

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Західноукраїнський національний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра інформаційно-обчислювальних систем і управління

**ВІВЧАР Каріна Сергіївна**

**Веб-орієнтована інформаційна система наукометричного аналізу та оцінки наукових здобутків / Web-based information system for scientometric analysis and evaluation of scientific achievements**

Спеціальність 122 – Комп'ютерні науки  
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконала студентка групи КН-41  
К.С. Вівчар

Науковий керівник:  
к.т.н., доцент І. В. Турченко

Кваліфікаційну роботу допущено до захисту  
«\_\_»\_\_\_\_\_ 2025 р.

В.о. завідувача кафедри  
\_\_\_\_\_ Н. М. Васильків

**Тернопіль – 2025**

**Факультет комп'ютерних інформаційних технологій**  
**Кафедра інформаційно-обчислювальних систем і управління**  
**Освітній ступінь «бакалавр»**  
**спеціальність 122 – Комп'ютерні науки**  
**освітньо-професійна програма – Комп'ютерні науки**

**ЗАТВЕРДЖУЮ**  
В.о. завідувача кафедри  
\_\_\_\_\_ Н.М. Васильків  
«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**  
**ВІВЧАР Каріна Сергіївна**  
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Веб-орієнтована інформаційна система наукометричного аналізу та оцінки наукових здобутків / Web-based information system for scientometric analysis and evaluation of scientific achievements.

керівник роботи І.В. Турченко к.т.н., доцент

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

затверджені наказом по університету від 20 грудня 2024 р. № 938.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- проаналізувати предметну область;
- проаналізувати існуючі рішення;
- визначити переваги і недоліки існуючих рішень та постановка задачі дослідження;
- розробити структуру інформаційної системи, що дозволить почати розробку системи;
- розробити алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи;
- розробити інтерфейс інформаційної системи;
- реалізувати та протестувати веб-орієнтовану інформаційну систему.

5. Перелік графічного матеріалу в роботі:

- клієнт-серверна архітектура;
- загальна схема інформаційної системи;
- загальна структура інформаційної системи та напрямок потоку даних;
- схема алгоритму пошуку авторів в інформаційній системі;

## 6. Консультанти розділів кваліфікаційної роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|-------------------------------------------|----------------|------------------|
|        |                                           | Завдання видав | Завдання прийняв |
|        |                                           |                |                  |
|        |                                           |                |                  |
|        |                                           |                |                  |

## 7. Дата видачі завдання 20 грудня 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи                                                                                      | Строк виконання етапів кваліфікаційної роботи | Примітка |
|-------|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|----------|
| 1     | Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи               | до 01.01. 2025 р.                             |          |
| 2     | Написання 1 розділу кваліфікаційної роботи                                                                               | до 01.02. 2025 р.                             |          |
| 3     | Написання 2 розділу кваліфікаційної роботи                                                                               | до 01.04.2025 р.                              |          |
| 4     | Написання 3 розділу кваліфікаційної роботи                                                                               | до 01.05. 2025 р.                             |          |
| 5     | Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником                        | до 15.05.2025 р.                              |          |
| 6     | Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів | до 25.05.2025 р.                              |          |
| 7     | Перевірка кваліфікаційної роботи на оригінальність тексту                                                                | до 30.05.2025 р.                              |          |
| 8     | Оформлення кваліфікаційної роботи та отримання допуску до захисту                                                        | до 10.06.2025 р.                              |          |
| 9     | Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії                                              | до 10.06. 2025 р.                             |          |

Студент \_\_\_\_\_ К.С. Вівчар  
( підпис ) (прізвище та ініціали)

Керівник кваліфікаційної роботи \_\_\_\_\_ І.В. Турченко  
( підпис ) (прізвище та ініціали)

## АНОТАЦІЯ

Кваліфікаційна робота на тему «Веб-орієнтована інформаційна система наукометричного аналізу та оцінки наукових здобутків» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 51 сторінок і містить 17 ілюстрації, 2 таблиці, 3 додатки та 32 використаних джерел.

Метою кваліфікаційної роботи є створення веб-орієнтованої інформаційної системи наукометричного аналізу та оцінки наукових здобутків.

Об'єктом дослідження є технології опрацювання наукометричних даних авторів.

Предмет дослідження – веб-орієнтована інформаційна система наукометричного аналізу та наукових здобутків.

Розроблено веб-орієнтовану інформаційну систему наукометричного аналізу та оцінки наукових здобутків, яка дозволяє отримувати, аналізувати та створювати звіти в pdf форматі про авторів.

Ключові слова: ВЕБ-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА, НАУКОМЕТРИЧНІ ДАНІ, НАУКОВІ ЗДОБУТКИ, ЗВІТ.

## ANNOTATION

The qualification work on the topic "Web-oriented information system for scientometric analysis and evaluation of scientific achievements" for obtaining the educational degree "bachelor" in specialty 122 "Computer Science" of the educational program "Computer Science" is written in a volume of 51 pages and contains 17 illustrations, 2 tables, 3 appendices and 32 used sources.

The purpose of the qualification work is to create a web-oriented information system for scientometric analysis and evaluation of scientific achievements.

The object of the study is the technology of processing authors' scientometric data.

The subject of the study is a web-oriented information system for scientometric analysis and scientific achievements.

A web-oriented information system for scientometric analysis and evaluation of scientific achievements has been developed, which allows you to receive, analyze and create reports in pdf format about the authors.

Keywords: WEB-ORIENTED INFORMATION SYSTEM, SCIENTIFIC METRIC DATA, SCIENTIFIC ACHIEVEMENTS, REPORT.

## ЗМІСТ

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| Вступ.....                                                                                    | 7  |
| 1 Стан та аналіз предметної області.....                                                      | 10 |
| 1.1 Опис предметної області та проблеми наукометричного аналізу .....                         | 10 |
| 1.2 Огляд і аналіз існуючих рішень.....                                                       | 13 |
| 1.3 Постановка задачі.....                                                                    | 16 |
| 2 Загальна структура та інформаційне забезпечення веб-орієнтованої інформаційної системи..... | 20 |
| 2.1 Загальна структура веб-орієнтованої інформаційної системи.....                            | 20 |
| 2.2 Алгоритмічне забезпечення системи .....                                                   | 25 |
| 2.3 Інформаційне забезпечення .....                                                           | 28 |
| 3 Реалізація та тестування веб-орієнтованої інформаційної системи .....                       | 31 |
| 3.1 Програмна реалізація системи.....                                                         | 31 |
| 3.2 Інтерфейс користувача .....                                                               | 36 |
| 3.3 Тестування веб-орієнтованої інформаційної системи.....                                    | 40 |
| Висновки.....                                                                                 | 44 |
| Список використаних джерел.....                                                               | 45 |
| Додаток А Код HTML шаблону сторінки з інформацією про автора .....                            | 48 |
| Додаток Б Код HTML шаблону для генерації звіту в форматі PDF .....                            | 49 |
| Додаток В Копія опублікованих результатів.....                                                | 50 |

## ВСТУП

У сучасному науковому світі, де знання та інновації відіграють ключову роль у розвитку суспільства, виникає потреба в ефективних методах оцінки наукових досягнень. Наукометричний аналіз є важливою частиною цього процесу, оскільки дозволяє кількісно оцінювати результативність наукових досліджень, вивчати динаміку розвитку науки, а також допомагає у виборі напрямків для подальших інвестицій у наукові дослідження.

Традиційні методи оцінки наукових здобутків часто ґрунтуються на ручному зборі даних і аналізі, що є трудомістким і часозатратним процесом. У зв'язку з ростом обсягів наукових публікацій і необхідністю оперативного доступу до інформації, традиційні способи оцінки стають менш ефективними. Тому, на зміну таким методам, приходять автоматизовані інформаційні системи, які здатні проводити наукометричний аналіз в режимі реального часу.

Веб-орієнтовані інформаційні системи для наукометричного аналізу дозволяють науковцям, дослідникам та організаціям здійснювати швидкий доступ до даних, таких як кількість публікацій, цитованість, індекс Хірша, наукові праці, конференції та інші важливі показники. Такі системи можуть автоматично збирати дані з наукометричних баз (наприклад, Scopus [1], Web of Science [2], Google Scholar [3]), обробляти та надавати інтуїтивно зрозумілі візуалізації для аналізу наукової діяльності.

Актуальність дослідження в цій сфері зростає через збільшення кількості наукових публікацій і зростання конкуренції серед науковців. Оцінка наукових здобутків не лише для академічних установ, але й для державних органів та приватного сектору стає важливою частиною стратегії підтримки наукових досліджень і розвитку інновацій. Це дозволяє ефективно розподіляти ресурси для підтримки перспективних напрямків науки, а також допомагає зібрати точну інформацію про наукову продуктивність науковців і установ.

На сьогодні існує кілька підходів до автоматизації наукометричного аналізу. Одним із найбільш поширених є використання таких метрик, як цитованість та індекс Хірша, які допомагають оцінити наукові досягнення. Проте, ці підходи мають деякі обмеження, адже не завжди враховують інші фактори, такі як якість публікацій або вплив на практичне застосування знань.

Сучасні веб-орієнтовані інформаційні системи вже включають механізми автоматичного збору даних з різних наукометричних баз, але більшість з них потребують значних обчислювальних ресурсів, мають обмеження по точності в залежності від специфіки бази даних і не завжди здатні адаптуватися до нових форматів даних. Важливим викликом є також необхідність обробки великих масивів даних та інтеграції різних джерел інформації для створення єдиної платформи.

Останніми роками науковці активно працюють над вдосконаленням існуючих методів наукометричного аналізу, застосовуючи технології машинного навчання та штучного інтелекту. Завдяки цим технологіям можна автоматизувати аналіз та прогнози, що дозволить виявляти нові тенденції у науці та точніше оцінювати вплив наукових робіт на розвиток певної галузі.

Один з важливих аспектів – це необхідність створення інтерфейсів для користувачів, що дозволяють легко та швидко доступити та інтерпретувати наукометричні дані. Зручні візуалізації (графіки, діаграми, теплові карти) на базі таких даних значно полегшують процес оцінки і дозволяють проводити глибший аналіз.

Отже, розробка таких систем є важливим кроком у забезпеченні сучасних підходів до оцінки наукових результатів, що дозволяє не лише покращити точність аналізу, а й значно знизити витрати часу та ресурсів, пов'язаних з традиційними методами оцінки. Тому тема кваліфікаційної роботи є актуальною.

Метою роботи є створення веб-орієнтованої інформаційної системи наукометричного аналізу та оцінки наукових здобутків.

Для реалізації мети кваліфікаційної роботи необхідно виконати наступні завдання:

- провести аналіз предметної області;
- проаналізувати існуючі рішення;
- розробити структуру веб-орієнтованої інформаційної системи;
- розробити алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи;
- розробити інтерфейс інформаційної системи;
- реалізувати та протестувати веб-орієнтовану інформаційну систему.

Об'єктом дослідження є наукометричні дані авторів.

Предметом дослідження є веб-орієнтована інформаційна система наукометричного аналізу та наукових здобутків.

Розроблений продукт може служити корисним для науковців та дослідників для моніторингу власної наукової активності та планування подальших досліджень, керівництва наукових установ для оцінки ефективності роботи підрозділів та прийняття управлінських рішень, освітніх установ для підготовки звітів та проходження акредитаційних процедур, державних органів для формування наукової політики та розподілу фінансування, грантодавців та інвесторів для оцінки потенціалу наукових проектів та прийняття рішень щодо фінансування.

Результати опубліковано в матеріалах 4 міжнародної науково-практичної конференції «Global Directions in Scientific Research and Technological Development», м.Цюріх, 09-11.06.2025 р. (додаток В).

# 1 СТАН ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис предметної області та проблеми наукометричного аналізу

У контексті стрімкого зростання обсягів наукової інформації особливої актуальності набуває питання автоматизації процесів наукометричного аналізу. Щороку публікуються сотні тисяч нових наукових статей, препринтів, технічних звітів, що робить ручне опрацювання та оцінювання практично неможливим. Обсяги інформації подвоюються з дедалі більшою частотою, охоплюючи різноманітні галузі знань, мови та типи публікацій. Традиційні методи збору й аналізу даних більше не здатні забезпечити оперативність і точність, необхідні для прийняття обґрунтованих управлінських рішень у науковій сфері. Автоматизація дозволяє істотно скоротити час на обробку великого масиву інформації, зменшити кількість помилок, пов'язаних із людським фактором, та забезпечити систематичний моніторинг наукової активності в реальному часі. Таким чином, впровадження автоматизованих рішень є не лише доцільним, а й необхідним етапом розвитку сучасної наукометрії.

