

НУБІП України

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НУБІП України

ГРЯЗЮКА ВЯЧЕСЛАВА ОЛЕКСАНДРОВИЧА

2022 р.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

НУБІП України

Факультет інформаційних технологій

НУБІП України

УДК
«ПОГОДЖЕНО»
Декан факультету

інформаційних технологій

«ДОПУСКАЄТЬСЯ ДО
ЗАХИСТУ»
Завідувач кафедри комп'ютерних наук

НУБІП України

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

2022 р.

2022 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НУБІП України

на тему «Експертна система оцінки наукової діяльності педагогічного складу
за інформацією з міжнародних джерел»

Спеціальність 122 «Комп'ютерні науки»

Освітня програма «Інформаційні управляючі системи та технології»

Орієнтація освітньої програми: освітньо-професійна

НУБІП України

Гарант освітньої програми
к.е.н., доцент

Густера О.М.

Керівник магістерської кваліфікаційної роботи

доктор фізико-математичних наук, професор

Малашонок Г.І.

НУБІП України

Виконав

Грязюк В. О.

КИЇВ-2022

НУБІП України

НУБІП України

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

НУБІП України

к.т.н., доцент _____ Голуб Б.Л.

“01” листопада 2021 року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Грязюку Вячеславу Олександровичу
(прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні науки»

Освітня програма «Інформаційні управлючі системи та технології»

Орієнтація освітньої програми: освітньо-професійна

Тема магістерської кваліфікаційної роботи «Експертна система оцінки наукової діяльності педагогічного складу за інформацією з міжнародних джерел»
затверджена наказом ректора НУБіП України від “01” листопада 2021 р. №1862 «С»

НУБІП України

Термін подання завершеної роботи на кафедру 27 жовтня 2022 р.

Вихідні дані до магістерської кваліфікаційної роботи:

1. дані про схожість робіт викладачів до їх напрямку діяльності;
2. дані про кількість робіт викладачів за певний період часу.

Перелік питань, що підлягають дослідженню:

1. аналіз подібних систем;
2. налаштування парсингу даних;
3. проектування системи;
4. аналіз методів порівняння слів;
5. розробка алгоритму оцінки діяльності викладачів;
6. дослідження отриманих результатів і висновків.

НУБІП України

Дата видачі завдання “01” листопада 2021 р.

Керівник магістерської кваліфікаційної роботи

(підпис)

Малашенок Т.І.

(прізвище та ініціали)

Завдання прийняв до виконання

(підпис)

Грязюк В.О.

(прізвище та ініціали студента)

НУБІП України

НУБІП України

ЗМІСТ

| | | |
|---------------|---|----|
| НУБІП України | Перелік умовних позначень..... | 4 |
| НУБІП України | Вступ..... | 5 |
| НУБІП України | 1 Системний аналіз програми, що здійснює оцінку наукової діяльності викладачів..... | 7 |
| НУБІП України | 1.1 Завдання розробки системи..... | 7 |
| НУБІП України | 1.2 Існуючі системи аналізу та оцінки діяльності викладачів..... | 8 |
| НУБІП України | 2 Моделювання системи..... | 11 |
| НУБІП України | 2.1 Опис реалізації БД..... | 12 |
| НУБІП України | 2.2 Опис Use Case діаграми..... | 16 |
| НУБІП України | 2.3 Функціональна схема..... | 19 |
| НУБІП України | 3 Розробка системи..... | 20 |
| НУБІП України | 3.1 Система управління базами даних MySQL..... | 20 |
| НУБІП України | 3.2 Мова програмування Python..... | 23 |
| НУБІП України | 3.3 Інтегроване середовище розробки PyCharm..... | 25 |
| НУБІП України | 3.4 Середовище для розробки графічних інтерфейсів Qt Designer.... | 26 |
| НУБІП України | 3.5 Бібліографічне джерело даних core.ac.uk..... | 28 |
| НУБІП України | 3.5 Парсинг даних за допомогою Selenium WebDriver..... | 30 |
| НУБІП України | 3.6 Алгоритм аналізу та оцінки діяльності викладачів, використовуючи метрику Подібність Джаро – Вінклера..... | 33 |
| НУБІП України | 4 Результати дослідження..... | 42 |
| НУБІП України | 4.1 Системні вимоги та встановлення програмного продукту..... | 42 |
| НУБІП України | 4.2 Робота з програмним продуктом..... | 44 |
| НУБІП України | 4.3 Експеримент у реальному часі..... | 58 |
| НУБІП України | Висновки..... | 61 |
| НУБІП України | Список використаних джерел..... | 63 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

НУБІП України

IDE – інтегроване середовище розробки.

PIP - інсталятор пакетів для Python.

НУБІП України

БД – база даних.

ОС – операційна система.

СУБД – система управління базами даних.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВСТУП

НУБІП України

Більше 75% громадян України здобувають вищу освіту. Постає питання, чи всі ці люди в закладах освіти отримують саме ту інформацію, яку потрібно?

НУБІП України

Важливу роль в цьому процесі відіграють викладачі, які навчають студентів. Тому важливо їх перевіряти.

НУБІП України

Існують різні методи оцінки діяльності викладачів. Для високого результату було вирішено спроектувати систему, яка зможе знаходити, збирати, аналізувати та оцінювати наукові праці одного викладача або й навіть групи викладачів однієї спеціальності чи кафедри. Програма буде збирати дані з робіт викладача чи групи викладачів і потім зможе провести оцінку їх діяльності, порівнявши ключові слова робіт з ключовими словами напрямку діяльності, якою займається викладач чи група викладачів.

НУБІП України

По-перше, це дасть змогу побачити кількість робіт викладачів по роках і зрозуміти активність появи нових робіт.

По-друге, можна оцінити діяльність викладача або групи викладачів і навіть створити рейтинг, щоб побачити, яка спеціальність чи кафедра мають кращі показники серед викладачів.

НУБІП України

Дана програма допоможе університетам. Можливо, певний викладач не компетентний на тій спеціальності, на яку його призначили і потрібно це змінити. Система ознайомить користувача з роботами цього викладача і надасть можливість перевірити, на яку спеціальність чи спеціалізацію він більше підходить.

НУБІП України

Дана система являє собою програмний продукт, який має функціональний і зручний інтерфейс. Користувачеві достатньо зробити декілька дій, щоб програма здійснила швидкий пошук даних про викладачів в інтернеті і відобразила їх на екрані. Пошук інформації відбувається методом парсингу з міжнародного бібліографічного сайту CORE. Програма має двоступню обробку даних. Спочатку перевіряє інформацію в базі даних і якщо такої немає, то

починає збирати з інтернету. Потім, використовуючи ці дані, проводить оцінку діяльності викладачів.

Завдання:

- провести аналіз подібних систем;
- налаштувати парсинг даних;
- розробити БД;
- провести аналіз методів порівняння слів;
- розробити алгоритм оцінки діяльності викладачів;
- сформулювати отримані результати і висновки.

Структура роботи: 64 сторінок, 20 використаних джерел; 4 розділи, кожен із них має як мінімум 2 підрозділи.

1 СИСТЕМНИЙ АНАЛІЗ ПРОГРАМИ, ЩО ЗДІЙСНЮЄ ОЦІНКУ НАУКОВОЇ ДІЯЛЬНОСТІ ВИКЛАДАЧІВ

Перед тим, як почати розробляти систему, потрібно повністю визначити, якою буде система, які функції буде підтримувати і що в результаті повинно вийти. Також потрібно проаналізувати уже існуючі схожі системи і порівняти з тією, яка розробляється.

1.1 Завдання розробки системи

Метою роботи є створення зручної системи для оцінки діяльності викладачів різних університетів України.

Одним із завдань даної роботи являється розробка алгоритму оцінки діяльності викладача. Для цього потрібно проаналізувати різні методи, які можуть давати оцінку схожості двох слів і обрати найбільш ефективніший для даного проекту.

Система повинна мати простий інтерфейс, щоб користувач легко міг працювати з програмою, але в той час і функціональний. Щоб почати працювати з програмний продуктом, користувач не повинен авторизуватися. Для пошуку робіт і аналізу діяльності викладача достатньо ввести ПІБ викладача чи групи викладачів, роки, за якими буде проводитися аналіз і ключові слова напрямку діяльності.

Також для того, щоб збільшити ефективність роботи даної програми потрібно знаходити роботи викладачів на англійській мові. Для цього потрібно додати поле вводу ПІБ викладача англійською мовою.

Система всю інформацію буде зберігати в БД, перед цим зібравши з міжнародного бібліографічного джерела. Якщо дані викладача уже є в базі даних, то програма про це повідомить. По бажанню користувача ці дані можна оновити.

Програмний продукт повинен мати 2 способи, як проаналізувати викладачів:

- по кількості робіт (хто має більше робіт за певний період часу);
- по ключовим словам (схожість ключових слів діяльності викладача з ключовими словами або абстрактном його наукових робіт).

1.2 Існуючі системи аналізу та оцінки діяльності викладачів

У процесі пошуку існуючих аналогів систем, що вирішують ідентичну задачу, не знайдено жодного. Однак були знайдені подібні системи управління бібліографічною інформацією. Це такі системи:

- Google Scholar
- Scopus
- arXiv.org

Розглянемо кожну із них детальніше.

Google Scholar – це бібліографічна база даних, що містить багато різноманітних наукових публікацій усіх форматів [1]. До неї входять найбільші наукові статті Америки та Європи. Google Scholar надає можливість користувачам знаходити цифрові або фізичні копії статей, як в Інтернеті, так і в бібліотеках. Однак більшість людей матимуть доступ лише до анотації статті та цитат, оскільки Google Scholar більше посилається на комерційні журнали, ніж на ті, які є у вільному доступі.

Але ця бібліографічна база даних має також і недолік. Вона зберігає в собі велику кількість псевдонаукових публікацій, і через це під час пошуку необхідної інформації можна витратити багато часу на вивчення праць, які не становлять наукової цінності.

Scopus – одна з найбільших бібліографічних і реферативних баз даних, яка має понад 50 мільйонів реферативних записів і 51 000 наукових публікацій [2]. Користувачі Scopus можуть здійснювати пошук за ключовими словами, фразами, назвою статті чи журналу. Видані системою результати фільтруються за роком

публікації, темою, афіліацією, типом документів. Також у базу даних вбудовано інструменти відстеження, аналізу та візуалізації даних [3].

Бібліометричні дані зі Scopus використовують для складання рейтингів університетів. Також на базу даних спираються безліч аналітичних публікацій, які спонсоруються університетами, урядовими та міжнародними організаціями.

Метрику даної системи часто використовують для оцінки якості досліджень, які проходять у рамках національних грантів. У березні 2022 року через вторгнення російських військ в Україну Scopus оголосили про закриття доступу російським і білоруським науковим організаціям. Доступ для російських користувачів був остаточно заблокований на початку травня 2022 року.

arXiv.org – електронний архів з відкритим доступом для наукових статей та препринтів з фізики, математики, астрономії, інформатики, біології, електротехніки, статистики, фінансової математики та економіки [4]. У даній базі часто публікуються роботи, що істотно впливають на розвиток науки. За даними порталу, станом на жовтень 2022 року на сайт arXiv.org було завантажено більше двох мільйонів робіт (рисунок 1.1).

