","start_date":"2013-09-
01","title":{"name":"technical
trainer","role":"health","sub_role":"fitness","levels":
[]},"is_primary":false}],"education":[{"school":{"name"
:"sumy state university","type":"post-secondary
institution","id":"1uKnAlFtcqiFMwDtWrkQbg_0","location"
:{"name":"ukraine","country":"ukraine","locality":null,
"continent":"europe","region":null},"linkedin_url":"lin
kedin.com/school/sumy-state-
university","facebook_url":"facebook.com/sumdu.ua","twi
tter_url":"twitter.com/sumdu_ua","linkedin_id":"17665",
"website":"sumdu.edu.ua","domain":"sumdu.edu.ua"},"degr
ees":["doctorates","doctor of
philosophy"],"start_date":"2013","end_date":"2014","maj
ors":["philosophy"],"minors":[],"gpa":null}],"profiles"

:[{"network":"linkedin","id":"445309058","url":"linkedin.com/in/borys-kuzikov-002172105","username":"borys-kuzikov-002172105"},{"network":"twitter","id":null,"url":"twitter.com/potapuff","username":"potapuff"}],"version_status":{"status":"","contains":[],"previous_version":"","current_version":""}}