Наукометрія є спеціалізованою галуззю досліджень, що займається кількісним аналізом наукової діяльності. Її завдання полягає у вимірюванні ефективності наукових досліджень за допомогою різноманітних показників, серед яких особливої уваги заслуговують кількість публікацій, рівень цитованості, індекси впливовості, такі як h-індекс, імпакт-фактор наукових журналів, а також співвідношення між локальною та міжнародною активністю авторів. У сучасних умовах інтенсифікації наукової комунікації та глобалізації дослідницьких процесів, об'єктивна та оперативна оцінка результативності наукової роботи є надзвичайно актуальною. Зокрема, вона є основою для формування стратегій розвитку наукових установ, розподілу ресурсів, ухвалення управлінських рішень, акредитації освітніх програм, присудження наукових ступенів і звань.

Основними об'єктами наукометричного аналізу є наукові публікації, окремі дослідники, наукові колективи, установи та наукові журнали. Результати аналізу дозволяють визначити тенденції розвитку окремих наукових напрямів, виявити

провідних дослідників, оцінити наукову продуктивність організацій і ефективність міжнародного співробітництва. Сучасні наукометричні системи дедалі частіше застосовуються у державному та інституційному управлінні наукою, оскільки дають змогу не лише фіксувати досягнення, але й виявляти вузькі місця, відстежувати динаміку змін та прогнозувати подальший розвиток.

Для реалізації наукометричного аналізу використовуються різні інформаційні системи, з-поміж яких найбільш поширеними є Google Scholar та Scopus. Google Scholar є відкритим інструментом, що охоплює широкий спектр наукових джерел, включно з неопублікованими матеріалами, дисертаціями та препринтами. Перевагами цієї платформи є її доступність, інтуїтивно зрозумілий інтерфейс і широке охоплення публікацій, включаючи гуманітарні та соціальні науки, що часто недооцінюються іншими базами. Разом з тим, вона має низку обмежень, зокрема, відсутність чітких критеріїв індексації, наявність дублікатів і обмежена точність даних, що ускладнює використання цієї платформи для об'єктивного аналізу.

Scopus, у свою чергу, є комерційною базою даних, яка забезпечує більш суворий відбір публікацій та вищий рівень структурованості інформації. Вона надає розширені інструменти аналітики, зокрема зведення про співпрацю між країнами та установами, інтердисциплінарність досліджень та індекси авторських профілів. Проте обмеження доступу до повного функціоналу без відповідної ліцензії робить її використання недосяжним для частини наукової спільноти, особливо у країнах із недостатнім фінансуванням науки.

Незважаючи на значний потенціал сучасних наукометричних систем, у процесі збору та обробки даних виникає низка суттєвих проблем. Однією з ключових є фрагментація та несумісність даних, зумовлені відмінностями у форматах метаданих, варіативністю написання імен авторів, назв установ, а також наявністю дублікатів і некоректних записів. Ці фактори суттєво ускладнюють об'єднання інформації з кількох джерел і знижують точність аналітичних висновків. Актуальним стає питання стандартизації даних і створення єдиної універсальної моделі наукових метаданих.

Ще одним викликом є обмежений доступ до автоматизованого отримання інформації. Зокрема, Scopus надає API [4] лише для ліцензованих користувачів, а Google Scholar взагалі не має офіційного API. У зв'язку з цим виникає потреба у використанні методів веб-скрейпінгу [5], які, однак, порушують умови користування сервісом і можуть призводити до блокування доступу. Таким чином, розробка легальних інструментів для роботи з цими платформами є важливою умовою ефективної автоматизації.

Суттєвою проблемою є також складність однозначної ідентифікації авторів. Відсутність єдиного стандарту авторських профілів, неуніфіковане використання імен, наявність співавторства, змін у місці роботи дослідника ускладнюють формування точних профілів і можуть призводити до хибної атрибуції наукових досягнень. Попри наявність систем, таких як ORCID [6], ідентифікація авторів залишається проблемною, особливо у випадках масової обробки даних, де людський контроль обмежений.

Окрім технічних обмежень, існують концептуальні обмеження наукометрії, пов'язані з недостатністю кількісних показників для повноцінної оцінки якості наукової діяльності. Наприклад, публікації можуть мати високу кількість цитувань, але не завжди це свідчить про їхню наукову новизну чи практичну значущість. Більше того, орієнтація на формальні метрики стимулює не завжди добросовісну наукову поведінку, зокрема надмірну публікаційну активність без реального внеску в розвиток науки. Також поширеним є явище "цитувального картелю" [7] – штучне підвищення метрик за рахунок взаємного цитування.

Додатковим чинником, що ускладнює наукометричний аналіз, є затримки в оновленні даних у базах. У Google Scholar ці оновлення відбуваються нерегулярно, тоді як у Scopus, попри більш контрольований процес індексації, також можуть спостерігатися часові затримки між публікацією статті та її появою в базі. Це знижує актуальність аналітичної інформації і впливає на прийняття своєчасних рішень, особливо в умовах конкуренції за фінансування чи в рамках оцінювання ефективності.

Візуалізація результатів також залишається недостатньо розвиненою. Існуючі інструменти здебільшого орієнтовані на експертів і вимагають спеціальної підготовки. Відсутність зрозумілого інтерфейсу та інтерактивної аналітики ускладнює використання даних широким колом користувачів, зокрема адміністративним персоналом, який приймає управлінські рішення. Застосування сучасних технологій візуалізації – таких як дашборди, графі, інтерактивні мапи – здатне підвищити доступність інформації та полегшити її інтерпретацію.

У зв'язку з викладеним, актуальним є створення інтегрованої системи, що дозволить автоматизувати процес збору, обробки, уніфікації та візуалізації наукометричних даних з різних джерел. Такий підхід сприятиме підвищенню об'єктивності аналітики, забезпечить зручний інструментарій для науковців та управлінців і стане підґрунтям для подальшого розвитку систем підтримки прийняття рішень у науковій сфері. Окрім цього, використання штучного інтелекту та машинного навчання у таких системах відкриває нові горизонти для глибшого й точнішого аналізу наукової продуктивності.

## 1.2 Огляд і аналіз існуючих рішень

У сучасному науковому просторі наукометричний аналіз відіграє ключову роль у процесах оцінювання результативності наукової діяльності окремих дослідників, колективів, установ та країн загалом. Зростання обсягів наукових публікацій, інтернаціоналізація науки, розвиток цифрових технологій зумовили необхідність автоматизації процесів збору та обробки наукометричної інформації. У зв'язку з цим на ринку з'явилася низка рішень – як комерційних, так і відкритих – які дозволяють здійснювати пошук, агрегацію та аналіз бібліографічних даних, цитувань, індексів впливовості та інших метрик.

Найпоширенішими базами даних для наукометричного аналізу сьогодні залишаються Google Scholar, Scopus та Web of Science. Водночас останні дві мають суттєві обмеження доступу, що робить їх менш зручними для індивідуального

використання без інституційної підтримки. На тлі цього Google Scholar, завдяки відкритості інтерфейсу, значно виграє за рівнем доступності. Крім цього, останніми роками наукова спільнота отримала доступ до нових ініціатив, таких як OpenAlex [8] – відкритої бази даних, яка є фактичним наступником Microsoft Academic. Кожне з цих рішень має свої переваги та недоліки, а тому доцільно здійснити їх системне порівняння у контексті функціоналу, зручності, охоплення та можливостей для інтеграції.

Google Scholar є найпопулярнішим безкоштовним ресурсом, що агрегує наукові статті з різноманітних джерел репозитаріїв, видавництв, персональних сторінок, електронних архівів тощо. Хоча Google не розкриває повної інформації щодо алгоритмів індексації, відомо, що система охоплює значно більше документів, ніж її комерційні аналоги. Проте серед основних недоліків варто зазначити неоднорідну якість метаданих, відсутність жорсткої верифікації публікацій, складність у нормалізації ідентифікаторів авторів та публікацій. Інструменти автоматизованої роботи з Google Scholar як-от SerpApi [9] – дозволяють отримувати доступ до даних через API, однак потребують платної підписки. Альтернативно використовуються утиліти типу Publish or Perish [10], які здійснюють скрейпінг, але не завжди стабільно працюють через часті зміни в інтерфейсі Google.

Scopus, платформа компанії Elsevier, забезпечує системний, структурований доступ до рецензованих наукових публікацій, індексів цитування та аналітики на високому рівні. Перевагами є якісна фільтрація матеріалів, уніфіковані профілі авторів, повноцінна підтримка унікальних ідентифікаторів (наприклад, ORCID), а також потужна система візуалізації даних. Однак суттєвим бар'єром є обмежений доступ до API, який можливий лише для підписників або установ, що мають ліцензію. Для індивідуальних користувачів це створює труднощі у використанні Scopus як джерела даних у власних програмних розробках.

OpenAlex – відкритий проєкт, який був створений після закриття Microsoft Academic і базується на публічних наукометричних джерелах. Система підтримує зв'язки між авторами, публікаціями, організаціями та концептами, надає безкоштовне

API з можливістю побудови власних запитів. Попри відкритість та сучасну архітектуру, платформа поки що має обмежений рівень готових інтерфейсів для кінцевих користувачів, відсутні засоби візуалізації, а інтеграція з деякими локальними джерелами може бути ускладненою. Водночас вона ідеально підходить як бекенд для розробки власних систем аналітики.

Є також ще декілька схожих наукометричних систем, проте вони схожі на інші платформи та не вирішують проблем, які є у порівняних платформах. Тому використано для порівняння Google Scholar, Scopus та OpenAlex.

Загалом, аналіз наявних рішень дозволяє зробити висновок, що жодна з існуючих платформ не забезпечує одночасно:

об'єднання даних з кількох джерел (наприклад Google Scholar і Scopus); локалізований інтерфейс; просту доступність для викладачів без спеціальних знань; вбудовану базову аналітику та формування звітів.

Саме ці можливості планується реалізувати в межах створюваної платформи, що дозволить заповнити наявну нішу між високотехнічними рішеннями й потребами практичних користувачів, а саме: викладачів, науковців, працівників кафедр та відділів якості освіти.

Узагальнені характеристики проаналізованих систем наведені в таблиці 1.1.

Таблиця 1.1 - Порівняння існуючих рішень за критеріями.

| Критерій              | Google Scholar +<br>SerpApi/PoP | Scopus + API                       | OpenAlex    |
|-----------------------|---------------------------------|------------------------------------|-------------|
| Тип доступу           | Частково платний                | Комерційний                        | Вільний     |
| API / автоматизація   | Так (через SerpApi / PoP)       | Так (для ліцензійних користувачів) | Так         |
| Аналіз і візуалізація | Відсутній                       | Потужний (але лише на платформі)   | Мінімальний |
| Звіти для викладачів  | Немає                           | Лише в межах Scopus                | Немає       |
| Мова інтерфейсу       | Англійська                      | Англійська                         | Англійська  |

Аналіз свідчить, що жодна з платформ не є універсальною для потреб пересічного науковця або адміністративного працівника закладу вищої освіти. Google

Scholar вирізняється відкритістю та охопленням, але програє в точності та структурованості даних. Scopus забезпечує високу якість та функціональність, однак має серйозні обмеження щодо доступу. OpenAlex, як сучасна альтернатива, є технічно перспективною, проте ще немає повноцінної підтримки для нетехнічних користувачів.

Тому перспективним напрямом є створення рішення, яке б поєднало сильні сторони згаданих платформ: автоматизований збір з відкритих джерел (Google Scholar, OpenAlex), структурованість даних, наявність базових аналітичних інструментів і локалізованого інтерфейсу. Такий підхід дозволить суттєво спростити доступ до наукометричної інформації широкому колу користувачів і вирішити актуальні проблеми моніторингу наукової активності у вищій освіті.

### 1.3 Постановка задачі

У сучасному науковому середовищі зростає роль наукометрії як інструменту для оцінки ефективності дослідницької діяльності, прийняття управлінських рішень, формування стратегій розвитку наукових установ та підвищення конкурентоспроможності університетів. Водночас, ефективне застосування наукометричних методів залежить насамперед від якості, повноти й актуальності даних, які використовуються для аналізу.

Проблеми, пов'язані з фрагментованістю джерел, нестачею єдиного стандарту представлення інформації, технічними обмеженнями доступу та недостатньою інтеграцією між платформами, значно ускладнюють процес отримання достовірних і цілісних результатів.

Наразі найбільш відомими платформами для наукометричного аналізу є Google Scholar, Scopus, Web of Science, OpenAlex, кожна з яких має свої переваги та недоліки. Наприклад, Google Scholar відзначається відкритістю й охопленням широкого спектру публікацій, але не має офіційного API для автоматизованої роботи, що створює складнощі для регулярного збору даних. Scopus, навпаки, є надійною та

структурованою системою, однак потребує ліцензованого доступу й має обмеження щодо обсягу API-запитів. Крім того, між платформами спостерігається відмінність у форматах метаданих, структурах профілів, написанні імен авторів і назв установ. Це призводить до фрагментації даних, ускладнює ідентифікацію осіб та створює ризики дублювання або втрати частини інформації.

Важливість вирішення зазначених проблем зумовлена не лише технічними, а й організаційними аспектами. У випадку, коли для формування аналітичного звіту про наукову активність викладачів потрібно вручну збирати та порівнювати дані з кількох джерел, значно зростають часові витрати та ризик помилок.