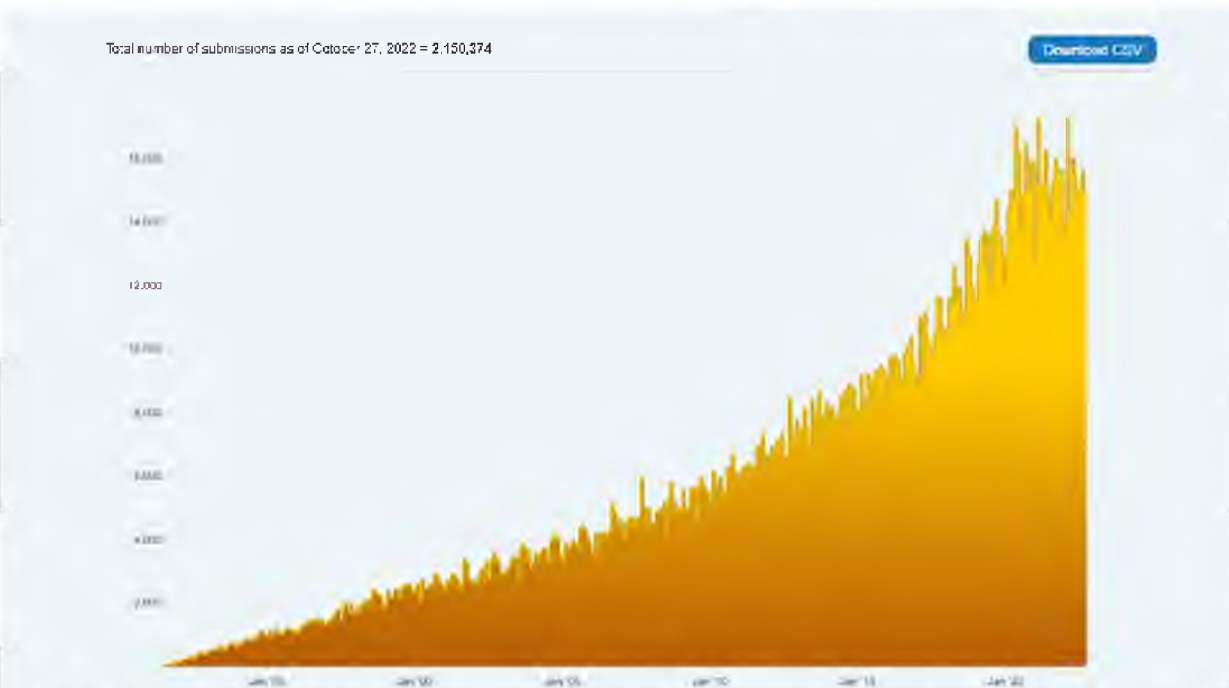


Рисунок 1.1. Кількість наукових робіт на сайті arXiv.org

Дана система також представлена у вигляді додатку на мобільні пристрої. Через нього користувачі можуть здійснювати пошук за каталогом через ключові слова або фільтри за заголовками, автором, анотаціями та цифровим ідентифікатором.

Окремі дослідники зазначають, що, незважаючи на свої переваги, arXiv не є ідеальною системою поширення наукових даних. Проблемою є відсутність системи рецензування – модератори не оцінюють завантажені матеріали з погляду достовірності чи наукової значущості.

Якщо порівняти систему дипломного проекту і перелічені вище аналоги, можна помітити деякі відмінності. Вищевказані системи можуть відображати наукові роботи, реферати, статті різних викладачів за певний період часу. Розроблена мною система відрізняється тим, що вона збирає інформацію про наукові роботи певного викладача за деякий проміжок часу з бібліографічного джерел CORE, фільтрує дані, зберігає їх у базі даних. І використовуючи цю інформацію за допомогою спеціального алгоритму оцінює діяльність викладача чи групи викладачів за ключовими словами або за кількістю робіт. Це допомагає перевірити ефективність роботи всієї кафедри, спеціальності чи окремого викладача.

Бібліографічна база даних CORE збирає інформацію з журналів відкритого доступу, а також дослідницькі статті з інституційних і тематичних репозиторіїв [5]. Зараз CORE містить понад 245 мільйонів статей у відкритому доступі. Мета даної бази – це об'єднати всі дослідження відкритого доступу в усьому світі та забезпечити необмежений доступ для всіх.

Сайт має дуже зручний інтерфейс, можна знайти повну інформацію про роботи викладачів. Це допоможе швидко методом парсингу зібрати дані, зберегти в БД, а потім провести аналіз і оцінку діяльності викладачів. Доступ до даного джерела є безкоштовним.

2. МОДЕЛЮВАННЯ СИСТЕМИ

Для того, щоб десь зберігати дані система має бути підключена до реляційної БД. База даних повинна складатися з таблиць, які будуть містити інформацію про викладачів англійською та українською мовами, їх роботи за роками, ключові слова та відсоток збігу наукових праць викладача до напрямку його діяльності. Поля в таблицях повинні мати достатні розміри, щоб вся інформація зберігалася коректно. Також взяти до уваги можливість вводити назву кафедри чи спеціальності, коли потрібно порівняти групу викладачів. БД має бути простою у використанні та швидко обробляти запити. Усі таблиці мають бути логічно пов'язані.

Для збору інформації з міжнародного джерела необхідно скористатися парсингом даних. Цей метод має бути простим у реалізації, швидко знаходити та збирати ту інформацію, яка потрібна для оцінки діяльності викладачів.

Для оцінки діяльності викладача чи групи викладачів необхідно використати одну із метрик подібності слів для порівняння ключового слова з одним із слів абстракт роботи. І маючи ці дані розробити алгоритм, який в результаті дасть оцінку схожості наукових робіт викладача до напрямку діяльності у вигляді відсотків.

Інтерфейс системи повинен відповідати сучасним стандартам. Система не повинна бути перевантажена функціями, які не є суттєвими для вирішення завдань аналізу та оцінки діяльності викладачів.

Усі елементи програми повинні бути зручно розташовані користувачеві повинно бути легко орієнтуватися в програмі. Текст повинен бути чітким і читабельним, а навігація між вікнами не повинна заплутати користувача. Якщо під час роботи виникають будь-які помилки, то програма повинна повідомляти про це користувача з відповідним повідомленням. Усі необхідні результати користувач має бачити у вигляді зрозумілих таблиць.

Перед тим, як перейти до розробки програмного продукту проведемо моделювання системи, детально описавши базу даних, функціональну модель та діаграму варіантів використання.

2.1 Опис реалізації БД

У базі даних дипломного проєкту реалізовано 5 таблиць: викладач, робота, кількість робіт, група викладачів та викладач в групі. Кожна таблиця грає важливу роль у роботі програми.

У таблиці “викладач” (рисунок 2.1) будуть зберігатися дані про окремого викладача, які надав користувач програми.

Таблиця має такі поля:

- id викладача (номер автора, у кожного він унікальний);
- прізвище викладача;
- імя викладача;
- по_батькові викладача;
- прізвище викладача англійською мовою;
- імя викладача англійською мовою;
- по_батькові викладача англійською мовою.

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------------------|-------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|
| id_автора | INT(10) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| прізвище_автора | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| імя_автора | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| по_батькові_автора | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| прізвище_автора_англ | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| імя_автора_англ | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| по_батькові_автора_англ | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

Рисунок 2.1. Таблиця “викладач”

У таблиці “робота” (рисунок 2.2) буде знаходитись інформація про окрему наукову роботу певного викладача, яка була знайдена в базі бібліографічного джерела CORE.

У цій таблиці наявні такі поля:

- id роботи (номер наукової роботи, він унікальний);
- id_автора (номер автора);
- абстракт (короткий опис, який присутній майже в усіх наукових роботах);
- назва роботи;
- автори (перелік авторів даної роботи);
- рік публікації.

Дана таблиця має зв'язок з таблицею “викладач” по id_автора

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|----------------|---------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|
| id_роботи | INT(10) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| id_автора | INT(10) | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| абстракт | VARCHAR(1010) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| назва_роботи | VARCHAR(1000) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| автори | VARCHAR(300) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| рік_публікації | INT(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

Рисунок 2.2. Таблиця “робота”

Таблиця “кількість робіт” (рисунок 2.3) буде зберігати дані про кількість робіт певного автора за кожен рік. Ця інформація потрібна для того, щоб програма швидко відобразила дані і аналітик міг проаналізувати викладача на наявність робіт протягом певних років.

Ця таблиця має наступні поля:

- id_автора;
- рік;
- кількість робіт за рік.

Дана таблиця має зв'язок з таблицею “викладач” по id автора

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|------------------|----------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| id_автора | INT(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| рік | INT(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| кількість_за_рік | INT(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

Рисунок 2.3. Таблиця “кількість робіт”

У таблиці “група викладачів” (рисунок 2.4) буде інформація про викладачів, які знаходяться в одній групі і додаткова інформація, яка потрібна для оцінки діяльності викладачів.

Таблиця має такі поля:

- id групи (номер групи, він унікальний);
- назва групи;
- викладачі (перелік викладачів, які відносяться до однієї групи, в групі може бути один викладач);
- ключові слова (ключові слова діяльності, якою займається група);
- роки (в даного проміжку часу беруться наукові роботи викладачів, які потім будуть проходити аналіз);
- кількість робіт (кількість робіт усіх викладачів за певний проміжок часу);
- відсоток схожості (відсоток, який демонструє наскільки наукові роботи викладачів схожі до їх напрямку діяльності).

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------------|---------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|
| id_групи | INT(10) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| назва_групи | VARCHAR(1000) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| викладачі | VARCHAR(1000) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ключові_слова | VARCHAR(500) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| роки | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| кількість_робіт | INT(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| відсоток_схожості | FLOAT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

Рисунок 2.4. Таблиця “група викладачів”

І в останній проміжній таблиці “викладач в групі” (рисунк 2.5) буде знаходитися два поля:

- id автора;
- id групи.

Дана таблиця має зв’язок з таблицями “викладач” та “група викладачів” по id автора та id групи відповідно.

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------|----------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| id_автора | INT(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| id_групи | INT(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

Рисунок 2.5. Таблиця “викладач в групі”

Всі таблиці даної БД з’єднані між собою та мають зв’язок один до багатьох. Концептуальна модель має такий вигляд (рисунк 2.6).

НУБІП України

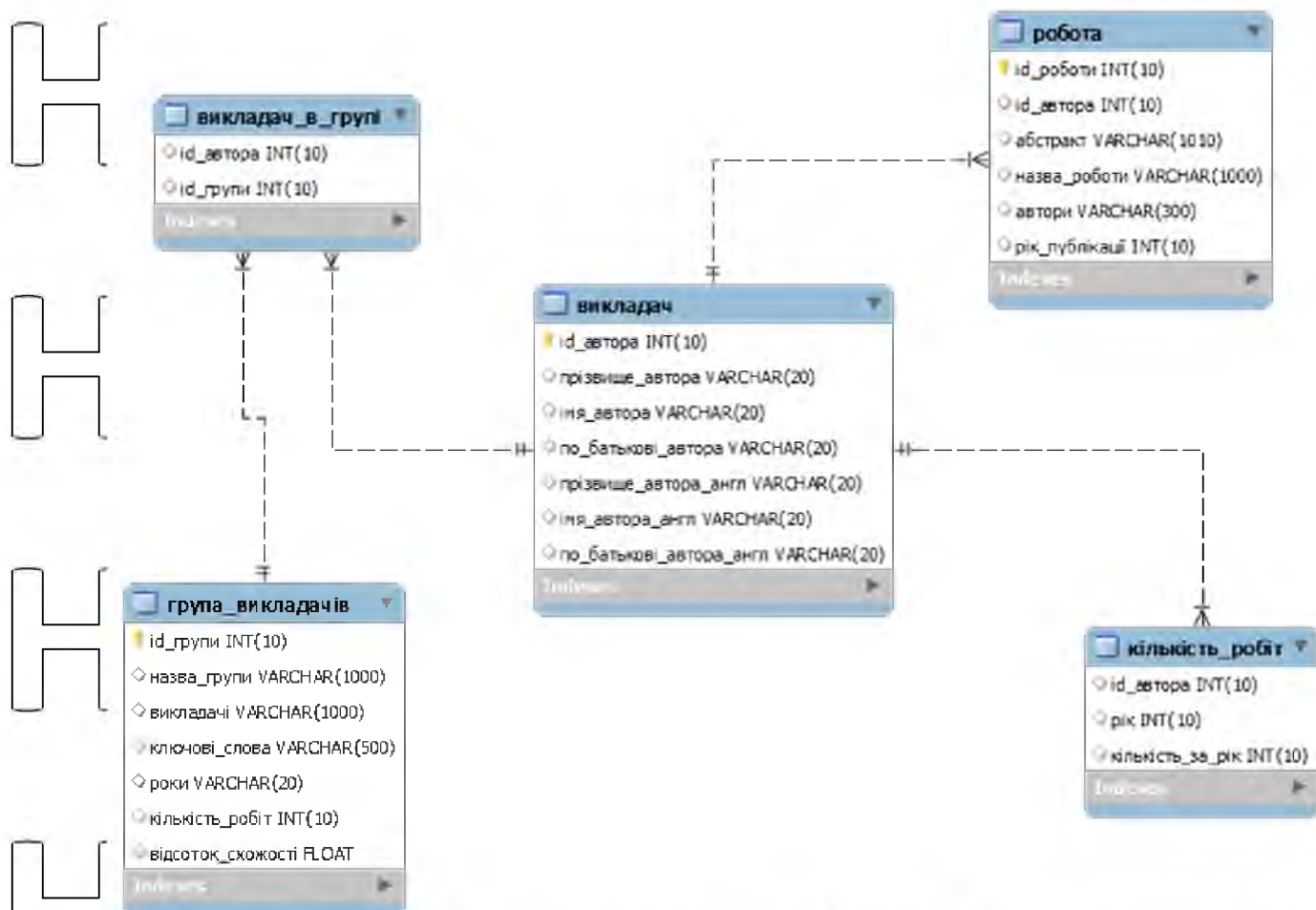


Рисунок 2.6. Концептуальна модель БД

Концептуальну модель даних зазвичай використовують для опису того, як зберігаються дані (запам'ятовуючому середовищі [6]). Дана модель містить усі деталі, необхідні конкретній СУБД для створення бази даних: імена таблиць, стовпців, визначення первинних і зовнішніх ключів, типи даних.

2.2. Опис Use Case діаграми

Перед тим, як перейти до побудови UML-діаграми варіантів використання визначимо акторів системи. Дана система має двох акторів: менеджер і аналітик.

Опишемо можливості кожного із них.

Обоє акторів мають вільний доступ до додатку, достатньо просто відкрити програму і почати з нею працювати.

Для менеджера присутні такі функції:

- створення запиту на отримання інформації про викладача по ПІБ;
- додавання нових викладачів в БД;
- додавання до уже існуючого викладача в базі даних ПІБ викладача англійською мовою;
- перегляд інформації про окремого викладача і його роботи за певний проміжок часу;
- перегляд аналітичної інформації у вигляді відсотку схожості викладача або групи викладачів до їх напрямку діяльності.

Для аналітика розроблені такі функції:

- ввід ключових слів, які відносяться до певної спеціальності;
- аналіз групи викладачів або окремих осіб на активність появи нових робіт протягом заданих років;
- оцінка групи викладачів або окремих осіб по наукових працях та ключовими словами їх діяльності;
- можливість дозбирати інформацію про викладачів, якщо її не вистачає для аналізу;
- перегляд аналітичної інформації у вигляді відсотку схожості викладача або групи викладачів до їх напрямку діяльності, а також дані про окремого викладача і його наукові роботи.

Тепер, коли описані функції кожного актора, побудуємо Use Case діаграму

(рисунок 2.7).



Рисунок 2.7. UML-діаграма варіантів використання

Діаграма варіантів використання є основною формою вимог до системи/програмного забезпечення для нової програми [7]. Ключова концепція моделювання варіантів використання полягає в тому, що воно допомагає проектувати систему з точки зору кінцевого користувача. Ця діаграма показує зв'язки між акторами та прецедентами. Це допомагає перед розробкою програмного продукту зрозуміти, для кого ця система буде створена і які функції потрібно реалізувати.

UML-діаграма варіантів використання зазвичай проста.

2.3. Функціональна схема

Функціональна схема роз'яснює певні процеси, що відбуваються у певний проміжок часу. За допомогою цієї схеми можна прослідкувати поетапний процес роботи програми (рисунок 2.8).

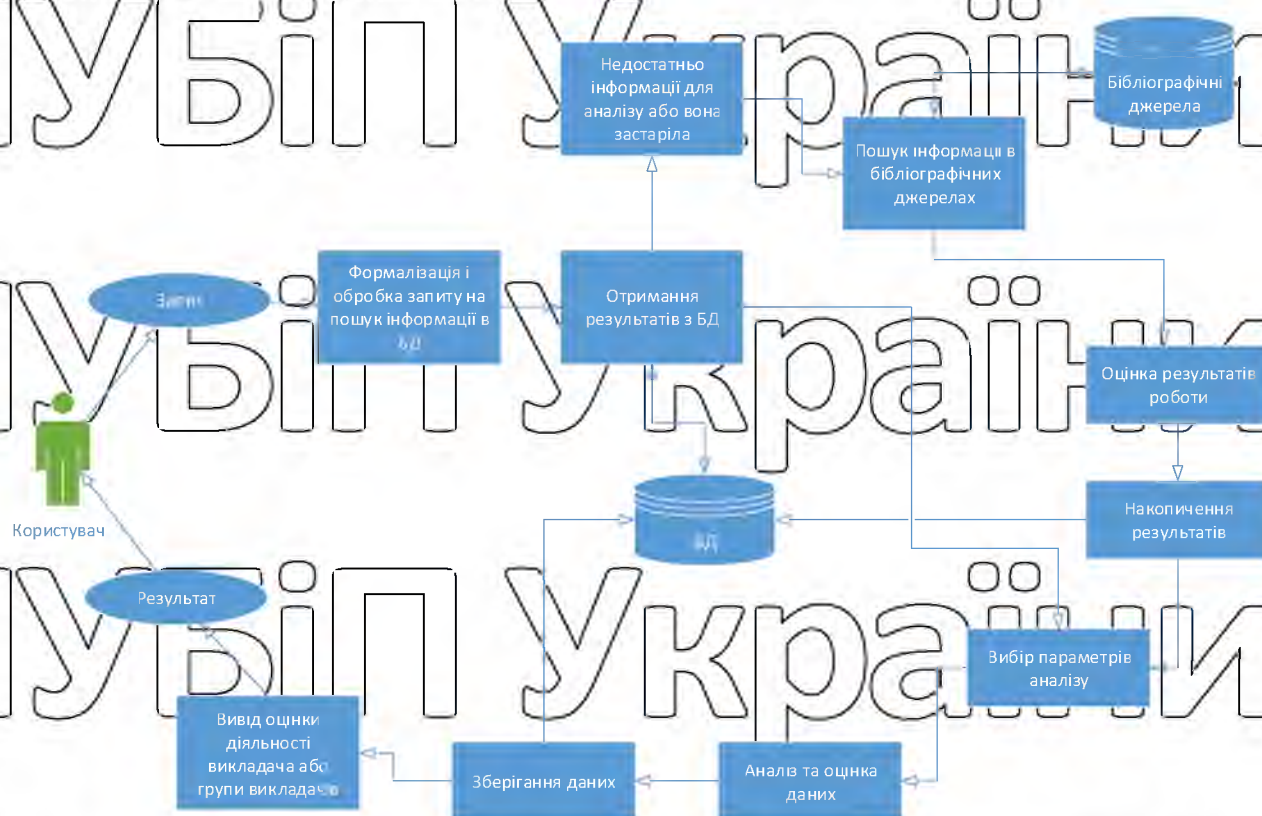


Рисунок 2.8. Функціональна схема збору та аналізу інформації про викладачів

Дана схема спроектована на прикладі одного користувача, який являється і менеджером, і аналітиком. Можна побачити, що програма має двоетапну вибірку інформації. Перед тим як збирати дані з бібліографічного джерела, система перевіряє дані викладачів в БД, якщо така інформація відсутня або її бракує, то здійснюється парсинг даних з сайту cite.ac.uk. Дані зберігаються в БД і потім ця інформація вже використовується для оцінки діяльності викладачів.

Послідовність та логічність операцій є перевагою даної схеми.

3 РОЗРОБКА СИСТЕМИ

Для розробки даної програми було вирішено використати реляційну базу даних MySQL для того, щоб зберігати всю потрібну інформацію. Програмний продукт буде являти собою додаток, який можна запускати на комп'ютері. Для цього було використано середовище PyCharm, а код був написаний на мові програмування Python. За допомогою фреймворку Qt був розроблений дизайн програми. Бібліотеку Selenium WebDriver було вирішено використати для парсингу даних з бібліографічного джерела CORE.

Для оцінки діяльності викладачів був розроблений окремий алгоритм, використовуючи рядкову метрику Подібність Джаро – Вінклера

3.1 Система управління базами даних MySQL

MySQL – вільна реляційна система управління базами даних [8]. Вона являється найпопулярнішою системою з відкритим кодом. Розроблена, розповсюджена та підтримується корпорацією Oracle.

Реляційна база даних зберігає дані в окремих таблицях, а не поміщає всі дані в одну велику комору. Структури бази даних організовано у фізичні файли, оптимізовані для швидкості. Логічна модель із такими об'єктами, як бази даних, таблиці, подання, рядки та стовпці, пропонує гнучке середовище програмування.

Плюсом даної бази являється встановлення правил, що регулюють зв'язки між різними полями даних, наприклад «один-до-одного», «один-до-багатьох», унікальні, обов'язкові чи необов'язкові, а також «вказівники» між різними таблицями. База даних забезпечує дотримання цих правил, а це означає, що з добре спроектованою базою даних ніколи не буде дублікатів, неузгоджених, застарілих або відсутніх даних.

SQL є найпоширенішою стандартизованою мовою, яка використовується для доступу до баз даних. Сервер бази даних MySQL дуже швидкий, надійний, масштабований і простий у використанні.

Окрім універсальності та поширеності, MySQL має низку важливих переваг перед іншими системами [9].

Переваги СУБД MySQL:

- Широкий функціонал. У MySQL є майже всі необхідні інструменти, які можуть бути потрібними для реалізації практично будь-якого

проекту.

- Простота використання. Дана СУБД досить проста в установці, є багато плагінів і утиліт, які полегшують роботу з даними.

- Швидкість. Висока продуктивність системи забезпечується за рахунок спрощення деяких стандартів, які в ній використовуються.

- Безпека. Вбудовані функції безпеки працюють за замовчуванням.

- Масштабованість. MySQL — це універсальна база даних, яку можна однаково легко використовувати для роботи як з великими, так і з малими обсягами даних.

- Основний потік, що використовується, є повністю багатопоточним і підтримує кілька процесорів.

- MySQL може працювати на різних платформах. Підтримка C, C++, Java, Perl, PHP, Python та TCL API.

Також дана СУБД має певні недоліки, які впливають на розробку програм, які мають свої вимоги.

До недоліків належать:

- MySQL не має мови процедур, які зберігаються. Це є найбільшим обмеженням для програмістів, що звикли до баз даних корпоративного рівня.

- MySQL не підтримує гаряче резервне копіювання.

Коли йдеться про надійність деяких процесів обробки даних (наприклад, зв'язок, аудит), MySQL іноді поступається іншим базам даних.

Існує не одна програма, яка підтримує створення запитів за допомогою MySQL. Одна із них – це MySQL Workbench 8.0 CE. Саме цю програму вирішено використовувати для роботи з базою даних дипломного проекту.

MySQL Workbench – це уніфікований візуальний інструмент для архітекторів, розробників і адміністраторів баз даних. MySQL Workbench забезпечує моделювання даних, розробку SQL і комплексні засоби адміністрування (для конфігурації сервера, адміністрування користувачів, резервного копіювання та багато іншого). MySQL Workbench доступний у Windows, Linux і Mac OS X.

Для початку роботи потрібно скачати і встановити програму, потім при запуску з'явиться вікно, де необхідно приєднатися до сервера (рисунк 3.1)

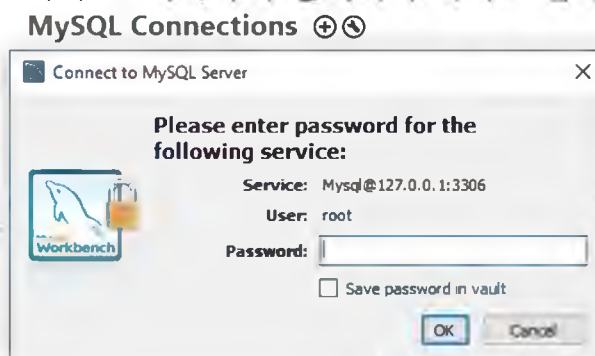


Рисунок 3.1. Під'єднання до MySQL Server.