Відсутність уніфікованих інструментів обробки таких даних унеможливорює оперативне реагування на зміни у науковій продуктивності або своєчасну оцінку потенціалу дослідників.

У цьому контексті очевидною є потреба у створенні інтегрованої платформи, яка б дозволила централізовано отримувати, структурувати та аналізувати наукометричну інформацію з кількох джерел, забезпечуючи її цілісність, точність та актуальність. Такий підхід дозволить подолати існуючі обмеження кожного окремого джерела, підвищити достовірність аналізу та надати користувачам більш повну картину наукової активності.

У межах даної кваліфікаційної роботи ставиться задача розробити інформаційно-аналітичну систему, яка забезпечить: автоматизований збір даних з Google Scholar (за допомогою сторонніх рішень, зокрема SerpApi або Publish or Perish) та Scopus (через API за умови наявності доступу); нормалізацію даних; уніфікацію форматів метаданих, зіставлення профілів авторів, ідентифікацію установ та видань; збереження інформації у структурованому форматі (наприклад, у вигляді бази даних або графової моделі зв'язків); візуалізацію результатів та побудову діаграм, графіків цитувань, динаміки публікаційної активності, мережових структур співавторства тощо; створення звітів у зручному для кінцевого користувача форматі (PDF [11], таблиці, інтерактивні панелі).

Особлива увага приділяється проблемі зіставлення профілів авторів, оскільки варіативність написання імен, наявність співавторства, зміни у місцях роботи та відсутність чітких ідентифікаторів ускладнюють процес ідентифікації. Незважаючи на існування ORCID, він використовується не всіма науковцями, що знижує ефективність автоматизованої ідентифікації. У цьому контексті доцільним є впровадження власних механізмів перевірки та порівняння профілів за сукупністю параметрів (співпадіння назв публікацій, афіліацій, тематики тощо).

Також передбачається реалізація гнучкого інтерфейсу, орієнтованого на різні категорії користувачів, а саме адміністрацію, науковців, працівників відділів якості освіти. Кожен з них має потребу в різних формах доступу до даних, тому платформа має підтримувати ролі користувачів, персоналізовані звіти та можливість вибору потрібного рівня деталізації.

Ще одним аспектом є розширюваність платформи – можливість у майбутньому підключення додаткових джерел даних, таких як Web of Science, OpenAlex або локальні репозитарії. Важливо, щоб система мала модульну архітектуру, що дозволить масштабувати функціонал без значних змін у вже реалізованій частині.

Створення такої платформи є актуальним і практично значущим завданням, здатним забезпечити: оптимізацію процесу підготовки звітності та акредитаційних документів; підвищення прозорості оцінювання наукової продуктивності; розвиток цифрової інфраструктури університетів; підтримку стратегічного планування в сфері науки та освіти.

Таким чином, реалізація веб-орієнтованої системи дозволяє вирішити низку як технічних, так і управлінських задач, що робить його цінним інструментом у контексті модернізації наукової та освітньої аналітики.

Загалом, створення системи, яка інтегрує інформацію з кількох наукометричних джерел, дозволяє автоматизувати її обробку та забезпечує зручний інтерфейс для перегляду й аналізу даних, є актуальним і практично значущим завданням. Такий інструмент може бути використаний як у межах окремих кафедр та факультетів, так і

на рівні університету чи наукової установи для стратегічного планування, акредитацій, підготовки звітності, моніторингу наукової активності тощо.

Отже, реалізація веб-орієнтованої системи не лише вирішує низку технічних проблем, а й сприяє загальному підвищенню ефективності управління вищою освітою та наукою.

Отже завданнями кваліфікаційної роботи є:

- провести аналіз предметної області;
- проаналізувати існуючі рішення;
- розробити структуру веб-орієнтованої інформаційної системи;
- розробити алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи;
- розробити інтерфейс інформаційної системи;
- реалізувати та протестувати веб-орієнтовану інформаційну систему.

## 2 ЗАГАЛЬНА СТРУКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Загальна структура веб-орієнтованої інформаційної системи

У процесі створення інформаційної системи для наукометричного аналізу одним із ключових етапів є формування її архітектурної структури. Цей етап передбачає визначення взаємозв'язків між компонентами, засобів збереження, обробки та представлення даних, а також забезпечення ефективного функціонування системи загалом. Правильне проектування структури має важливе значення для забезпечення масштабованості, продуктивності, надійності та зручності використання системи. Структура системи охоплює три основні рівні: рівень збору та обробки даних (бекенд [12]), рівень збереження даних (база даних [13]), а також рівень представлення даних користувачеві (фронтенд [14]).

Інформаційна система наукометричного аналізу та оцінки наукових здобутків розроблена на основі клієнт-серверної архітектури яку зображено на рисунку 2.1, що є одним із найпоширеніших підходів до побудови сучасних веб-орієнтованих програмних систем. Така архітектура забезпечує чіткий розподіл функціональних обов'язків між клієнтською та серверною частинами, сприяючи гнучкості, масштабованості та зручності в обслуговуванні системи.

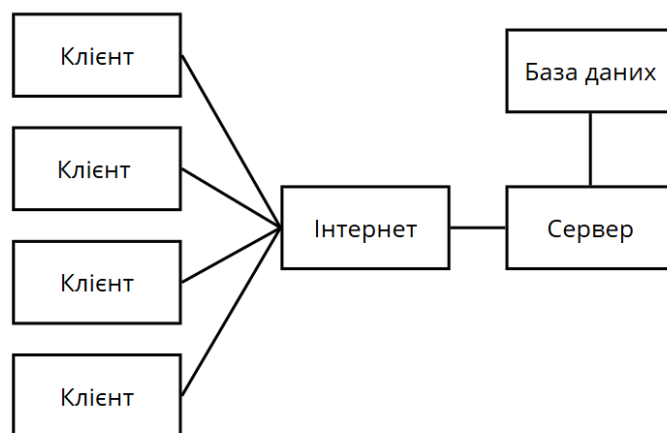


Рисунок 2.1 – Клієнт-серверна архітектура

У клієнт-серверній моделі обробка запитів користувача відбувається за допомогою двох основних компонентів:

Серверна частина системи – логічне ядро інформаційної системи, яке приймає, обробляє та відповідає на запити від клієнта. Саме на сервері відбувається основна обробка наукометричних даних, виконання алгоритмів аналізу, зберігання інформації в базі даних, взаємодія з зовнішніми API (наприклад, Google Scholar або Scopus). Серверна логіка реалізована з використанням таких технологій, як Python. Бекенд частина реалізується з використанням веб-фреймворку Django, що базується на мові програмування Python. Django [15] було обрано з огляду на його широкі функціональні можливості, включаючи вбудовані засоби ORM (Object-Relational Mapping [16]), підтримку REST API [17], зручну систему маршрутизації, а також велику кількість готових рішень для авторизації, обробки форм, безпеки та інше. Крім того, Django активно підтримується спільнотою, має добру документацію та дозволяє швидко створювати надійні вебсервіси, що є особливо важливим на етапі розробки прототипу системи.

Клієнтська частина системи – інтерфейс користувача, з яким взаємодіє кінцевий користувач через веб-браузер. Вона відповідає за збирання вхідних даних, відправлення запитів до сервера, відображення результатів обробки, візуалізацію аналітичних звітів, графіків тощо. Клієнтська частина реалізована із використанням сучасних веб-технологій, таких як HTML [18], CSS [18], JavaScript [18].

Обмін даними між клієнтом і сервером здійснюється через мережу з використанням протоколу HTTP(S) [19], при цьому формат передавання структурованої інформації представлений здебільшого у вигляді JSON [20]. Це дозволяє досягти високої сумісності та взаємодії з іншими сервісами.

Однією з ключових переваг обраної архітектури є її масштабованість – можливість розподілу обчислювальних навантажень між кількома серверними вузлами або винесення окремих компонентів (наприклад, бази даних чи модуля API) на окремі сервери. Також важливою є безпека, яка забезпечується шляхом

централізованого контролю доступу до даних, автентифікації користувачів та шифрування переданої інформації.

Завдяки чіткій структурі та розмежуванню відповідальностей клієнт-серверна модель дозволяє забезпечити модульність системи, що спрощує її супровід, оновлення та подальший розвиток. Наприклад, зміни в інтерфейсі користувача не потребують втручання в серверну частину, і навпаки.

Таким чином, клієнт-серверна архітектура є оптимальним вибором для реалізації веб-орієнтованої інформаційної системи наукометричного аналізу, що дозволяє ефективно обслуговувати велику кількість користувачів, обробляти значні обсяги даних та забезпечувати високу продуктивність і надійність функціонування системи.

Збір наукометричних даних здійснюється з двох основних джерел – Google Scholar та Scopus. Оскільки Google Scholar не надає офіційного API, було прийнято рішення використовувати зовнішні сервіси-агрегатори, такі як SerpApi, або програмне забезпечення типу Publish or Perish, які дозволяють отримувати публікаційну інформацію в автоматизованому режимі. Для Scopus, за наявності ліцензійного доступу, передбачено використання офіційного API, який забезпечує більш структурований та контрольований доступ до бази даних. На рівні бекенду реалізовані окремі модулі, які відповідають за формування запитів, обробку відповідей, нормалізацію метаданих та фільтрацію отриманих результатів.

Для збереження опрацьованої інформації використовується реляційна база даних SQLite [21]. Основною причиною вибору саме цієї СУБД є її простота, відсутність необхідності у налаштуванні серверного середовища, а також придатність для побудови прототипів. SQLite є вбудованим рішенням, яке не потребує окремого серверного процесу, що дозволяє розробникам зосередитися на логіці взаємодії з даними, не витрачаючи ресурси на адміністративне налаштування. Окрім цього, Django має повну підтримку SQLite, включаючи автоматичне створення таблиць на основі моделей, що суттєво спрощує розгортання системи.

У структурі бази даних передбачено кілька ключових сутностей, зокрема: профілі авторів, публікації, джерела даних, цитування, наукові інтереси, афіліції тощо. Особливу увагу приділено питанню уніфікації ідентифікаторів, зокрема використанню ORCID, якщо він наявний, або ж комбінації інших атрибутів (прізвище, ініціали, афіліція, тематика досліджень) для зіставлення профілів з різних джерел. Таким чином, база даних виступає як централізоване сховище нормалізованої інформації, на основі якої формуються аналітичні звіти.

Фронтенд системи реалізується з використанням HTML, CSS, JavaScript, а також шаблонізатора Jinja2 [22], що інтегрується у структуру Django. Застосування Jinja дозволяє генерувати динамічні HTML-сторінки на сервері з вбудованими даними, отриманими з бази даних. Це забезпечує зручну інтерактивну візуалізацію результатів, включаючи графіки цитувань, діаграми продуктивності, списки публікацій, звіти тощо.

Розроблений користувацький інтерфейс орієнтований на максимальну простоту взаємодії з системою. Реалізовано можливість фільтрації даних за авторами, роками, тематиками, джерелами, а також перегляду деталізованої інформації про окремі наукові доробки. У подальшому можливе доповнення фронтенду JavaScript-фреймворками (наприклад, Vue.js або React [23]) для реалізації розширеної клієнтської логіки, але у межах прототипу використано мінімалістичний підхід для зменшення складності.

На рисунку 2.2 продемонстровано загальну схему інформаційної системи.

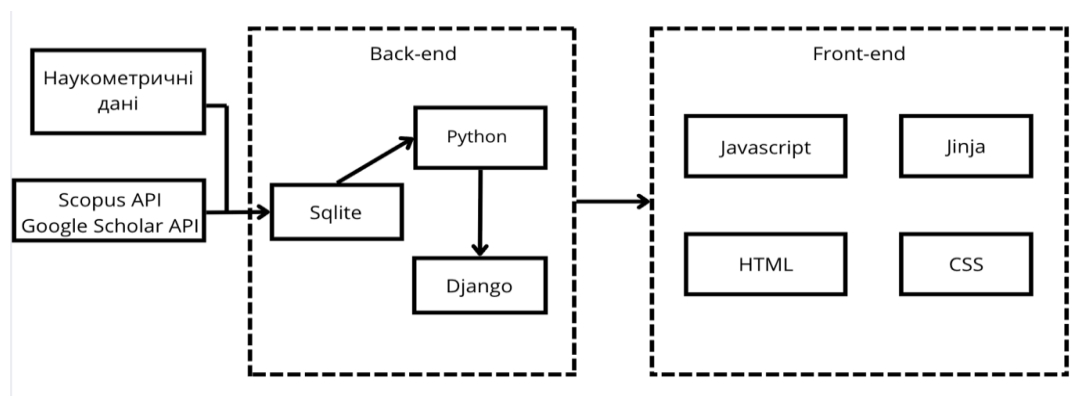


Рисунок 2.2 – Загальна схема інформаційної системи

Обрана структура інформаційної системи ґрунтується на принципах модульності, простоти, доступності та можливості масштабування. Ця структура зображена на рисунку 2.3.