Після цього буде відображене вікно, де вже можна розробляти запити для створення бази даних і таблиць (рисунк 3.2)

```

1 create database publications10;
2 use publications10;
3 create table викладач (
4     id_автора int unsigned auto_increment primary key,
5     прізвище_автора varchar(20),
6     імя_автора varchar(20),
7     по_батькові_автора varchar(20),
8     прізвище_автора_англ varchar(20),
9     імя_автора_англ varchar(20),
10    по_батькові_автора_англ varchar(20)
11 );
12
13 create table робота (
14     id_роботи int unsigned auto_increment primary key,
15     id_автора int unsigned,
16     абстракт varchar(1010),

```

Рисунок 3.2. Створення бази даних і таблиць в програмі MySQL Workbench

3.2 Мова програмування Python

Існують різні рейтинги з відображенням найпопулярніших мов програмування за 2022 рік. В більшості випадків мова програмування Python входить в топ-5 найбільш затребуваних мов.

Python - високорівнева мова програмування загального призначення з динамічно строгою типізацією та автоматичним управлінням пам'яттю, що орієнтована на підвищення продуктивності розробника, читання коду та його якості [10].

Мова повністю об'єктно-орієнтована в тому сенсі, що все є об'єктами [11]. Незвичайною особливістю мови є виділення блоків коду відступами з пробілів. Синтаксис базової мови мінімалістичний, через що на практиці рідко виникає потреба звертатися до документації. Сама мова відома як інтерпретована і використовується, серед іншого, для написання скриптів. Недоліками мови вважається низька швидкість і більш високе споживання пам'яті програмами, що написані на ньому, порівняно з подібним кодом, написаним на скомпільованих мовах, таких як C або C++ [12].

Стандартна бібліотека є однією з найпривабливіших сторін Python. Вона має засоби для роботи з багатьма мережевими протоколами та форматами інтернету, наприклад, модулі для написання HTTP-серверів та клієнтів, для розбирання та створення поштових повідомлень, для роботи з XML тощо.

Великим плюсом даної мови програмування являється те, що є дуже багато різних додаткових бібліотек, які можна підключити і тим самим прискорити процес написання програмного коду.

Вибір мови програмування зазвичай залежить від розв'язуваних завдань, особливостей мов та наявності бібліотек, необхідних для вирішення тих чи інших завдань. Одне й те завдання, написане різними мовами може сильно відрізнятися за ефективністю виконання, зокрема розбіжності можуть бути і під час виконання у різних операційних системах, чи за використання різних компіляторів.

Переваги мови програмування Python:

- Зрозумілість коду. Синтаксична риса Python – виділення блоків коду відступами, що значно спрощує візуальне сприйняття програм, написаних цією мовою.
- Інтерпретованість. Програми, написані мовою програмування Python, не переводяться в машинний код, а одразу виконуються програмою-інтерпретатором. Це дозволяє запускати код на будь-якій платформі із встановленим заздалегідь інтерпретатором.
- Об'єктно-орієнтовність. Python — це мова, створена згідно з парадигмою об'єктно-орієнтованого програмування (ООП). У ній основними є поняття об'єкта та класу. Класи – це спеціальні типи даних, об'єкти – екземпляри класів. У Python ви можете не тільки використовувати вже існуючі класи, але й створювати власні.
- Динамічна типізація. На відміну від C-подібних мов програмування, у Python змінні зв'язуються з типом у момент привласнення в них конкретних значень.

Цю мову програмування було обрано, тому що вона виділяється легкістю у написанні коду. Якщо потрібно буде розширити програмний продукт додатковими функціями, то майже кожному програмісту не складно буде розібратися в коді, який уже написаний. А також для Python є окремі бібліотеки, які допоможуть розробити функціональний інтерфейс з гарним дизайном.

3.3 Інтегроване середовище розробки PyCharm

PyCharm - це IDE (інтегроване середовище розробки) для мови програмування Python, розроблена компанією JetBrains. Для того, що безкоштовно працювати з даною програмою потрібно встановити PyCharm Community Edition [13]. Цієї версії буде достатньо для того, щоб розробити дипломний проект.

Основні можливості PyCharm:

- Функціональний редактор коду з підсвічуванням синтаксису, автоформатуванням та авто-відступами для мов, що підтримуються.
- Проста та зручна навігація в коді.
- Допомога при написанні коду, що включає автодоповнення, автоімпорт, шаблони коду і багато чого іншого.
- Швидкий перегляд документації прямо у вікні редактора, можливість через браузер переглядати зовнішню документацію.
- Потужний рефакторинг коду, який надає широкі можливості для виконання швидких глобальних змін у проекті.
- Повна підтримка нових версій Django фреймворку.
- Підтримка Flask фреймворку та мов Make та Jinja2.
- Величезна колекція плагінів, що постійно поповнюється.

Головні можливості, які надає дане середовище: підвищення якості коду, економія часу та все необхідне для продуктивної розробки на Python.

Після встановлення даної IDE, достатньо відкрити програму, створити проект і почати писати код (рисунок 3.3).

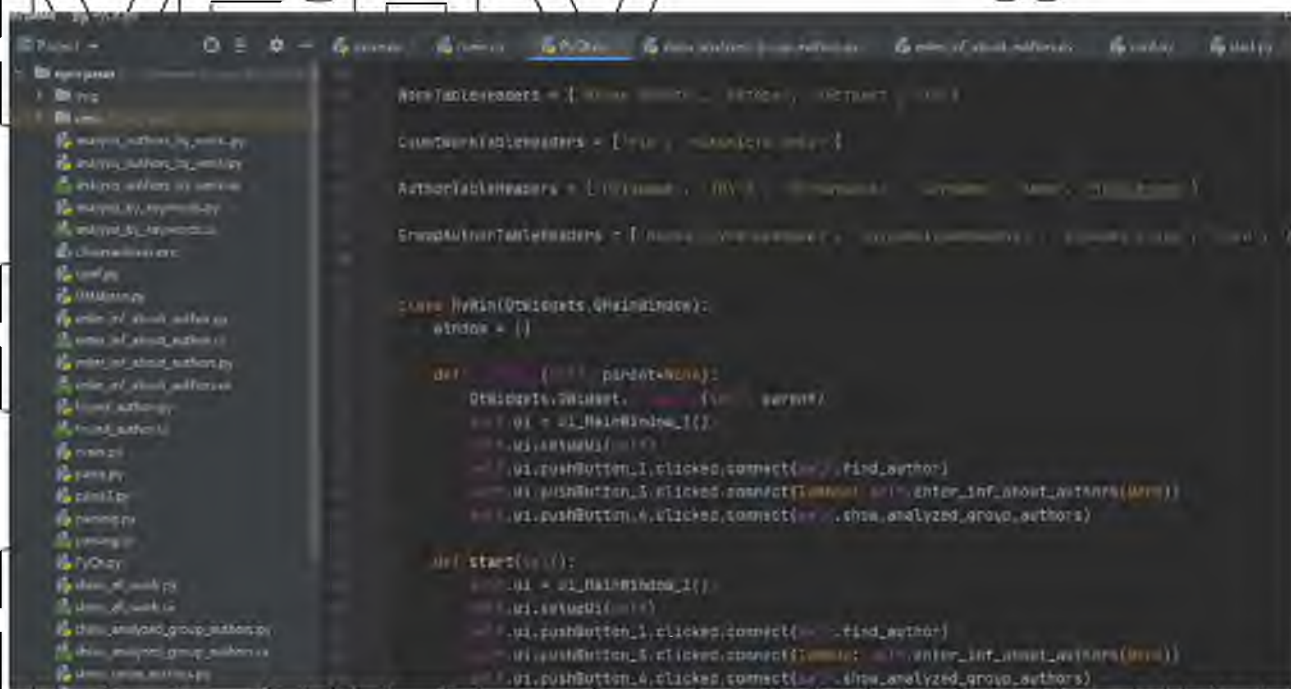


Рисунок 3.3. Робота з програмою PyCharm

Однією з найкращих інтегрованих середовищ розробки для Python являється PyCharm. Тому для розробки даного додатку краще використати саме це IDE.

3.4 Середовище для розробки графічних інтерфейсів Qt Designer

Qt Designer – вільне середовище для розробки графічних інтерфейсів (GUI) для програм, що використовують бібліотеку Qt. Входить до складу Qt framework.

Qt Designer дозволяє створювати графічні інтерфейси користувача за допомогою ряду інструментів. Існує панель інструментів в якій доступні для використання елементи інтерфейсу – віджети ("список, що випадає", "поле введення", "кнопка" і багато інших). Для кожного віджета присутній свій набір властивостей, що визначається відповідним класом бібліотеки Qt. Властивості віджету можуть бути змінені

За допомогою даного середовища можна розробити функціональний додаток з гарним дизайном, що являється одним із завдань даного проєкту. Потрібно створити окремо кожне вікно, яке буде відображатися в фінальному програмному продукті. Приклад роботи в програмі зображений на рисунку 3.4.