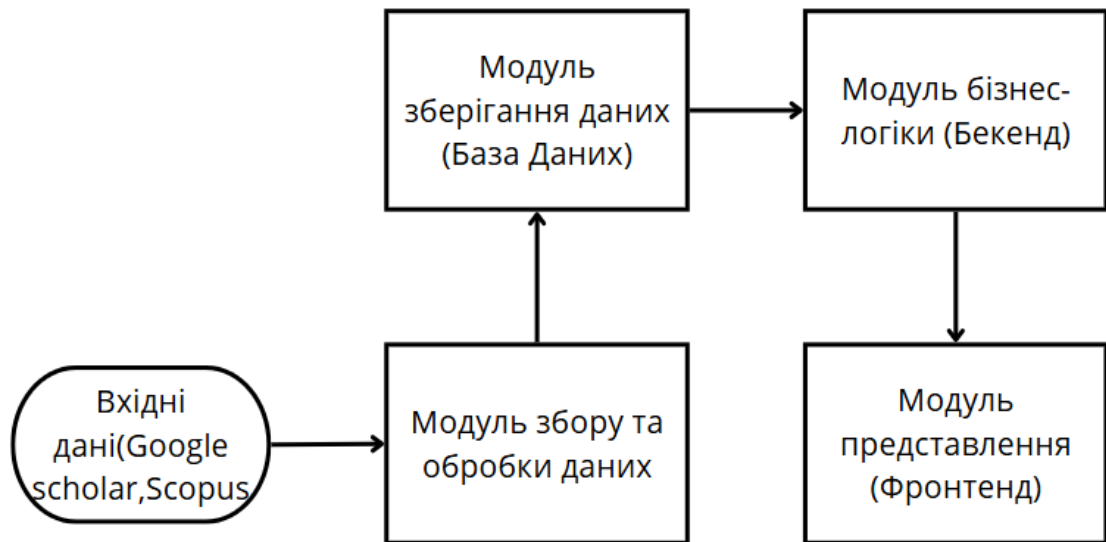


Рисунок 2.3 – Загальна структура інформаційної системи та напрямок потоку даних

Багаторівнева архітектура дозволяє незалежно розвивати окремі компоненти системи: джерела даних можна змінювати або додавати нові, не впливаючи на логіку обробки чи представлення; база даних легко замінюється на більш потужну СУБД (наприклад, PostgreSQL [24]) без істотних змін у коді завдяки абстракції ORM. Django як фреймворк дозволяє зберігати чітку логіку взаємодії між частинами системи та забезпечує високий рівень безпеки та надійності. Простий і зрозумілий фронтенд, побудований на основі стандартних веб-технологій, забезпечує зручність для кінцевого користувача без потреби в складному навчанні.

Таким чином, запропонована структура є оптимальним рішенням для реалізації прототипу системи, що дозволяє з мінімальними витратами ресурсів створити функціональний продукт, придатний для подальшого розширення, вдосконалення та впровадження у навчальних або наукових установах. Структура інформаційної системи дозволяє масштабування системи, отримання даних з інших сторонніх

сервісів, створення додаткового функціоналу, який буде корисний для користувачів інформаційної системи.

## 2.2 Алгоритмічне забезпечення системи

Алгоритм пошуку та обробки інформації про автора є ключовим елементом функціонування системи аналізу наукової діяльності. Його мета полягає у забезпеченні користувача актуальною, достовірною та структурованою інформацією щодо науковців, що дозволяє ефективно здійснювати подальший аналіз.

В основі цього алгоритму лежить послідовність дій, які включають як роботу з локальною базою даних, так і звернення до зовнішніх джерел через API сучасних наукометричних платформ, таких як Scopus чи Google Scholar.

Алгоритмічне забезпечення системи охоплює не лише формальні кроки виконання пошуку, але й визначає логіку прийняття рішень у процесі обробки запитів, зберігання результатів та взаємодії з користувачем.

Такий підхід дозволяє системі бути адаптивною, масштабованою та придатною до інтеграції з іншими модулями програмного забезпечення. Чітке структурування алгоритмів сприяє зменшенню помилок, підвищує продуктивність роботи сервісу та забезпечує ефективну автоматизацію рутинних завдань, що є надзвичайно важливим у сфері обробки великих обсягів наукової інформації.

На рисунку 2.4 зображено алгоритм пошуку авторів в інформаційній системі наукометричного аналізу та оцінки наукових здобутків. Алгоритм передбачає не тільки пошук авторів в існуючій базі даних, а й пошук у реальному часі якщо інформації про автора немає у базі даних.

Алгоритм складається з наступних кроків:

- ввід інформації про автора;
- пошук інформації про автора;
- опрацювання знайденої інформації;
- запис інформації в базу даних.

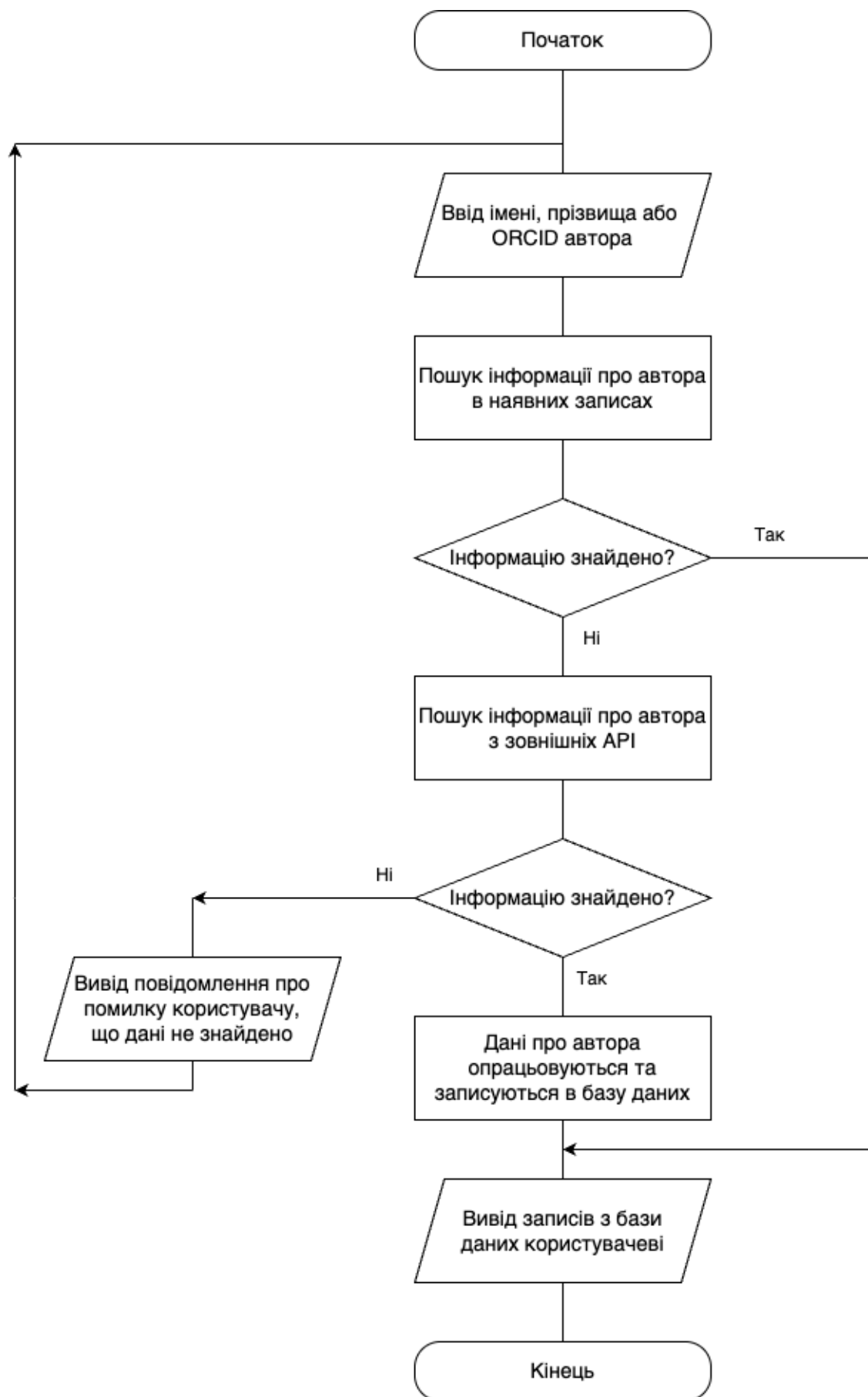


Рисунок 2.4 – Схема алгоритму пошуку авторів в інформаційній системі

Першим кроком взаємодії з системою є введення користувачем пошукового запиту, що містить ім'я, прізвище або ORCID-ідентифікатор науковця. Далі система ініціює пошук відповідної інформації у локальному сховищі даних. На цьому етапі здійснюється перевірка наявності раніше збережених записів, які відповідають введеним параметрам. Якщо інформація знайдена, вона миттєво передається у модуль представлення даних, де користувач може переглянути детальну інформацію про автора, зокрема його наукові інтереси, кількість публікацій, індекс Хірша, кількість цитувань, список співавторів тощо.

У випадку, якщо відповідних записів у локальній базі даних не виявлено, система переходить до наступного етапу – звернення до зовнішніх джерел даних. Формується запит до API обраного сервісу, що може містити ORCID, ім'я та прізвище автора. Важливим аспектом цього етапу є обробка помилок і відповідей від сторонніх сервісів: система має бути здатна ідентифікувати випадки, коли автор не знайдений, запит сформульовано некоректно або сервіс тимчасово недоступний.

Якщо у відповідь на API-запит система отримує структуровані дані про автора, ці дані проходять етап обробки: здійснюється перевірка повноти інформації, очищення від зайвих або дубльованих записів, стандартизація форматів тощо. Після обробки дані записуються до локальної бази даних, з урахуванням можливості подальшого оновлення чи синхронізації. Це забезпечує збереження результатів для подальшого використання без необхідності повторного звернення до зовнішнього джерела.

Після завершення обробки інформації система формує результат у зручному для користувача форматі. Візуалізація даних може супроводжуватися графічними елементами – наприклад, гістограмами розподілу публікацій за роками, діаграмами наукових інтересів або мережами співавторства. Такий підхід не лише забезпечує ефективний доступ до необхідної інформації, але й сприяє кращому її сприйняттю та подальшому аналізу.

У разі, якщо автор не знайдений ні у локальній базі даних, ні за допомогою зовнішніх API, система повідомляє користувача про відсутність відповідних записів

або про можливу помилку у введених даних. Це дозволяє оперативно скоригувати запит або повторити спробу через деякий час. Таким чином, алгоритм пошуку та обробки інформації про автора реалізує гнучку та надійну модель роботи з науковими даними, забезпечуючи високу точність і швидкість пошуку в умовах змінного обсягу інформації.

## 2.3 Інформаційне забезпечення

Інформаційне забезпечення веб-орієнтованої інформаційної системи включає в себе процес збору, обробки, зберігання в базі даних інформацію про авторів. База даних для інформаційної системи буде створена засобами SQLite3, який є середовищем для створення вбудованого типу бази даних, і не вимагає окремого серверу баз даних. База даних буде зберігатися в одному файлі, що полегшує роботу з нею. SQLite3 є базовою СУБД в Django, тому буде легко застосувати саме її. На рисунку 2.5 зображено код моделі у Django.

```
class AuthorModel(models.Model):
    first_name = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    author_name = models.CharField(max_length=200, unique=True)
    sub_title = models.CharField(max_length=1000, default="", blank=True)
    url = models.URLField(blank=True, null=True)
    orcid_id = models.CharField(max_length=100, default="", blank=True)
    scopus_id = models.CharField(max_length=100, default="", blank=True)
    eid = models.CharField(max_length=1000, default="", blank=True)
    date_created = models.CharField(max_length=100, default="", blank=True)
    affiliation_current = models.CharField(max_length=1000, default="", blank=True)
    interests = models.CharField(max_length=1000, default="", blank=True)
    affiliation_count = models.CharField(max_length=100, default="", blank=True)
    coauthor_count = models.CharField(max_length=100, default="", blank=True)
    h_index = models.CharField(max_length=100, default="", blank=True)
    document_count = models.CharField(max_length=100, default="", blank=True)
    citation_count = models.CharField(max_length=100, default="", blank=True)
```

Рисунок 2.5 – Модель Django для таблиці Author у базі даних.

У цьому коді визначається модель Author (сутність), яка є підкласом

models.Model в Django.Ця модель фактично відповідає таблиці Author у базі даних. Тобто модель в Django використовується для створення міграцій Django. Міграції Django це інструкції для створення таблиць або їх модифікування для Django ORM.