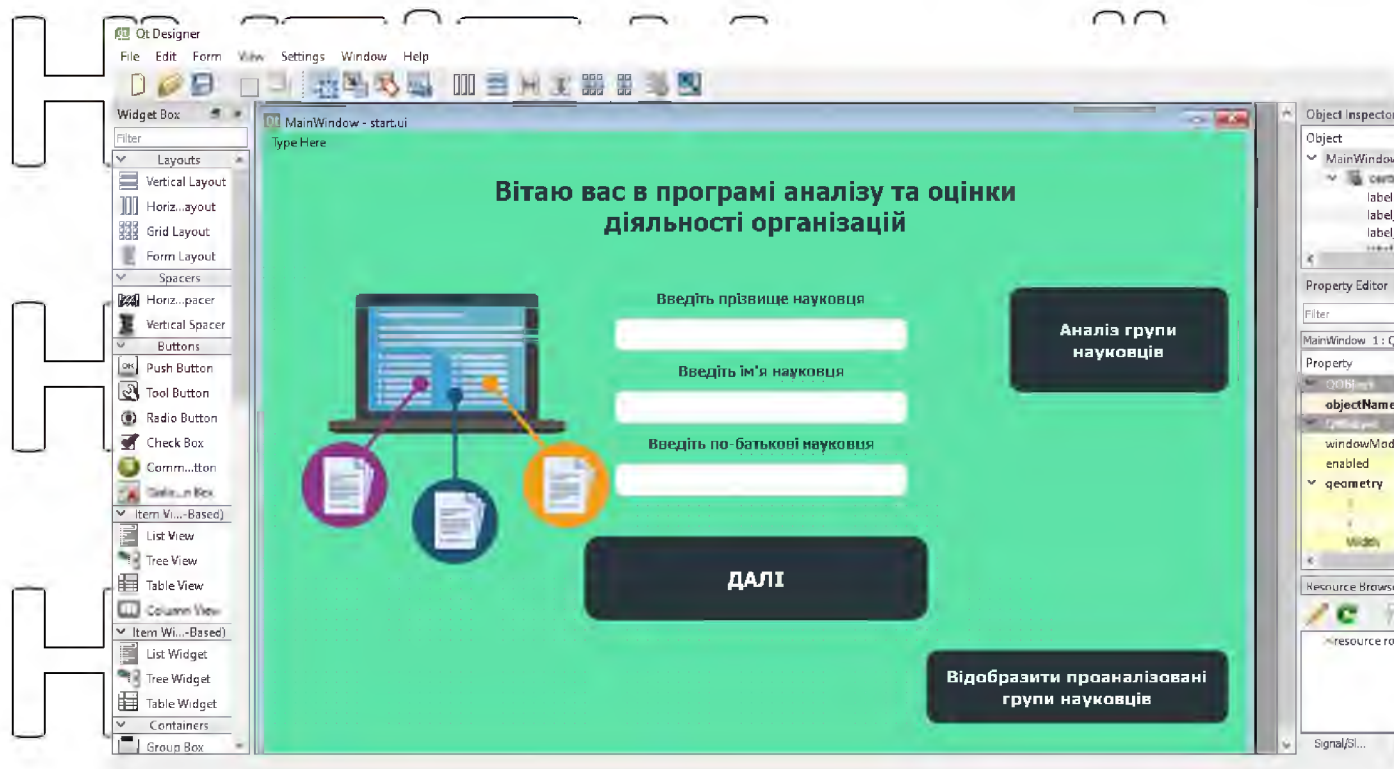


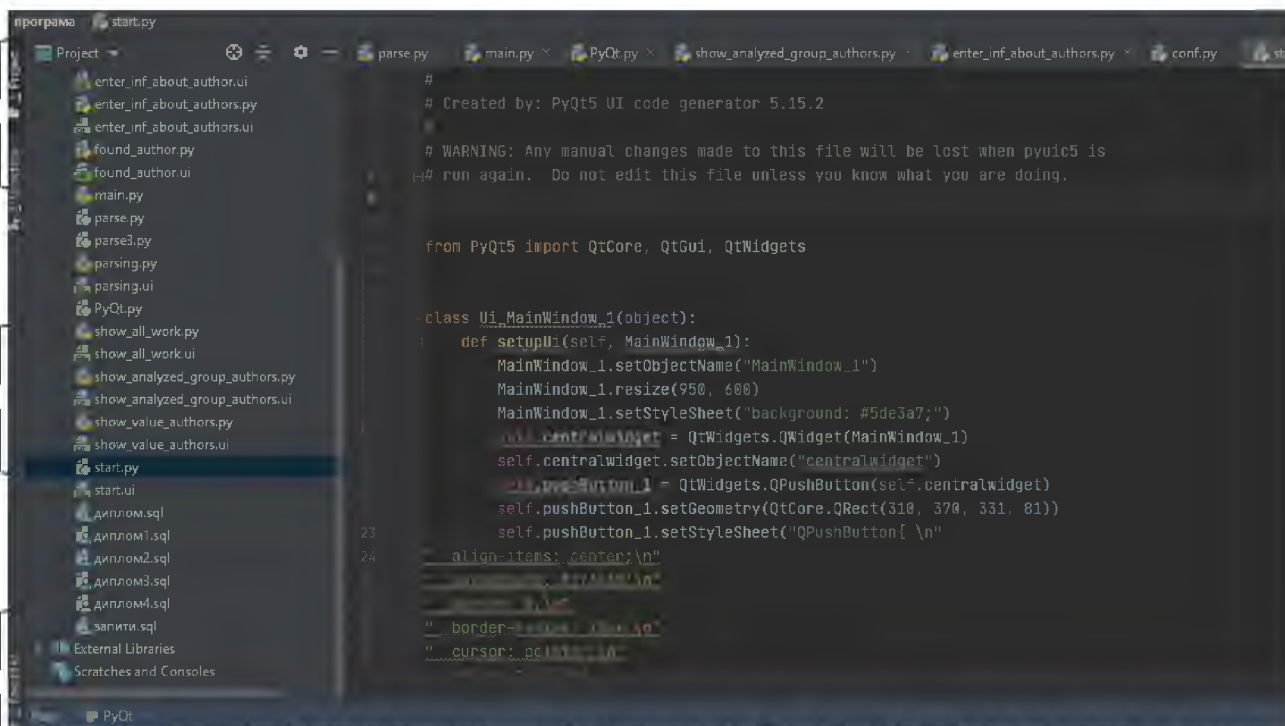
Рисунок 3.4. Робота в Qt Designer

Після створення кожного вікна потрібно форматувати кожен файл в файл, який підтримує мова програмування Python. Для цього в командній строці для кожного файлу потрібно прописати такий текст: `“pyuic4 input.ui -o output.py”`.

Після чого отримуємо файли, які можна відкрити в PyCharm (рисунок 3.5).

НУБІП України

НУБІП України



```

# Created by: PyQt5 UI code generator 5.15.2
# WARNING: Any manual changes made to this file will be lost when pyuic5 is
# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow_1(object):
    def setupUi(self, MainWindow_1):
        MainWindow_1.setObjectName("MainWindow_1")
        MainWindow_1.resize(950, 600)
        MainWindow_1.setStyleSheet("background: #5de3a7;")
        self.centralwidget = QtWidgets.QWidget(MainWindow_1)
        self.centralwidget.setObjectName("centralwidget")
        self.pushButton_1 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_1.setGeometry(QtCore.QRect(310, 370, 331, 81))
        self.pushButton_1.setStyleSheet("QPushButton { \n"
23         "    align-items: center;\n"
24         "    border-radius: 10px;\n"
         "    border: 2px solid #000000;\n"
         "    cursor: pointer;\n"

```

Рисунок 3.5. Програмний код створеного початкового вікна

Щоб все правильно працювало, до Python повинна бути підключена бібліотека PyQt (набір графічних розширень) [14]. В будь-який момент можна змінити дизайн програми вручну в коді програми або ж в програмі Qt Designer, після чого повторити переформатування в файл, підтримуваний Python.

3.5 Бібліографічне джерело даних core.ac.uk

Бібліографічне джерело CORE є найбільшим у світі агрегатором дослідницьких робіт відкритого доступу зі сховищ і журналів [5]. Це некомерційна служба, присвячена місії відкритого доступу. Дана система обслуговує глобальну мережу репозиторів і журналів, підвищуючи можливості виявлення та повторного використання вмісту відкритого доступу.

Місія CORE полягає в тому, щоб об'єднати всі дослідження відкритого доступу в усьому світі та забезпечити необмежений доступ для всіх.

Джерело даних CORE може:

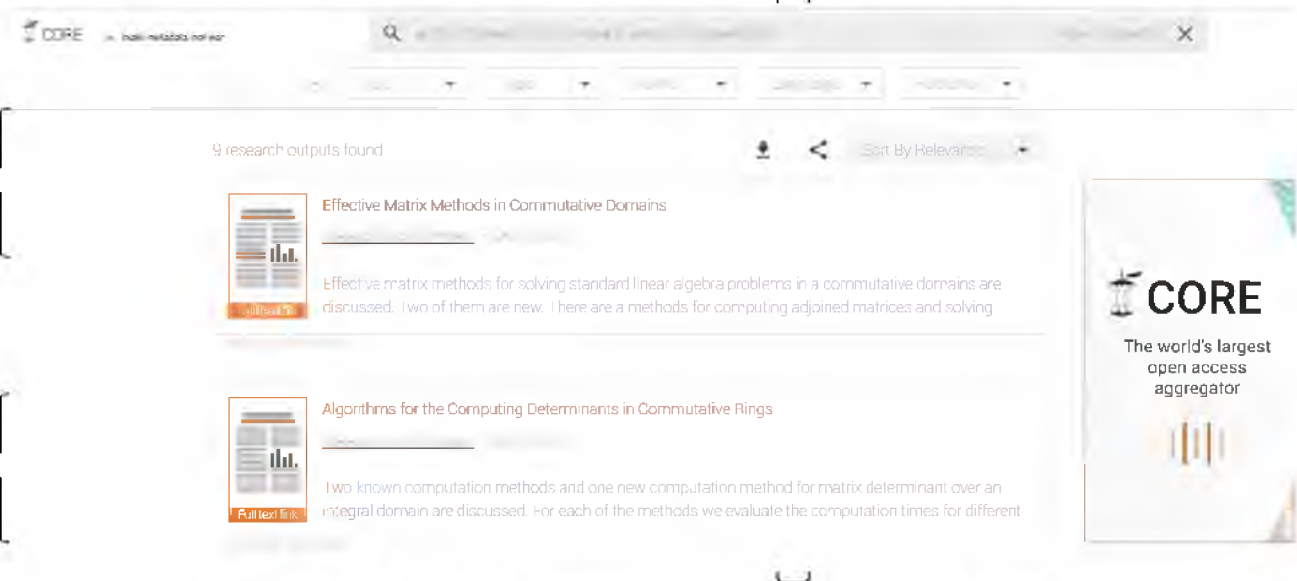
збагачувати наукові дані за допомогою найсучасніших технологій інтелектуального аналізу тексту, щоб полегшити їх виявлення;

- дозволити іншим розробляти нові інструменти та сценарії використання на платформі CORE;

- підтримувати мережу репозиторіїв відкритого доступу та журналів інноваційними технічними рішеннями.

Наразі CORE містить 246 мільйонів статей у відкритому доступі, зібраних від 11 тисяч постачальників даних у всьому світі.

Дане бібліографічне джерело надає можливість шукати роботи по ПІБ науковця і фільтрувати їх по рокам, що чудово підходить для даного проекту. Достатньо в полі створити запит і сайт надасть всю інформацію (рисунк 3.6).



Рисунк 3.6. Відображення робіт викладача по запиту

Для даного проекту, щоб проаналізувати викладачів, потрібна така інформація про їх наукові роботи: назва роботи, рік публікації, абстракт, автори.

Можна побачити, що даний сайт всю цю інформацію надає. Це дає чудову можливість зібрати всі дані не переходячи на додаткові сторінки, що зекономить час на збір інформації. Для того, щоб ці дані зібрати і зберегти в БД, потрібно налаштувати парсинг.

3.5 Парсинг даних за допомогою Selenium WebDriver

Selenium WebDriver – це бібліотека, яка здійснює управління браузером [15]. WebDriver – це компактний об’єктно-орієнтований API, керує браузером так, як це робив би користувач локально або на віддаленій машині за допомогою сервера Selenium. Дана бібліотека сама відкриває браузер і переходить по тих посиланнях, які задав програміст в програмі. Це робиться автоматизовано і дана система здатна здійснювати парсинг великої кількості даних.

Selenium підтримує автоматизацію всіх основних браузерів на ринку за допомогою WebDriver. Selenium використовує сторонні драйвери, де це можливо, але також надає власні драйвери, які підтримуються проектом, для випадків, коли це нереально.

Однією з важливих функцій Selenium є те, що процес парсингу є візуальним і можна відслідкувати його роботу. Також це вимагає мінімум часу, після збору інформації на одній сторінці система переходить до іншої. Веб-драйвер чекає, поки сторінка прогрузиться і тільки після цього збирає дані, це допомагає уникнути помилок. Якщо проблеми з інтернетом і сторінка не прогрузиться, то програма почекає відповідний час, якщо відповіді від сайту так і не отримає, то інформує користувача про помилку.

Налаштування Selenium значно відрізняється від налаштувань інших комерційних інструментів. Перш ніж почати писати код Selenium, потрібно встановити бібліотеки прив’язок мови для вибраної мови, браузера, який потрібно використовувати, і драйвер для цього браузера.

Для даного проєкту буде використовуватися браузер Google Chrome, тому відповідно для коректної роботи потрібно встановити драйвер ‘chromedriver’ підключити бібліотеку Selenium (рисунок 3.7).

```
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
```

Рисунок 3.7. Підключення бібліотеки Selenium

НУБІП України

Отже, Selenium WebDriver являється гарним варіантом для парсингу даних з бібліографічного джерела CORE.

Для ефективного збору інформації потрібно ознайомитися з міжнародним джерелом, з якого потрібно збирати інформацію (рисунок 3.8).