Django ORM це інструмент за допомогою якого можна модифікувати базу даних за потреби. Кожне поле моделі відображає стовпець в базі даних, яку Django створить під час міграцій. Основні поля (атрибути), які використовуються в цій моделі:

- author\_name – поле CharField, яке містить повне ім'я та прізвище автора;
- sub\_title – поле CharField, яке містить сферу діяльності автора, вказується максимальна довжина поля (1000 символів);
- url – поле URLField, яке містить посилання на сторінку автора на платформах Scopus або Google Scholar, якщо посилання було отримано;
- orcid\_id – поле CharField, яке містить ORCID автора, вказується максимальна довжина поля (100 символів);
- scopus\_id – поле CharField, яке містить SCOPUS\_ID автора, якщо автор є на Scopus, вказується максимальна довжина поля (100 символів);
- date\_created – поле CharField, яке містить першу дату реєстрації автора на платформах Scopus або Google Scholar;
- affiliation\_current – поле CharField, яке містить ідентифікатор інституційної приналежності автора (університету, наукового інституту тощо);
- interests – поле CharField, яке містить список інтересів автора, це поле може бути досить великим за обсягом символів тому встановлено обмеження (1000 символів);
- coauthor\_count – поле типу CharField, яке містить кількість авторів разом з якими даний автор має спільні роботи;
- h\_index – поле типу CharField, яке містить індекс Хірша автора;
- document\_count – поле типу CharField, яке містить кількість документів (робіт) автора;
- citation\_count – поле типу CharField, яке містить кількість цитувань автора.

За допомогою цієї моделі можна реалізувати функціонал пошуку авторів та виведення інформації про них. Також можна створити модель Affiliation (сутність) – ця модель буде відповідати за таблицю в базі даних Affiliation, яка буде містити дані про університети та інституції до яких будуть належні автори. На рисунку 2.6 зображено модель таблиці Affiliation.

```
class Affiliation(models.Model):
    affiliation_description = models.CharField(max_length=1000, default="", blank=True)
    affiliation_name = models.CharField(max_length=150, default="", blank=True)
    affiliation_id = models.IntegerField(default=0, blank=True)
    list_of_authors = models.CharField(max_length=1000, default="", blank=True)
```

Рисунок 2.6 – Модель Django таблиці Affiliation у базі даних

Ця модель є потрібною для того щоб отримувати список авторів, які належні до певної наукової інституції або університету. Це значно спростить в майбутньому реалізацію експортування звітності про усіх авторів певної інституції, про це детальніше описано у розділі 3. Основні поля (атрибути), які використовуються в цій моделі:

- affiliation\_name – поле типу CharField, яке містить назву певної наукової інституції або університету наприклад: Західноукраїнський національний університет;
- affiliation\_id – поле типу IntegerField, яке містить ідентифікатор певної наукової інституції або університету;
- affiliation\_description – поле типу CharField, яке містить опис певної наукової інституції або університету;
- list\_of\_authors – поле типу CharField, яке містить список ORCID авторів які належні до цієї наукової інституції або університету.

Ці моделі описують структуру таблиць у базі даних і визначають зв'язки між ними. Після виконання міграцій Django створить відповідні таблиці у базі даних на основі цих моделей.

## 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 3.1 Програмна реалізація системи

Перед початком розробки веб-орієнтованої інформаційної системи на Python, потрібно виконати декілька підготовчих кроків, а саме установити редактори коду:

- Visual Studio Code (VS Code) [25] – це легкий і потужний редактор коду, який підтримує багато мов програмування та надає розширення для поліпшення розробки веб-додатків;

- PyCharm [26] – це інтегроване середовище розробки (IDE) для мови програмування Python, яке надає розширені можливості для розробки веб-додатків.

Необхідно завантажити та налаштувати мови програмування:

- Python [27] – це високорівнева мова програмування, яка має простий синтаксис та велику кількість бібліотек для розробки веб-додатків;

- JavaScript (JS) – мова програмування, яка використовується для створення інтерактивних елементів на веб-сторінках і взаємодії з користувачем;

- HTML (HyperText Markup Language) – мова розмітки, яка використовується для створення структури веб-сторінок;

- CSS (Cascading Style Sheets) – мова опису стилів, яка використовується для задання зовнішнього вигляду веб-сторінок.

Необхідно завантажити потрібні фреймворки та стилі:

- Django – високорівневий веб-фреймворк на основі Python, який дозволяє швидко розробляти потужні веб-додатки з використанням моделей бази даних, URL-маршрутизації, автентифікації та інших функцій;

- Bootstrap [28] – популярний фреймворк для розробки веб-додатків з використанням HTML, CSS та JavaScript. Він надає набір стилів, компонентів та інструментів для швидкого створення сучасного та адаптивного інтерфейсу користувача.

Наступним кроком є встановлення необхідного оточення розробки Python. Це включає встановлення Python, налаштування віртуального середовища і установку необхідних пакетів і залежностей.

Після кроків, описаних вище, було розроблено функціональні компоненти веб-сайту, такі як сторінки, форми, робота з базою даних тощо, розподілено функціональність на окремі модулі або файли для зручності управління і підтримки коду. Розглянемо реалізацію деяких функцій з прикладами коду. Для початку розглянемо функцію пошуку авторів в інформаційній системі. На рисунку 3.1 зображено цю функцію.

```
def search_author(request):  
    if request.method == "POST":  
        form = SearchForm_Elsevier(request.POST)  
        if form.is_valid():  
            # Отримуємо дані з форми  
            first_name = form.cleaned_data["first_name"]  
            last_name = form.cleaned_data["last_name"]  
            orcid_id = form.cleaned_data.get("orcid_id")  
            results = None  
  
            try:  
                # Логіка пошуку  
                if orcid_id:  
                    # Пошук за ORCID  
                    results = ElsevierModel.objects.filter(orcid_id__icontains=orcid_id)  
                elif first_name and last_name:  
                    # Пошук за ім'ям та прізвищем  
                    results = ElsevierModel.objects.filter(  
                        author_name__icontains=f"{first_name} {last_name}"  
                    )  
                else:  
                    # Порожній результат, якщо нічого не введено  
                    results = ElsevierModel.objects.none()  
  
                return render(  
                    request,  
                    "search_results_els.html",  
                    {"results": results, "form": form},  
                )  
            except Exception as e:  
                # Обробка помилок  
                print(f"Error during search: {e}")  
                results = ElsevierModel.objects.none()  
                return render(  
                    request,  
                    "search_results_els.html",  
                    {"results": results, "form": form},  
                )  
        else:  
            form = SearchForm_Elsevier()  
  
    return render(request, "search_template_els.html", {"form": form})
```

Рисунок 3.1 – Функція пошуку авторів в інформаційній системі

За допомогою цієї функції реалізовано пошук авторів в інформаційній системі за ім'ям та прізвищем або ORCID автора. Наступними функціями є функції збору даних із двох сторонніх API сервісів а саме Google Scholar та Scopus. Для цього використані їх офіційні API, для Google Scholar - SerpApi, для Scopus - ElsevierApi. Ці функції реалізовано за допомогою бібліотеки Python під назвою requests, яка дозволяє надсилати запити до API.

Наступною функцією є функція генерації звітів про автора. Ця функція використовується на сторінці з інформацією про конкретного автора та дає можливість отримати звіт з даними про автора в форматі PDF. На рисунку 3.2 зображено цю функцію.

```
def generate_pdf(request, id_):
    # Отримуємо дані автора
    author = get_object_or_404(ElsevierModel, id=id_)

    # Генеруємо HTML із шаблону
    html = render_to_string('author_report.html', {'author': author, 'now': datetime.now()})

    # Створюємо HTTP-відповідь із типом PDF
    response = HttpResponse(content_type='application/pdf')
    response['Content-Disposition'] = f'attachment; filename="author_{id_}.pdf"'

    # Конвертуємо HTML у PDF
    pisa_status = pisa.CreatePDF(html, dest=response)

    # Перевіряємо на помилки
    if pisa_status.err:
        return HttpResponse('Помилка при створенні PDF', status=500)

    return response
```

Рисунок 3.2 – Функція створення звітів про авторів

Також реалізовано функцію для сторінки з усіма авторами в системі. Ця функція відповідає за пагінацію сторінок де можна побачити по 12 авторів на сторінці. Цю функцію зображено на рисунку 3.3.

```
def pagination(request):
    page = request.GET.get("page", 1)
    per_page = 12

    data_list = MyData.objects.order_by("full_name")
    paginator = Paginator(data_list, per_page)

    try:
        data = paginator.page(page)
    except:
        data = paginator.page(paginator.num_pages)

    return render(request, "list_authors.html", {"data": data})
```

Рисунок 3.3 – Функція пагінації та виводу авторів на сторінку з усіма авторами

Дані для пошуку авторів вводяться в систему за допомогою розробленої форми, яка є вбудованою в Django.Forms. На рисунку 3.4 зображено дана форма для вводу даних за якими виконується пошук авторів.

```
class SearchForm_Author(forms.Form):
    first_name = forms.CharField(
        max_length=50,
        widget=forms.TextInput(
            attrs={"class": "form-control", "placeholder": "Enter first name"}
        ),
        required=False,
    )

    last_name = forms.CharField(
        max_length=50,
        widget=forms.TextInput(
            attrs={"class": "form-control", "placeholder": "Enter last name"}
        ),
        required=False,
    )

    orcid_id = forms.CharField(
        label="ORCID ID",
        widget=forms.TextInput(attrs={
            "class": "form-control",
            "placeholder": "Введіть ORCID ID"
        })),
        required=False,
    )

    def clean(self):
        cleaned_data = super().clean()
        first_name = cleaned_data.get("first_name")
        last_name = cleaned_data.get("last_name")
        orcid_id = cleaned_data.get("orcid_id")
        # Check if both first_name and last_name are empty
        if not orcid_id and not (first_name and last_name):
            raise forms.ValidationError(
                "Введіть або ORCID, або ім'я та прізвище."
            )
        return cleaned_data
```

Рисунок 3.4 – Форма вводу даних для пошуку авторів

Загальна структура системи зображена на рисунку 3.5. Створено усі необхідні файли для коректної роботи усіх компонентів системи, розроблено спеціальні

шаблони сторінок, які Django використовує для формування усіх сторінок веб-орієнтованої інформаційної системи.

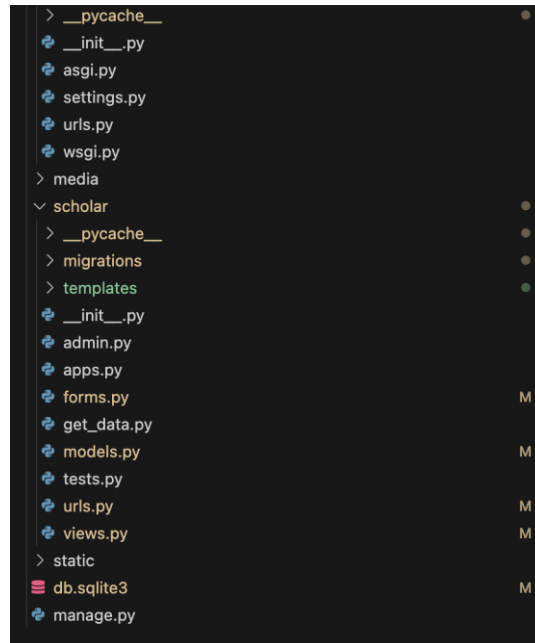


Рисунок 3.5 – Структура файлів інформаційної системи

Для правильної маршрутизації [29] по інформаційній системі створено Django маршрутизацію, яку описано у файлі urls.py. Ця маршрутизація є інструкцією для Django, яку функцію використати при переході на сторінки. На рисунку 3.6 зображено файл маршрутизації інформаційної системи.

```
from django.urls import path
from . import views
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
from django.conf.urls.static import static
from django.conf import settings
from scholar.views import PersonListView

urlpatterns = [

    path("author_detail/<int:scopus_id>/", views.elsevier_detail, name="elsevier_detail"),
    path('persons/', views.indexx, name='indexx'),
    path('leaderboard/', views.leaderboard, name='leaderboard'),
    path('author/<int:id>/generate-pdf/', views.generate_pdf, name='generate_pdf'),

]

You, 19 months ago • startproject

if settings.DEBUG:
    urlpatterns += staticfiles_urlpatterns() + static(
        settings.MEDIA_URL, document_root=settings.MEDIA_ROOT
    )
```

Рисунок 3.6 – Файл маршрутизації інформаційної системи

### 3.2 Інтерфейс користувача

Інтерфейс користувача є важливою складовою будь-якої інформаційної системи, адже саме він забезпечує взаємодію користувача з функціональними модулями [30]. У межах розробленої інформаційної системи наукометричного аналізу та оцінки наукових здобутків інтерфейс реалізовано у вигляді сучасного веб-застосунку з використанням HTML, CSS та JavaScript. Основною метою було створення простого, інтуїтивного та візуально привабливого середовища, що дозволяє швидко здійснювати пошук науковців та авторів за різними параметрами та переглядати результати аналізу їхньої наукової діяльності.