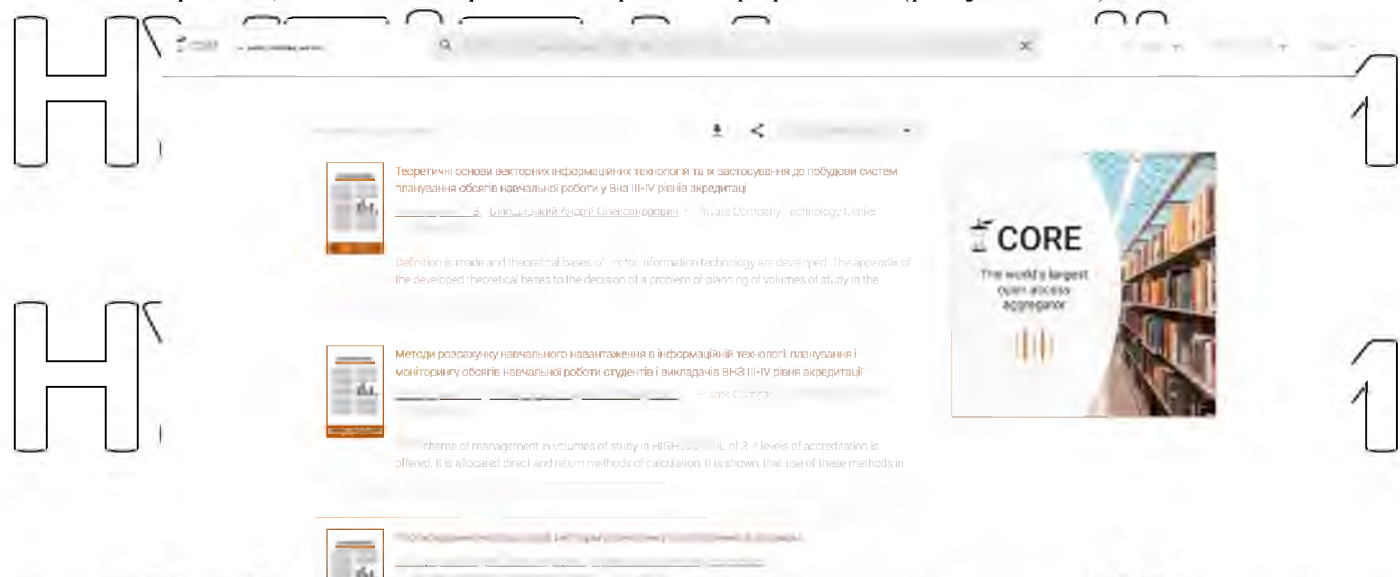


Рисунок 3.8. Ознайомлення з сайтом CORE

НУБІП України

Після аналізу даного сайту, можна побачити, що по прізвищу викладача і вказаних років можна знайти його роботи. На одній сторінці знаходяться назва

НУБІП України

роботи, автори, дата публікації і абстракт. Якраз все те, що потрібно для подальшої оцінки викладачів.

У браузері в режимі розробника необхідно ознайомитися з кодом сайту, назвою класів, щоб зрозуміти, де знайти саме ту інформацію, яка потрібна (рисунок 3.9).

НУБІП України

НУБІП України

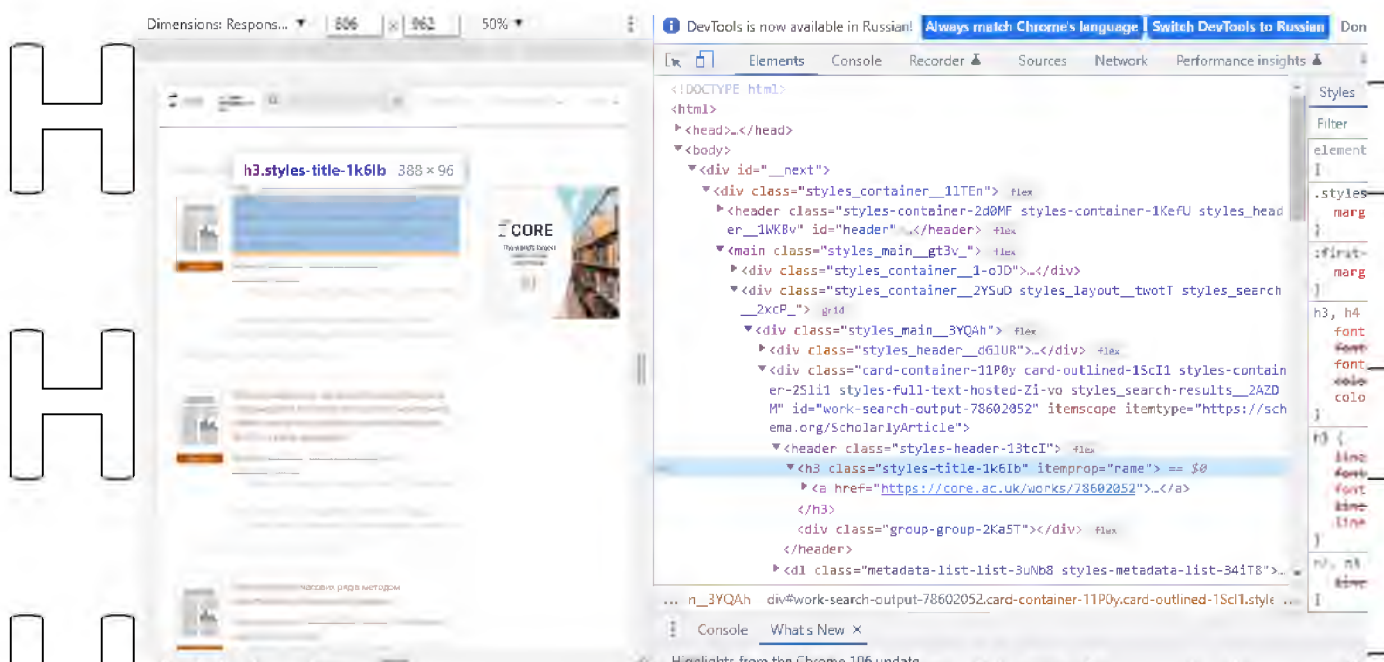


Рисунок 3.9. Ознайомлення з кодом сайту перед парсингом даних

Для того, щоб збільшити ймовірність знаходження наукових робіт викладача, пошук кожного викладача буде проводитися в дев'ятька запитів: по прізвищу і імені, а також тільки по самому прізвищу. Якщо в БД є дані викладача англійською мовою, то створюються додаткові запити і пошук займає трошки більше часу. Також кожна робота ще додатково буде перевірятися, чи точно цей викладач є автором даної роботи. Так, як автора можуть позначити по різному (Грязюк В. О., В. О Грязюк), то розроблена додаткова перевірка по ПІБ. Враховано 12 різних варіантів написання ПІБ (рисунки 3.10).

```

def parser_core(update_id, author, from_year, to_year, surname=..., name=..., middle_name=..., date_author=(...)):
    driver = webdriver.Chrome(executable_path='C:\chromedriver.exe')
    author = [
        (1) (1) (1) .format(surname, name, middle_name),
        (1) (1) (2) .format(name, middle_name, surname),
        (1) (1) (2) .format(name[0], middle_name[0], surname),
        (1) (1) (2) .format(name[0], middle_name[0], surname),
        (1) (1) (2) .format(surname, name[0], middle_name[0]),
        (1) (1) (2) .format(surname, name[0], middle_name[0]),
        (1) (1) (2) .format(name[0], middle_name[0], surname),
        (1) (1) (2) .format(surname, name[0], middle_name[0]),
        (1) (1) (2) .format(surname, name[0], middle_name[0]),
        (1) (1) (2) .format(surname, name[0], middle_name[0]),
        (1) (1) (2) .format(surname, name[0], middle_name[0]),
        (1) (1) (2) .format(name, middle_name[0], surname)]
    parser = Parser_core(driver, surname, name, middle_name, from_year, to_year, author, id_author, update, date_author)
    parser.parse_core()

```

Рисунок 3.10. Різні варіанти написання ПІБ

Це допоможе відкинути помилкові роботи, які покаже сайт. Якщо в автора на сайті не буде відображено ім'я і по-батькові, але прізвище буде таке, яке шукає система, то дану роботу не буде взято, так як існують люди з однаковими прізвищами і не факт, що ця робота саме його. Програма збирає тільки ті роботи, які є 100% цієї людини, яку аналізує.

3.6 Алгоритм аналізу та оцінки діяльності викладачів, використовуючи метрику Подібність Джаро – Вінклера

Якщо простими словами, то Подібність Джаро – Вінклера – це спосіб вимірювання схожості двох рядків. Дана метрика легка в розумінні і швидка в реалізації.

Метрики подібності рядків мають різноманітне використання, від користувацьких функцій пошуку до перевірки орфографії. У випадку з дипломним проектом, дана метрика потрібна для того, щоб порівнювати ключові слова зі словами з абстракту наукових робіт викладачів.

Подібність Джаро – Вінклера вимірює відстань між двома послідовними символами та знаходить мінімальне число, необхідне для заміни одного слова іншим. Чим більше число, тим відповідно більший збіг слів (числа лежать на проміжку від 0 до 1) [16]. Якщо результатом буде число 1, то це означає що слова повністю однакові.

Для того, щоб отримати число схожості слів, потрібно використати дві формули. Спочатку необхідно розглянути Відстань Джаро.

Відстань Джаро між двома заданими рядками s_1 і s_2 можна обчислити за формулою (1):

$$d_j = \begin{cases} 0 & \text{if } s_1 = s_2 \\ \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & \text{otherwise} \end{cases} \quad (1)$$

де $|s_i|$ – довжина рядка s_i ,
 m – число співпадаючих символів,
 t – половина числа транспозицій.

Два символи з першого (s_1) і другого (s_2) рядка співпадають лише тоді,

коли вони однакові і не далі, ніж $\frac{\max(|s_1|, |s_2|)}{2}$. Кожен символ рядка s_1

порівнюється з усіма відповідними йому символами в s_2 [17]. Кількість транспозицій визначають кількість співпадаючих символів, які діляться на 2.

Порівняємо 2 слова: «MARTHA» і «MARHTA»
 $s_1 = \text{«MARTHA»};$
 $s_2 = \text{«MARHTA»};$

$a = \text{довжина рядка}(s_1);$

$b = \text{довжина рядка}(s_2);$
 $d = \max(a, b) / 2 - 1;$
 $m = 6;$

$tr = 2;$

$t = tr / 2;$

Отже, використовуючи формулу (1) маємо такий розв'язок:
 $d_j = (6/6 + 6/6 + (6 - 1)/6) / 3;$
 $d_j = 0.944;$

Можна побачити, що відстань між двома словами дорівнює майже 1, це означає, що всі букви в цих словах майже співпадають.

В Подібності Джаро – Вінклера використовується коефіцієнт масштабування, він дає більш сприятливу оцінку рядків, які збігаються один з одним від початку до певної довжини, ця довжина називається префіксом.

Відстань Джаро – Вінклера можна розрахувати за формулою (2):

$d_w = d_j + (lp(1 - d_j)),$ (2)

де d_j – відстань Джаро для рядків, S_1 і S_2 ,
 l – довжина загального префіксу від початку рядка до максимуму 4-х символів,

p – постійний коефіцієнт масштабування, який використовується для коригування оцінки для виявлення наявності загальних префіксів. Він не може перевищувати 0,25, інакше фінальне число може стати більше 1. Стандартне значення цієї константи дорівнює 0,1.

Взявши до прикладу задачу, яка була представлена вище, можна побачити, що безперервний ланцюжок символів, які збіглися дорівнює 3. Тому $l=3$, p – це стандартне значення, яке дорівнює 0,1. Використовуючи формулу (2) маємо:

$$dw = 0.944 + (3 * 0.1 * (1 - 0.944));$$

$$dw = 0.961;$$

Отже, схожість слів «MARTHA» і «MARHTA» мають значення 0.961. Це говорить про те, що ці слова мають дуже високий показник схожості.

Існують також інші популярні міри відстані редагування, які обчислюються з використанням іншого набору допустимих операцій редагування [17].

Відстань Левенштейна
 Принцип залишається той самий: чим більша відстань, тим менше схоже слово один на одного. Але знаходження відстані зовсім інше. Тут вводиться поняття односимвольної операції (їх лише три):

- вставка - додаємо новий символ (сон > слон);
- видалення - видаляємо символ (гідрат > гідрат);
- заміна - замінюємо символ (сон > сом).

Недоліки відстані Левенштейна:

- важко знаходити мінімальну кількість односимвольних операцій;
- при перестановці слів показує великі відстані;
- відстань між короткими, але зовсім різними словами - невеликі, у той час як між довгими рядками, але схожими – великі.

Маючи дані односимвольної операції, ми можемо перетворити одне слово в інше. Відстань по Левенштейну між двома словами визначається як мінімальна кількість односимвольних операцій, необхідних для перетворення з одного слова в інше [18]. Застосовував автор дану метрику під час перевірки правопису чи пошуку дублікатів.

Відстань Дамерау – Левенштейна

Працює так само як і відстань Левенштейна, але тут додано четверту односимвольну операцію, яка називається транспозиція - заміна місцями двох символів (наприклад шишка - шишак). Це частково вирішує проблему великих відстаней під час перестановки слів, але ускладнює алгоритм знаходження мінімальної кількості операцій.