На рисунку 3.7 представлено головну сторінку інформаційної системи – це центральний елемент взаємодії користувача з системою. У верхній частині розміщено заголовок системи: «Інформаційна система наукометричного аналізу та оцінки наукових здобутків», що чітко окреслює призначення системи. Праворуч – навігаційне меню у вигляді трьох кнопок: Leaderboard, Home Page та List Of Authors, які дозволяють переміщатися між розділами.

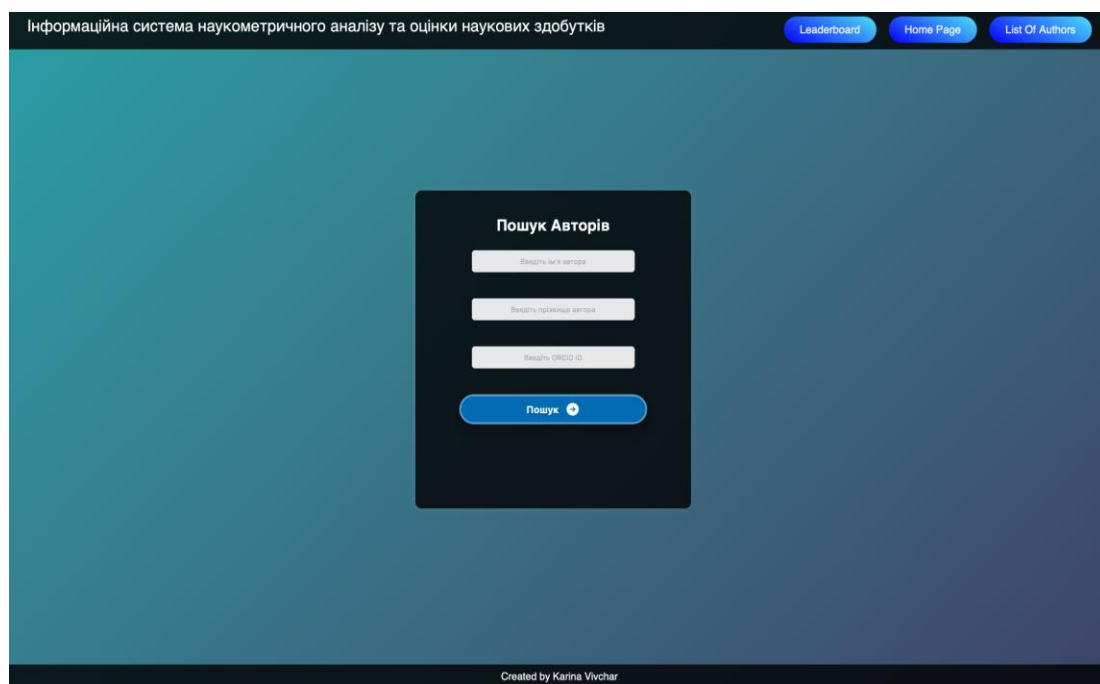


Рисунок 3.7 – Головна сторінки інформаційної системи

При переході на цю сторінку користувач повинен ввести дані у форму по яких буде виконаний пошук авторів. Пошук можливо здійснити при введенні імені, прізвища або ORCID автора. При введенні коректних даних та при натисканні кнопки “Пошук” буде здійснено пошук авторів, якщо їх знайдено то буде виконано перехід на наступну сторінку з результатами пошуку. Сторінку з результатами пошуку зображено на рисунку 3.8.

| Результати пошуку: |                           |
|--------------------|---------------------------|
| Ім'я               | Посада                    |
| Iryna Turchenko    | доцент кафедри ІОСУ, ЗУНУ |

Рисунок 3.8 – Сторінка з результатами пошуку авторів

Блок результатів реалізовано у вигляді таблиці з відповідними колонками: Ім’я та Посада. Знайдений автор це Iryna Turchenko, яка є доцентом кафедри ІОСУ, ЗУНУ. Такий підхід дозволяє одразу переглянути ключову інформацію про науковця без переходу на додаткові сторінки.

Інтерфейс адаптований для відображення різної кількості результатів: за потреби можливе розширення таблиці, впровадження пагінації або фільтрації результатів. Дизайн результатів відповідає загальному стилю системи це темна панель з білим текстом забезпечує контрастність і комфорт при читанні.

Далі можна перейти на сторінку з інформацією про конкретного автора натиснувши на відповідного автора в таблиці результатів пошуку. демонструє розширений профіль автора. Ця сторінка відкривається після натискання на автора в результатах пошуку. У профілі відображається детальна інформація про науковця, включаючи: повне ім'я; наукове звання та посаду; місце роботи; ключові напрями досліджень; кількість публікацій; кількість цитувань; h-індекс.

Сторінка структурована логічно, із чітко виділеними блоками інформації, що дозволяє швидко знайти потрібні дані. При наявності API-зв'язку з Google Scholar або Scopus дані автоматично оновлюються, що гарантує актуальність інформації.

Оформлення сторінки витримано в єдиному стилі з головною сторінкою: темний фон, білий шрифт, плавні градієнти. Кожен інформаційний блок візуально відділений для зручності сприйняття. Сторінку з інформацією про автора зображено на рисунку 3.9.

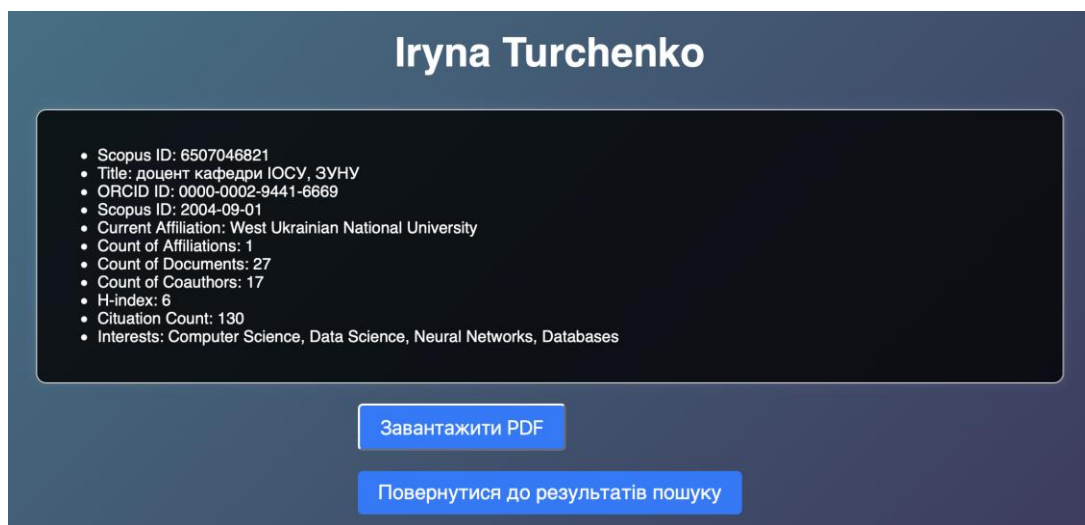


Рисунок 3.9 – Сторінка з інформацією про автора

На рисунку можна побачити усю знайдену інформацію про автора та кнопки для повернення до форми пошуку для наступного пошуку авторів і кнопку для формування звіту в форматі PDF з інформацією про автора. Це особливо корисно для аналітиків, викладачів, співробітників наукових установ або адміністрації закладів освіти, оскільки дозволяє зберігати, роздруковувати або передавати результати

наукометричного аналізу у зручному форматі. Автоматичне створення звіту економить час, зменшує кількість ручної роботи та забезпечує стандартизований вигляд представлених даних. На рисунку 3.10 зображено, як виглядає звіт про автора.

| Author Report                                   |                           |
|-------------------------------------------------|---------------------------|
| Iryna Turchenko                                 |                           |
| ID: 39                                          |                           |
| ORCID: 0000-0002-9441-6669                      |                           |
| Affiliation: West Ukrainian National University |                           |
| Details                                         |                           |
| Field                                           | Value                     |
| H-Index                                         | 6                         |
| Citations                                       | 130                       |
| Research Area                                   | доцент кафедри ІОСУ, ЗУНУ |
| Publications Count                              | 27                        |
| Citations Count                                 | 130                       |
| Coauthor Count                                  | 17                        |
| Affiliation Count                               | 1                         |

Generated on 08 Jun 2025 23:27

Рисунок 3.10 – Приклад згенерованого звіту з інформацією про автора

Звіт містить ключову інформацію про обраного автора, а саме ім'я та прізвище автора, Id автора в інформаційній системі, ORCID автора та наукометричні дані які було знайдено про даного автора.

Для реалізації функціоналу формування PDF використовується шаблон HTML, в якому динамічно підставляються значення, отримані з бази даних або API. Це робить процес гнучким, масштабованим та незалежним від типу отриманих даних.

Таким чином, PDF-звіт є зручною формою представлення результатів аналізу наукової діяльності, що може бути використаний як у внутрішньому документообігу навчальних та наукових установ, так і для зовнішньої презентації наукових досягнень.

### 3.3 Тестування веб-орієнтованої інформаційної системи

Тестування веб-орієнтованої інформаційної системи наукометричного аналізу та оцінки наукових здобутків, включає проведення різних видів тестів для перевірки його функціональності, коректності роботи та відповідності вимогам. Нижче розглянемо кілька типів тестів, які були проведені:

- виконання тестового сценарію для перевірки правильності роботи основних функцій системи, таких як пошук автора лише за іменем, пошук автора лише за прізвищем, пошук автора за ORCID, пошук авторів без введення даних у форму пошуку.

- перевірка зручності взаємодії користувача з веб-ресурсу, перевірка коректності відображення інформації, перевірка відповідності дизайну та вимогам щодо вигляду та поведінки;

- перевірка точності пошуку;

- перевірка, забезпечення коректної обробки та відображення помилок для користувача;

- перевірка швидкості завантаження сторінок, відповідного реагування на запити користувачів та можливості масштабування сайту при великій кількості користувачів або великому обсязі даних;

- перевірка відображення та роботи веб-орієнтованої інформаційної системи на різних браузерах, операційних системах та пристроях, щоб забезпечити її сумісність різними платформами.

Тестування відображення системи в різних браузерах приведено в таблиці 3.1. Результати тестування відображення системи на різних браузерах свідчать про те, система коректно відкривається за допомогою різних браузерів.

Таблиця 3.1 – Тестування на різних браузерах

| Опис                              | Кроки                                                                                 | Результат                                                               |
|-----------------------------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Відкриття сайту в Google Chrome   | 1. Відкриття браузера Google Chrome<br>2. Введення URL-адреси<br>3. Натиснути Enter   | Сайт успішно відкривається і коректно відображається на Google Chrome   |
| Відкриття сайту в Mozilla Firefox | 1. Відкриття браузера Mozilla Firefox<br>2. Введення URL-адреси<br>3. Натиснути Enter | Сайт успішно відкривається і коректно відображається на Mozilla Firefox |
| Відкриття сайту в Microsoft Edge  | 1. Відкриття браузера Microsoft Edge<br>2. Введення URL-адреси<br>3. Натиснути Enter  | Сайт успішно відкривається і коректно відображається на Microsoft Edge  |
| Відкриття сайту в Opera           | 1. Відкриття браузера Opera<br>2. Введення URL-адреси<br>3. Натиснути Enter           | Сайт успішно відкривається і коректно відображається на Opera           |
| Відкриття сайту в Safari          | 1. Відкриття браузера Safari<br>2. Введення URL-адреси<br>3. Натиснути Enter          | Сайт успішно відкривається і коректно відображається на Safari          |

При тестуванні системи виявлено, що інтерфейс системи є зручним для користувача. Усі елементи управління логічно згруповані, форма пошуку авторів є інтуїтивно зрозумілою, що дозволяє користувачу швидко орієнтуватися та здійснювати пошук без додаткових інструкцій. Кольорова гама інтерфейсу сприяє комфортному сприйняттю інформації, а кнопки мають чітке візуальне оформлення та

зрозумілі підписи. Навігація між сторінками відбувається без затримок, а також реалізовано зручний механізм повернення до головного меню. Таким чином, система відповідає сучасним вимогам до юзабіліті та забезпечує позитивний досвід взаємодії з користувачем.

При тестуванні системи була проведена перевірка швидкості завантаження сторінок, відповідного реагування на запити користувачів та можливості масштабування веб-інтерфейсу при високому навантаженні. Завантаження головної сторінки, сторінки результатів пошуку та сторінки з інформацією про автора відбувалося в межах допустимого часу (менше 2 секунд при стандартному з'єднанні), що свідчить про оптимізовану структуру коду та ефективну обробку даних. Система коректно реагувала на всі запити користувача, навіть при паралельному надсиланні кількох пошукових запитів, не втрачаючи стабільності роботи. Також було змодельовано навантаження з великою кількістю одночасних користувачів і запитів до бази даних, що дозволило переконатися у можливості масштабування ця система продовжувала функціонувати без збоїв, зберігаючи прийнятний рівень продуктивності. Це підтверджує, що розроблений програмний модуль готовий до використання в умовах реального середовища з потенційно високим трафіком.

Перевірка точності пошуку проводилась з метою оцінити, наскільки ефективно система знаходить релевантну інформацію про науковців за заданими вхідними даними. Для цього використовувалися різні комбінації пошукових запитів: повне ім'я автора, часткове ім'я, ім'я з помилками або додатковими пробілами. У процесі тестування перевірялися відповідність знайдених результатів реальним даним, отриманим із наукометричних баз (наприклад, Scopus або Google Scholar).

Результати показали, що при введенні повного імені автора система з високою точністю повертала правильну інформацію, включаючи ORCID, афіліацію, кількість публікацій, h-індекс тощо. У випадках введення часткового імені або незначних помилок система також демонструвала стійкість, повертаючи найбільш відповідні варіанти, що свідчить про наявність реалізованих механізмів обробки нечіткого пошуку. Загалом, точність пошуку була оцінена як висока, що підтверджує

ефективність реалізованого алгоритму пошуку та надійність роботи системи у реальних умовах використання. Результати виконання тестових сценаріїв для перевірки правильності роботи основних функцій системи приведено в таблиці 3.2.

Таблиця 3.2 – Виконання тестових сценаріїв

| Опис                             | Кроки                                              | Результат                                                |
|----------------------------------|----------------------------------------------------|----------------------------------------------------------|
| Пошук авторів за іменем          | 1. Введення імені<br>2. Натиснути кнопку пошуку    | Отримано правильний список авторів за введеним іменем    |
| Пошук авторів за прізвищем       | 1. Введення прізвища<br>2. Натиснути кнопку пошуку | Отримано правильний список авторів за введеним прізвищем |
| Пошук авторів за ORCID           | 1. Введення ORCID<br>2. Натиснути кнопку пошуку    | Отримано правильний список авторів за введеним ORCID     |
| Пошук авторів без введення даних | 1. Не вводити нічого<br>2. Натиснути кнопку пошуку | Не отримано списку авторів                               |

За результатами наведеними у таблиці можна побачити, що система працює правильно під час виконання тестових сценаріїв, проте користувач не отримує повідомлення про неправильний ввід даних, якщо нічого не введено в форму пошуку, тому реалізовано функціонал, який забезпечує виведення помилки для користувача. Помилка свідчить про те що потрібно ввести ім'я, прізвище або ORCID автора для того щоб отримати правильні результати пошуку. Приклад виведення помилка для користувача зображено на рисунку 3.11.

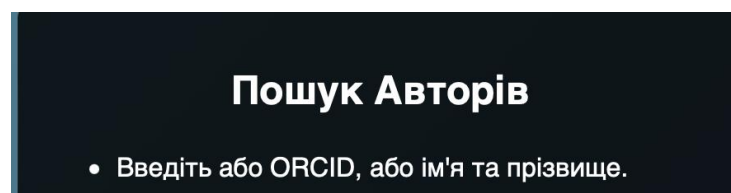


Рисунок 3.11 – Помилка при пошуку авторів без введення даних в форму

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи:

1. Проведено аналіз предметної області, що дало змогу отримати необхідні знання про наукометричні дані та проблеми предметної області.
2. Проаналізовано існуючі аналоги і досліджено їх переваги та недоліки, Цей аналіз дав змогу сформулювати основні вимоги до веб-орієнтованої інформаційної системи і зробити постановку задачі проектування.
3. Описано структуру веб-орієнтованої інформаційної системи кінопродукції, яка включає модулі, компоненти та взаємодію між ними.
4. Розроблено алгоритмічне забезпечення веб-орієнтованої інформаційної системи.
5. Спроектовано та реалізовано базу даних в середовищі SQLite, в якій зберігається інформація про користувачів та фільми.
6. Реалізовано основний функціонал веб-орієнтованої інформаційної системи на мові програмування Python за допомогою фреймворку Django та Python бібліотек;
7. Розроблено інтерфейс веб-орієнтованої системи за допомогою технологій Bootstrap, HTML та CSS.
8. Проведено тестування та налагодження веб-орієнтованої інформаційної системи для перевірки правильності роботи функціоналу. Виявлені помилки були виправлені.
9. Розроблена веб-орієнтована інформаційна система наукометричного аналізу та оцінки наукових здобутків може використовуватися науковцями та дослідниками для моніторингу власної наукової активності, керівництвом наукових установ для оцінки ефективності роботи, освітніми установами для підготовки звітів та проходження акредитаційних процедур та іншими користувачами для отримання інформації про авторів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bui Y. N. Navigating the academic landscape: Scopus, Web of Science, and beyond / Y. N. Bui. Thousand Oaks : SAGE Publications, 2020. 180 с.
2. Web of Science. URL: <https://dntb.gov.ua/web-of-science>
3. Hennig N. Google Scholar and more: the librarian's quick guide to digital research tools / N. Hennig. Chicago : ALA Editions, 2016. 120 с.
4. Moed H. F. API-based bibliometric analysis with Scopus and Web of Science / H. F. Moed. Cham : Springer, 2020. 280 с.
5. Mitchell R. Web scraping with Python: collecting data from the modern web / R. Mitchell. 2nd ed. Sebastopol : O'Reilly Media, 2018. 432 с.
6. Rice R., Southall J. The data librarian's handbook / R. Rice, J. Southall. London : Facet Publishing, 2016. 208 с.
7. Gaming the metrics: misconduct and manipulation in academic research / ed. by M. Biagioli, A. Lippman. Cambridge : MIT Press, 2020. 400 с.
8. Priem J., Piwowar H., Orr R. OpenAlex: a fully-open index of scholarly works, authors, venues, institutions, and concepts / J. Priem, H. Piwowar, R. Orr. arXiv preprint arXiv:2205.01833, 2022. 15 с.
9. Документація SerpApi. URL: <https://serpapi.com/>
10. Harzing A.-W. Publish or perish: the rise and fall of research metrics / A.-W. Harzing. Cham : Palgrave Macmillan, 2019. 250 с.
11. Whittington J. PDF explained / J. Whittington. Sebastopol : O'Reilly Media, 2001. 300 с.
12. Vincent W. Django for professionals: production websites with Python & Django / W. Vincent. Self-published, 2020. 250 с.
13. Silberschatz A., Korth H. F., Sudarshan S. Database system concepts / A. Silberschatz, H. F. Korth, S. Sudarshan. 7th ed. New York : McGraw-Hill Education, 2019. 1376 с.

14. Duckett J. HTML and CSS: design and build websites / J. Duckett. Hoboken : Wiley, 2011. 512 c.
15. Vincent W. Django for beginners: build websites with Python and Django / W. Vincent. Self-published, 2018. 200 c.
16. Fowler M. Patterns of enterprise application architecture / M. Fowler. Boston : Addison-Wesley Professional, 2002. 560 c.
17. Richardson L., Amundsen M., Ruby S. RESTful web APIs / L. Richardson, M. Amundsen, S. Ruby. Sebastopol : O'Reilly Media, 2013. 384 c.
18. Robbins J. N. Learning web design: a beginner's guide to HTML, CSS, JavaScript, and web graphics / J. N. Robbins. 5th ed. Sebastopol : O'Reilly Media, 2018. 600 c
19. Gourley D., Totty B. HTTP: the definitive guide / D. Gourley, B. Totty. Sebastopol : O'Reilly Media, 2002. 496 c.
20. Marrs T. JSON at work: practical data integration for the web / T. Marrs. Sebastopol : O'Reilly Media, 2012. 350 c.
21. Kreibich J. A. Using SQLite / J. A. Kreibich. Sebastopol : O'Reilly Media, 2010. 358 c.
22. Melé A. Django 3 by example / A. Melé. Birmingham : Packt Publishing, 2020. 448 c.
23. Banks A., Porcello E. Learning React: modern patterns for developing React apps / A. Banks, E. Porcello. 2nd ed. Sebastopol : O'Reilly Media, 2020. 350 c.
24. Obe R., Hsu L. PostgreSQL: up and running: a practical introduction to the advanced open source database / R. Obe, L. Hsu. Sebastopol : O'Reilly Media, 2017. 234 c.
25. Johnson B. Visual Studio Code: end-to-end editing and debugging tools for web developers / B. Johnson. Birmingham : Packt Publishing, 2016. 168 c.
26. Sorokin D. Mastering PyCharm / D. Sorokin. Birmingham : Packt Publishing, 2019. 250 c.
27. Matthes E. Python crash course: a hands-on, project-based introduction to programming / E. Matthes. 2nd ed. San Francisco : No Starch Press, 2019. 544 c.

28. Lett J. Bootstrap 4 quick start: a beginner's guide to building responsive layouts with Bootstrap 4 / J. Lett. Scotts Valley : CreateSpace Independent Publishing Platform, 2018. 110 с.

29. Greenfeld D. R., Greenfeld A. R. Two scoops of Django 3.x: best practices for the Django web framework / D. R. Greenfeld, A. R. Greenfeld. 3rd ed. Austin : Two Scoops Press, 2020. 460 с.

30. Tidwell J. Designing interfaces: patterns for effective interaction design / J. Tidwell. 3rd ed. Sebastopol : O'Reilly Media, 2019. 576 с.

31. Вівчар К. Турченко І. Веб-орієнтована інформаційна система наукометричного аналізу та оцінки наукових здобутків. *Розвиток науки: теорії, відкриття та практичні результати*: матеріали 4-ї міжнародної науково-практичної конференції (Цюріх, 9-11 черв. 2025 р.). Цюріх, 2025. С. 128-130

32. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

## Додаток А

### Код HTML шаблону сторінки з інформацією про автора

```
<body>
<div class="bg"></div>
<div class="body_container">
  <div class="header-container">
    <div class="header-title">Інформаційна система наукометричного аналізу та оцінки наукових здобутків</div>
    <div class="header-links">
      <a href="/leaderboard/">Leaderboard</a>
      <a href="/search">Home Page</a>
      <a href="/persons/?page=1">List Of Authors</a>
    </div>
  </div>

  <div class="search-container">
    <h2 class="title">Пошук Авторів</h2>
    <form method="post" action="{% url 'search_elsevier' %}">
      {% csrf_token %}
      {% if form.non_field_errors %}
        <div class="alert alert-danger">
          {{ form.non_field_errors }}
        </div>
      {% endif %}
      <div class="form-group">
        <label for="first_name"></label>
        {{ form.first_name }}
      </div>
      <div class="form-group">
        <label for="last_name"></label>
        {{ form.last_name }}
      </div>
      <div class="form-group">
        <label for="orcid"></label>
        {{ form.orcid_id }}
      </div>
      <button type="submit" class="button">
        Пошук
        <svg fill="currentColor" viewBox="0 0 24 24" class="icon">
          <path
            clip-rule="evenodd"
            d="M12 2.25c-5.385 0-9.75 4.365-9.75 9.75s4.365 9.75 9.75 9.75 9.75-4.365 9.75-9.75S17.385 2.25 12 2.25"
            fill-rule="evenodd"
          ></path>
        </svg>
      </button>
    </form>
  </div>

  <div class="results-container">
    {% if results %}
      <h3>Результати пошуку:</h3>
      <table class="table table-striped">
        <thead>
          <tr>
            <th scope="col">Ім'я</th>
            <th scope="col">Посада</th>
            <!-- ... (додайте інші необхідні стовпці) ... -->
          </tr>
        </thead>
        <tbody>
          {% for result in results %}
            <tr>
              <td>
                <a href="{% url 'elsevier_detail' result.id %}">{{ result.author_name }}</a>
                <td>{{ result.sub_title }}</td>
                <!-- ... (додайте дані інших стовпців) ... -->
              </tr>
            {% endfor %}
          </tbody>
        </table>
      {% else %}
        <p>Результати пошуку відсутні.</p>
      {% endif %}
    </div>

    <div class="created-by">Created by Karina Vivchar</div>

    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-DfXdz2htPH0lsSS5S5nCTpuj/zy4C+06pmoFVy38MV8nE+IbbVYUew+0rCXaRkfj"
      crossorigin="anonymous"
    ></script>
  </body>
</html>
```

## Додаток Б

### Код HTML шаблону для генерації звіту в форматі PDF

```
<body>
  <h1>Author Report</h1>
  <h2>{{ author.author_name }}</h2>

  <p><strong>ID:</strong> {{ author.id }}</p>
  <p><strong>ORCID:</strong> {{ author.orcid_id }}</p>
  <p><strong>Affiliation:</strong> {{ author.affiliation_current }}</p>

  <h3>Details</h3>
  <table>
    <thead>
      <tr>
        <th>Field</th>
        <th>Value</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>H-Index</td>
        <td>{{ author.h_index }}</td>
      </tr>
      <tr>
        <td>Citations</td>
        <td>{{ author.citation_count }}</td>
      </tr>
      <tr>
        <td>Research Area</td>
        <td>{{ author.sub_title }}</td>
      </tr>
      <tr>
        <td>Publications Count</td>
        <td>{{ author.document_count }}</td>
      </tr>
      <tr>
        <td>Citations Count</td>
        <td>{{ author.citation_count }}</td>
      </tr>
      <tr>
        <td>Coauthor Count</td>
        <td>{{ author.coauthor_count }}</td>
      </tr>
      <tr>
        <td>Affiliation Count</td>
        <td>{{ author.affiliation_count }}</td>
      </tr>
    </tbody>
  </table>

  <div class="footer">
    <p>Generated on {{ now|date:"d M Y H:i" }}</p>
  </div>
</body>
```