Алгоритм пошуку знаходить застосування в реалізації нечіткого пошуку, а також в біоінформатиці (порівняння ДНК), незважаючи на те, що алгоритм розроблявся для порівняння текстів, набраних людиною (Дамерау показав, що 80% людських помилок при наборі текстів становлять перестановки сусідніх символів, пропуск символу, додавання нового символу і помилка в символі, тому метрика Дамерау-Левенштейна часто використовується в редакторських програмах для перевірки правопису) [19].

Відстань Хеммінга

Маючи безліч слів, ми можемо задати правило за яким обчислюватимемо відстань між словами, аналогічно тому як ми обчислюємо в просторі точок відстань між точками, також ми можемо задати правило за яким обчислюватимемо відстань між словами. Чим більша ця відстань, тим менше схожі слова і навпаки. Найпростіший приклад - це відстань Хеммінга. Дане правило працює тільки для слів однакової довжини і обчислюється як число позицій символів, що відрізняються. Приклад, порівняємо два слова:

- ромашка;
- монашка.

У даному випадку 2 позиції відрізняються (перша та третя), означає відстань 2.

Отже, максимальна відстань між словами дорівнює довжині порівнюваних слів.

Метрика була сформульована Річардом Хеммінгом під час його роботи в Bell Labs для визначення міри відмінності між кодовими комбінаціями (двійковими векторами) у векторному просторі кодових послідовностей [20].

Можна зробити висновок, що найкращим методом розрахунку для даної роботи являється Подібність Джаро - Вінклера або просто відстань Джаро, ця метрика видає відносно найкращі результати, не складні алгоритми обчислення,

нормований результат, який можна переводити у відсотки, а також в подібності

Джаро - Вінклера використовується коефіцієнт масштабування, що дає більш сприятливу оцінку рядків.

Для даного програмного продукту було вирішено використовувати саме цю метрику подібності слів і за допомогою неї провести аналіз і оцінку

діяльності викладачів. Порівнювати потрібно кожне слово абстракту наукової роботи з кожним ключовим словом, отримати список значень. З цього списку обрати 10 максимальних значень і визначити середнє арифметичне цих чисел. Це і буде число схожості наукової роботи до ключових слів діяльності викладача.

Після чого провести такий аналіз з усіма роботами викладача, отримати список чисел, знайти середнє значення цих чисел. Отримане значення помножити на 100 і отримати відсоток схожості наукових робіт автора до його ключових слів.

У даному додатку розроблене окреме вікно для вводу ключових слів (рисунок 3.11).

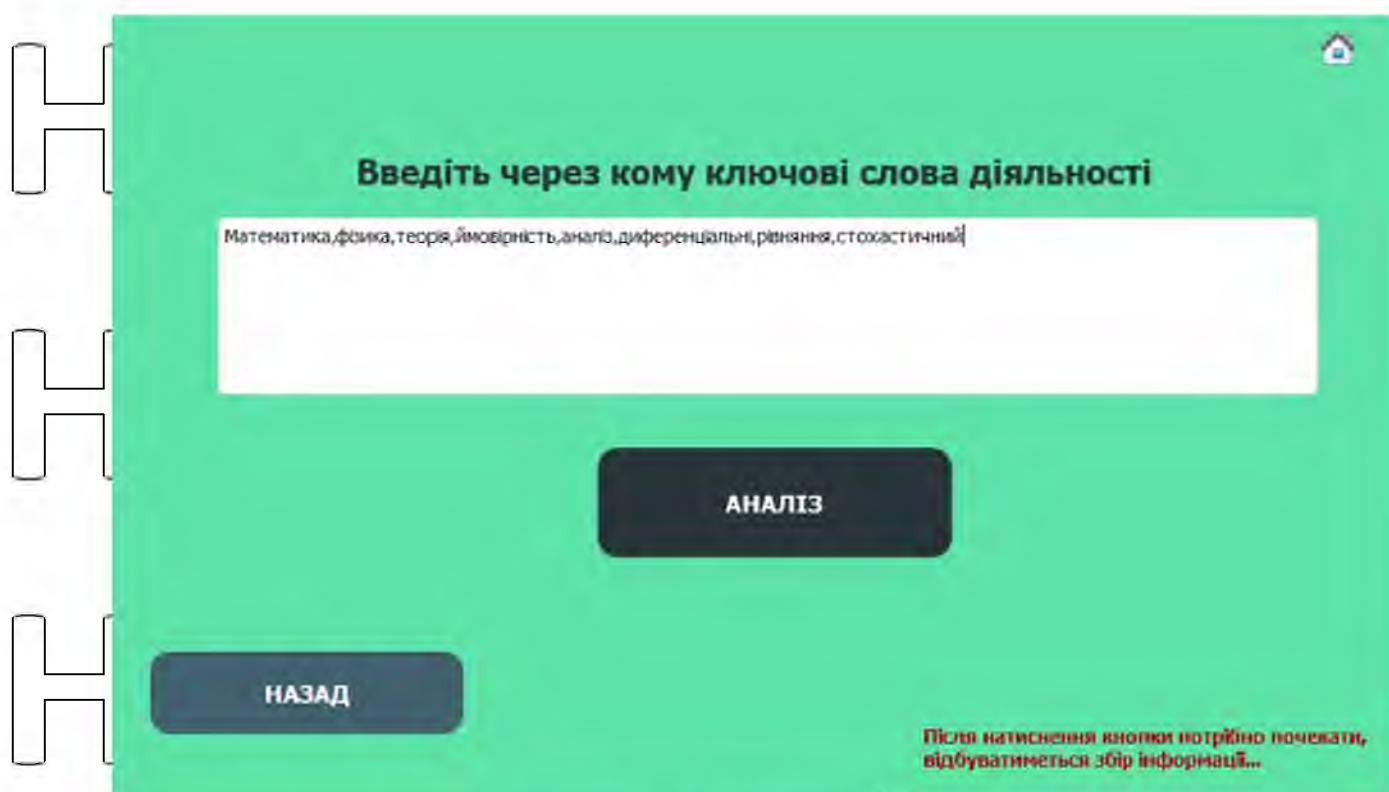


Рисунок 3.11. Ввід ключових слів

Після чого відбувається процес порівняння ключових слів зі словами із абстракту наукових робіт викладачів.

Для прикладу візьмемо декілька ключових слів наукової роботи і порівняємо з ключовими словами діяльності викладача.

Ключові слова роботи: енергетичний потенціал, природно-ресурсний комплекс.

Ключові слова діяльності: енергетика, природа

З кожною парою слів знайдемо схожість Джаро – Вінклера:

1. Енергетика – Енергетичний

$$\begin{matrix} a = 10 \\ b = 12 \end{matrix}$$

$$d = 5$$

$$m = 8$$

$$tr = 0$$

$$t = 0$$

$$dj = (8/10 + 8/12 + 8/8) / 3 = 0.82$$

$$dw = 0.82 + (4 * 0.1 * (1 - 0.82)) = 0.892$$

2. Енергетика – Потенціал

$$dj = (4/10 + 4/9 + 3/4) / 3 = 0.53$$

$$dw = 0.53 + (0 * 0.1 * (1 - 0.66)) = 0.53$$

3. Енергетика – Природно-ресурсний

$$dj = (4/10 + 4/18 + 3/4) / 3 = 0.46$$

$$dw = 0.46 + (0 * 0.1 * (1 - 0.46)) = 0.46$$

4. Енергетика – Комплекс

$$dj = (2/10 + 2/8 + 2/2) / 3 = 0.48$$

$$dw = 0.48 + (0 * 0.1 * (1 - 0.48)) = 0.48$$

5. Природа – Енергетичний

$$dj = (2/7 + 2/12 + 2/2) / 3 = 0.48$$

$$dw = 0.48 + (4 * 0.1 * (1 - 0.48)) = 0.48$$

6. Природа – Потенціал

$$dj = (3/7 + 3/9 + 3/3) / 3 = 0.59$$

$$dw = 0.59 + (1 * 0.1 * (1 - 0.59)) = 0.631$$

7. Природа – Природно-ресурсний

$$dj = (6/7 + 6/18 + 6/6) / 3 = 0.73$$

$$dw = 0.73 + (4 * 0.1 * (1 - 0.73)) = 0.838$$

8. Природа – Комплекс

$$dj = (2/7 + 2/8 + 2/2) / 3 = 0.51$$

$$dw = 0.51 + (0 * 0.1 * (1 - 0.51)) = 0.51$$

Для реалізації аналізу робіт використовується абстракт, який має достатньо слів, тому кожне ключове слова порівнюється з кожним словом абстракту, потім береться 10 максимальних значень і знаходиться середнє значення. І так для кожного ключового слова. У прикладі, наведеному вище, маємо тільки 4 слова з абстракту, тому беремо тільки по одному максимальному

значенню кожного ключового слова і знаходимо середнє значення. Це і буде відсоток схожості роботи до наукової діяльності викладача.

Отже, найбільш схожі слова до ключових «енергетичний» і «природно-ресурсний» з показниками 0.892 та 0.838 відповідно.

$$(0.892 + 0.838)/2 * 100 = 86.5\%$$

Підрахунки показали, що викладач має високий відсоток схожості однієї роботи до його наукової діяльності, а саме 86.5%.

Для кожної роботи викладача проробляються ті самі дії, отримуємо відсоток схожості одного автора для кожної його роботи, потім розраховується

середнє значення для всіх цих робіт і в результаті отримуємо число, яке показує наскільки роботи схожі по тематиці з науковим напрямком викладача.

У результаті отримуємо дані, де відображається по черзі кожен викладач, кількість робіт за обраний проміжок часу і відсоток збігу робіт до напрямку діяльності кожного викладача, а також середній відсоток групи (рисунок 3.12).

Оцінка діяльності

| | Прізвище | Ім'я | По-батькові | Surname | Name | Middlename | К-сть робіт(2005 - 2021 роки) | % |
|---|------------|----------|---------------|------------|-------|---------------|-------------------------------|-------|
| 1 | Тимошенко | Олена | Анатолівна | Tymoshenko | Olena | Anatoliivna | 9 | 87.81 |
| 2 | Голіченко | Ірина | Ігорівна | None | None | None | 2 | 83.9 |
| 3 | Москвичова | Катерина | Костянтинівна | None | None | None | 7 | 72.0 |
| 4 | Блажівська | Ірина | Петрівна | None | None | None | 4 | 66.55 |
| 5 | Орловський | Ігор | Володимирович | Orlovsky | Igor | Vladimirovich | 15 | 50.29 |
| 6 | Репета | Леся | Анатолівна | None | None | None | 2 | 41.05 |

Співпадіння робіт до напрямку діяльності науковців дорівнює 66.93%

НАЗАД

Рисунок 3.12. Оцінка діяльності викладачів по ключовим словам

Програма, використовуючи метрику Подібність Джаро-Вінклера, проаналізувала викладачів по ключовим словам і абстракту робіт цих авторів, якщо в деяких роботах відсутній абстракт, то система робила аналіз по назвам наукових робіт.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У ході виконання даного дослідження було створено систему у вигляді програмного додатку, який має функціональний інтерфейс і виконує усі функції, які були поставлені в завданні даного проекту.

4.1 Системні вимоги та встановлення програмного продукту

Для того, щоб програма працювала на персональному комп'ютері чи ноутбуці без збоїв, потрібні, як мінімум, такі характеристики:

- процесор Intel Core i3, 3.00 GHz;
- пам'ять на жорсткому диску: 10 ГБ;
- об'єм оперативної пам'яті: 4 ГБ;
- ОС Windows 7/8/10/11.

Ще однією важливою характеристикою являється безперебійне підключення до інтернету.

Якщо ПК відповідає мінімальним характеристикам, то можна перейти до налаштування програми. Спочатку потрібно скачати і встановити Python 3.8, СУБД MySQL і MySQL Workbench. На комп'ютері повинен бути браузер Google Chrome. Для роботи з Python краще всього встановити PyCharm Community Edition (для успішного запуску програми).