Додаток В

Копія опублікованих результатів

eoss-conf.com



ISSUE  
Nº39



COLLECTION OF SCIENTIFIC PAPERS



4<sup>TH</sup> INTERNATIONAL  
SCIENTIFIC  
AND PRACTICAL  
CONFERENCE

EVOLVING SCIENCE:  
THEORIES, DISCOVERIES  
AND PRACTICAL  
OUTCOMES

JUNE 9-11, 2025, ZURICH, SWITZERLAND



|                                                                                                                                                                              |     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>Білоус В.І., Лапін А.В.</b><br>ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ<br>ЗБЕРЕЖЕННЯ ТА ОНОВЛЕННЯ ІНФОРМАЦІЇ ПРО РОДИННУ<br>СПАДЩИНУ.....                               | 108 |
| <b>Кравченко І.В., Ковальчук М.О.</b><br>ІНТЕЛЕКТУАЛЬНА ОБРОБКА ЕКОЛОГІЧНИХ ДАНИХ ДЛЯ<br>СИСТЕМ ЛАНДШАФТНОГО МОНІТОРИНГУ В ІНФОРМАЦІЙНІЙ<br>БЕЗПЕЦІ.....                     | 113 |
| <b>Makhruka Aia, Turchenko I.</b><br>DEVELOPMENT OF A WEB-BASED PERSONALIZED TUTORING<br>PLATFORM: A HUMAN-CENTERED APPROACH TO ADAPTIVE<br>LEARNING.....                    | 126 |
| <b>Вівчар К.С., Турченко І.В.</b><br>ВЕБ-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА<br>НАУКОМЕТРИЧНОГО АНАЛІЗУ ТА ОЦІНКИ НАУКОВИХ<br>ЗДОБУТКІВ.....                                    | 128 |
| <b>Mushtyn O., Sobko O., Molchanova M., Mazurets O.</b><br>CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE FOR<br>IMAGE-BASED ARCHITECTURAL STYLE RECOGNITION.....                 | 130 |
| <b><u>Section: Jurisprudence</u></b>                                                                                                                                         |     |
| <b>Хорольська Т.В., Телійчук В.Г.</b><br>ПОНЯТТЯ ТА СУТНІСТЬ ЗАЛУЧЕННЯ ГРОМАДСЬКОСТІ ДО<br>ВИКОНАННЯ ЗАВДАНЬ ОПЕРАТИВНО-РОЗШУКОВОЇ<br>ДІЯЛЬНОСТІ.....                        | 144 |
| <b>Шевченко О.О., Логінова М.В.</b><br>ЗАХИСТ ТРУДОВИХ ПРАВ ПРАЦІВНИКА В УМОВАХ<br>ВОЄННОГО СТАНУ: СУЧАСНІ ВИКЛИКИ ТА ШЛЯХИ<br>ПОДОЛАННЯ.....                                | 146 |
| <b>Білецький Є.І., Копилов Е.В.</b><br>МІЖНАРОДНО-ПРАВОВІ ЗАСАДИ ТА ПРАКТИКА ВЗАЄМОДІЇ<br>УКРАЇНИ З ЄВРОПОЛОМ ТА ІНТЕРПОЛОМ У СФЕРІ<br>ОПЕРАТИВНО-РОЗШУКОВОЇ ДІЯЛЬНОСТІ..... | 149 |

2. Khosravi H., Sadiq S., Gasevic D. Learning analytics and learner modeling for intelligent tutoring systems: A systematic review // *Computers & Education: Artificial Intelligence*. – 2022. – Vol. 3. – Article 100068.3. Zhang X., Liu C., Yu H. A review of sign language recognition methods // *Pattern Recognition Letters*. – 2021. – Vol. 148. – P. 3–11.
3. Xu B., Wang J. The impact of interaction design on learner engagement in adaptive learning environments // *British Journal of Educational Technology*. – 2020. – Vol. 51(5). – P. 1420–1435.
4. Manouselis N., Drachsler H., Verbert K., Duval E. Recommender systems for learning // *SpringerBriefs in Electrical and Computer Engineering*. – 2013. – 97 p.

## **ВЕБ-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА НАУКОМЕТРИЧНОГО АНАЛІЗУ ТА ОЦІНКИ НАУКОВИХ ЗДОБУТКІВ**

**Вівчар Каріна Сергіївна**

здобувач вищої освіти бакалаврського рівня

Науковий керівник:

**Турченко Ірина Василівна**

к. т. н., доцент

Кафедра інформаційно-обчислювальних систем і управління

Західноукраїнський національний університет, Україна

**Анотація.** Розглядається створення веб-орієнтованої інформаційної системи для наукометричного аналізу та оцінки наукової діяльності. Акцент зроблено на інтеграції даних з платформ Google Scholar і Scopus, подоланні існуючих обмежень, а також на забезпеченні зручної візуалізації та обробки наукометричної інформації.

**Ключові слова:** веб-орієнтована інформаційна система, збір даних, опрацювання даних.

У сучасному науковому середовищі, де знання, інновації та кількісні показники відіграють ключову роль у формуванні наукової політики, зростає потреба у якісних інструментах для аналізу результативності наукової діяльності. Наукометрія, як наука про кількісне вимірювання наукової продуктивності, дозволяє аналізувати динаміку розвитку окремих дослідників, установ, галузей знань, а також допомагає приймати обґрунтовані рішення щодо фінансування наукових проєктів та кадрових призначень.

Незважаючи на наявність існуючих платформ (наприклад, Scopus, Web of Science, Google Scholar), користувачі стикаються з рядом обмежень, серед яких – фрагментованість інформації, труднощі з ідентифікацією авторів, недостатні засоби візуалізації та аналітики. Враховуючи ці виклики, актуальною є розробка

веб-орієнтованої інформаційної системи, яка забезпечує інтеграцію, аналіз і зручну подачу наукометричних даних.

Тематичне поле наукометричного аналізу активно досліджується в контексті цифрових інструментів. Згідно з [1], платформи Google Scholar та Scopus мають свої переваги і недоліки: перша має більшу відкритість, але меншу точність, друга – надає перевірені дані, однак обмежена в доступі. У [2] обґрунтовано важливість створення систем з єдиним доступом до даних для ефективного наукового моніторингу. Дослідження [3] показують перспективність веб-орієнтованих систем з використанням мови Python і бібліотеки Django, Потреба в автоматизованому зборі та уніфікованій візуалізації авторських профілів обґрунтована у [4].

Розроблена веб-орієнтована інформаційна система наукометричного аналізу та оцінки наукових здобутків дозволяє отримувати, структурувати та візуалізувати наукометричні дані з різних джерел (Google Scholar, Scopus), основними завданнями було:

- реалізація серверної частини (Back-end) з використанням Python;
- побудова клієнтської частини (Front-end) з HTML, CSS, Jinja та JavaScript;
- створення бази даних (SQLite) для зберігання зібраних даних;
- формування модулів авторського профілю, пошуку авторів, сторінок.

Архітектура системи складається з:

- Back-end – модуль збору, обробки і передачі даних;
- Front-end – модуль відображення результатів, що забезпечує зручний інтерфейс для взаємодії користувача із системою.

Сторінка профілю автора (рисунок 1) містить узагальнену інформацію: ім'я, місце роботи, кількість публікацій, індекс Хірша, кількість цитувань тощо. Реалізовано механізми пошуку авторів за ім'ям, а також перегляду пов'язаних публікацій.

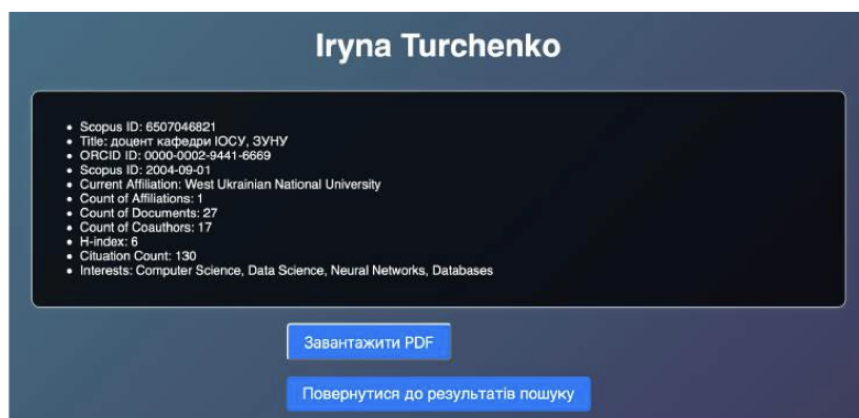


Рисунок 1 – Приклад сторінки з інформацією про автора

Система має потенціал масштабування: додавання нових джерел, реалізація експорту даних (PDF). Можлива також візуалізація наукових зв'язків (со-

authorship networks), що дозволить виявляти наукові кластери та дослідницькі колаборації.

**Висновки.** Розроблена веб-орієнтована інформаційна система є ефективним інструментом для наукометричного аналізу, що дозволяє об'єднати дані з різних платформ, надати їх у зручному візуальному форматі та здійснити аналітичну оцінку наукових здобутків. Вона має значний потенціал для впровадження в установах освіти і науки, а також подальшого розвитку в напрямку розширення джерел даних та інтелектуального аналізу.

#### Список використаних джерел

1. Бесєдіна Н. О. Наукометричні показники: переваги та недоліки використання в системі оцінки наукової діяльності // Вісник Національної академії наук України. – 2021. – № 3. – С. 42–48.
2. Поліщук В. І., Кондратенко Н. М. Інформаційні системи наукометричного моніторингу: сучасний стан та перспективи розвитку // Наукові праці НУХТ. – 2020. – № 26(3). – С. 124–130.
3. Vardi M. Y. Scientific impact and scientific productivity: A case of apples and oranges // Communications of the ACM. – 2009. – Vol. 52(6). – P. 5–5.
4. Harzing A.-W. Publish or Perish [Електронний ресурс]. – Режим доступу: <https://harzing.com/resources/publish-or-perish> (31.05.2025)
5. Scopus. What is Scopus? [Електронний ресурс]. – Режим доступу: <https://www.scopus.com> (31.05.2025)

## CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE FOR IMAGE-BASED ARCHITECTURAL STYLE RECOGNITION

**Mushtyn Oleksandr**

Bachelor student

**Sobko Olena**

Teacher

**Molchanova Maryna**

Teacher

**Mazurets Oleksandr**

Ph.D in Engineering Science, Associate Professor

Computer Science Department

Khmelnyskyi National University, Ukraine

In today's era of rapid digital development, automatic recognition of visual content is becoming increasingly important in various fields, from cultural heritage to urban planning and commercial design. Identifying architectural styles from images is

authorship networks), що дозволить виявляти наукові кластери та дослідницькі колаборації.

**Висновки.** Розроблена веб-орієнтована інформаційна система є ефективним інструментом для наукометричного аналізу, що дозволяє об'єднати дані з різних платформ, надати їх у зручному візуальному форматі та здійснити аналітичну оцінку наукових здобутків. Вона має значний потенціал для впровадження в установах освіти і науки, а також подальшого розвитку в напрямку розширення джерел даних та інтелектуального аналізу.

#### Список використаних джерел

1. Беседіна Н. О. Наукометричні показники: переваги та недоліки використання в системі оцінки наукової діяльності // Вісник Національної академії наук України. – 2021. – № 3. – С. 42–48.
2. Поліщук В. І., Кондратенко Н. М. Інформаційні системи наукометричного моніторингу: сучасний стан та перспективи розвитку // Наукові праці НУХТ. – 2020. – № 26(3). – С. 124–130.
3. Vardi M. Y. Scientific impact and scientific productivity: A case of apples and oranges // Communications of the ACM. – 2009. – Vol. 52(6). – P. 5–5.
4. Harzing A.-W. Publish or Perish [Електронний ресурс]. – Режим доступу: <https://harzing.com/resources/publish-or-perish> (31.05.2025)
5. Scopus. What is Scopus? [Електронний ресурс]. – Режим доступу: <https://www.scopus.com> (31.05.2025)

## CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE FOR IMAGE-BASED ARCHITECTURAL STYLE RECOGNITION

**Mushtyn Oleksandr**

Bachelor student

**Sobko Olena**

Teacher

**Molchanova Maryna**

Teacher

**Mazurets Oleksandr**

Ph.D in Engineering Science, Associate Professor

Computer Science Department

Khmelnytskyi National University, Ukraine

In today's era of rapid digital development, automatic recognition of visual content is becoming increasingly important in various fields, from cultural heritage to urban planning and commercial design. Identifying architectural styles from images is