Після встановлення Python потрібно за допомогою PIP підключити такі бібліотеки: `mysql-connector-python`, `selenium`, `PyQt5`, `requests`.

Після успішного встановлення бібліотек, потрібно відкрити програму MySQL Workbench, при запуску створити свій сервер, придумати логін, пароль і запустити файл "База даних.sql". Натиснути відповідну кнопку (рисунком 4.1), після чого створиться база даних, в яку надалі буде зберігатися вся інформація.

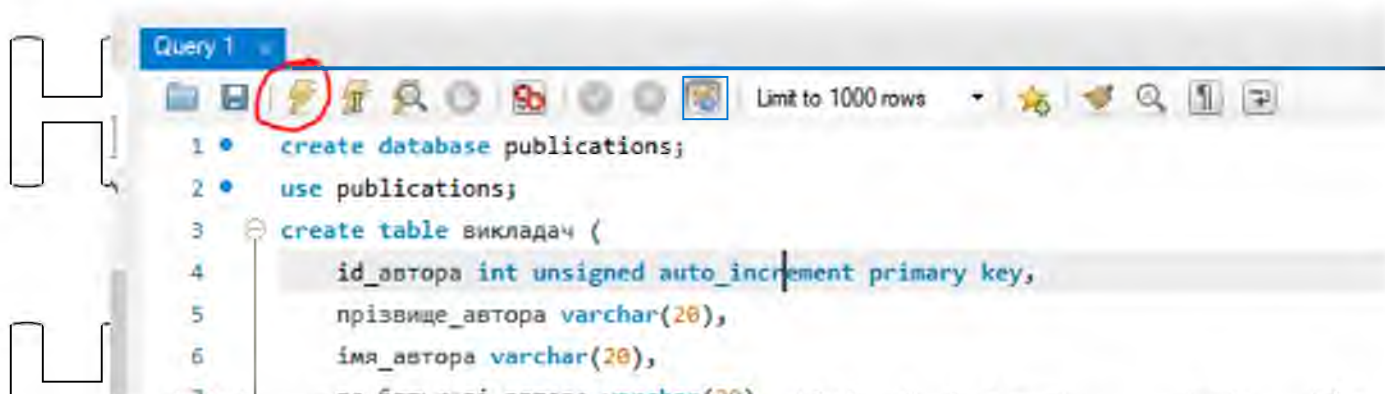


Рисунок 4.1. Створення локальної бази даних

Після створення бази даних потрібно відкрити файл “PyQt.py”. На початку коду, де інформація про підключення до БД, потрібно ввести свій логін і пароль до серверу (рисунок 4.2), який був створений вище, а також ще в одному місці ввести тільки пароль (рисунок 4.3).



Рисунок 4.2. Ввід логіну та паролю до власного сервера

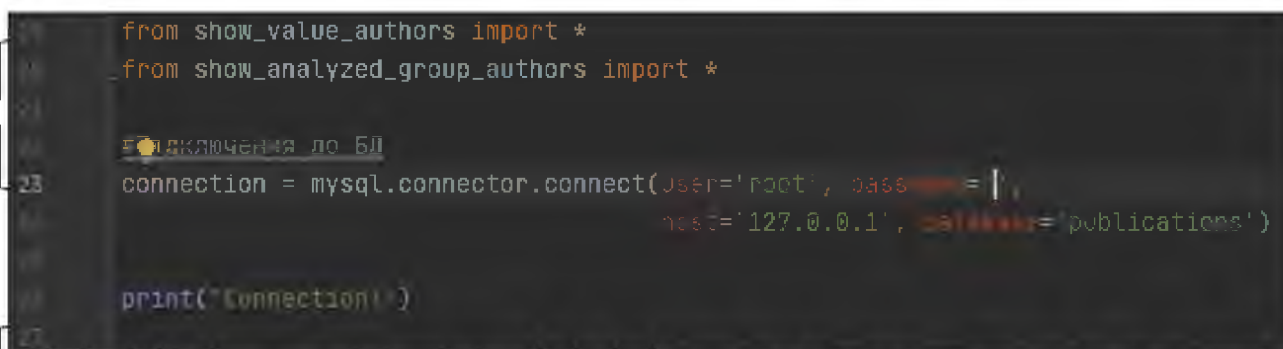


Рисунок 4.3. Ввід паролю до власного сервера

На цьому налаштування програми завершене, натискаємо праву кнопку миші і обираємо “Run” (рисунок 4.4).

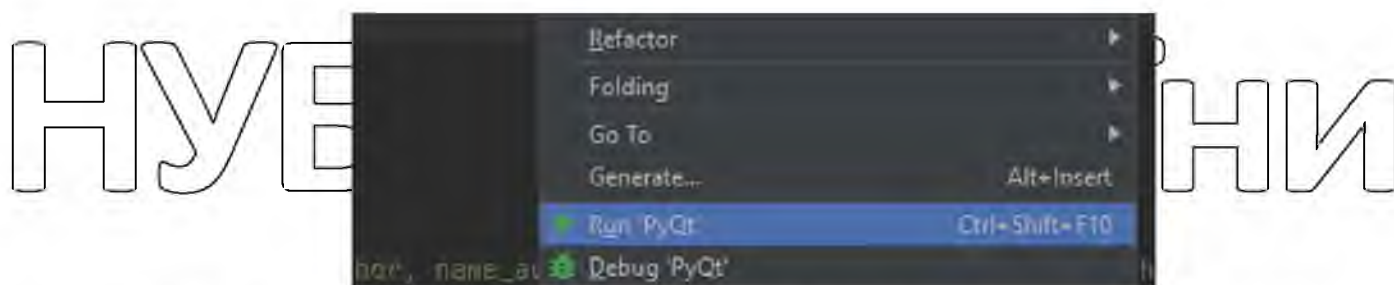


Рисунок 4.4. Запуск програми

Після чого програма повинна відкрити початкове вікно, цим самим показати про готовність роботи.

4.2 Робота з програмним продуктом

Як тільки користувач запустив програму, відкривається початкове вікно (рисунок 4.5).



Рисунок 4.5. Початкове вікно програми

На даному етапі користувач має декілька варіантів продовження роботи. Якщо потрібно проаналізувати одного викладача або подивитися чи є цей

викладача у базі даних, то потрібно ввести прізвище, ім'я і по-батькові викладача у відповідні поля (рисунок 4.6).

Введіть прізвище науковця

Морозов

Введіть ім'я науковця

Віктор

Введіть по-батькові науковця

Володимирович

Рисунок 4.6. Ввід даних для пошуку одного викладача

Якщо потрібно проаналізувати зразу декількох викладачів, які входять до однієї кафедри чи спеціальності, то потрібно натиснути відповідну кнопку (рисунок 4.7).

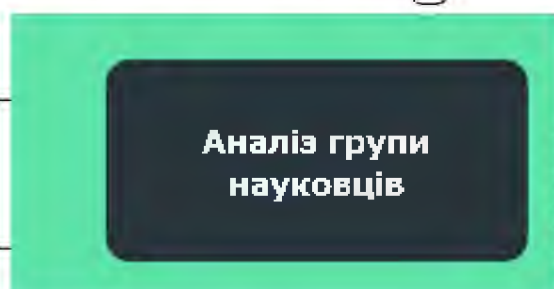


Рисунок 4.7. Кнопка “Аналіз групи науковців”

Для відображення уже проаналізованих груп є спеціальна кнопка, яка знаходиться знизу справа. Натиснувши її, можна ознайомитися з цими групами (рисунок 4.8). Висвітиться табличка, у якій відображаються групи. А саме назва групи, викладачі, які входять до окремих груп, ключові слова діяльності цих груп, роки, за які бралися наукові роботи, кількість цих робіт і відсоток збігу наукових робіт до напрямку діяльності окремої групи.

| Ідентифікатор групи | Ім'я групи(кафедри) | Науковці(викладачі) | Ключові слова | Роки | Кількість робіт | Відсоток збігу |
|---------------------|------------------------------------|---|--|-------------|-----------------|----------------|
| 1 | Філологія (2015-2020) | Сліпушко Оксана Миколаївна; Бернадська Ніна Іванівна; Романенко Олена Віталіївна | Філологія, наука, українська, література, художні, твори, письменник, історія, мова | 2015 - 2020 | 9 | 73.93 |
| 2 | Інформаційні системи та технології | Білощицький Андрій Олександрович; Палій Сергій Володимирович; Бронін Сергій Вадимович | Інформаційні технології, системи, проектування, ІТ, математична модель, архітектура, програмне забезпечення, програма, програмування, системний аналіз | 2009 - 2018 | 20 | 70.02 |
| 3 | ІТ | Морозов Віктор Володимирович; Лукова-Чуйко Наталія Вікторівна, Lukova-Chuiko Natalia Viktorivna | ІТ, проекти, технології, штучний інтелект, програмне забезпечення, бази даних, управління, моделі, програмування, системи | 2018 - 2021 | 5 | 89.64 |

НАЗАД

Рисунок 4.8. Проаналізовані групи викладачів

На всіх сторінках, крім початкової, в правому верхньому кутку присутня кнопка “Home” (рисунок 4.9), натиснувши на неї, користувач перейде на головне вікно.



Рисунок 4.9 – Кнопка “Home”

Якщо повернутися до аналізу одного викладача або перевірки його наявності в БД, то на початковій сторінці (рисунок 4.5), ввівши дані викладача, натискаємо “Далі” (рисунок 4.10).

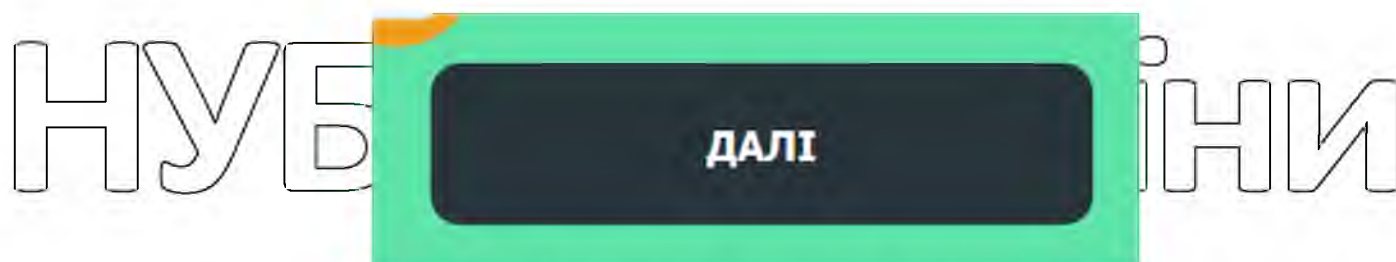


Рисунок 4.10. Перехід до пошуку викладача в БД

Після натиснення даної кнопки відбувається пошук інформації в БД по ПІБ викладача, якщо така інформація знаходиться, то відображається вікно, яке зображене на рисунку 4.11.



Рисунок 4.11. Дані викладача, які присутні в БД

У даному вікні можна побачити дві таблиці. У одній із них розміщена інформація про викладача у вигляді ПІБ українською і англійською мовами. А в іншій – кількість робіт за певні роки.

Далі користувач може виконати декілька дій:

- переглянути всі роботи даного викладача,

• оновити дані (додати ПІБ англійською або ж зібрати наукові роки за інші роки);

• перейти до оцінки діяльності цього викладача або створити групу і додати ще інших викладачів;

- знайти іншого викладача.

Якщо даного викладача б не було в базі даних, то відкрилося б вікно, на якому програма попросила б додаткові дані: ПІБ англійською мовою та роки, за якими буде проводитися пошук наукових робіт (рисунок 4.12). ПІБ, які користувач ввів на початковому вікні автоматично перенесуться відповідно в поля, де потрібно ввести українською мовою.

Для більш точного відображення даних введіть ПІБ науковця на українській та англійській мові

| | Прізвище | Ім'я | По-батькові |
|------|----------|--------|---------------|
| УКР | Морозов | Віктор | Володимирович |
| АНГЛ | | | |

Зібрати роботи науковця з по рік

Зібрати дані

НАЗАД

Після натиснення кнопки потрібно почекати, відбуватиметься збір інформації...

Рисунок 4.12. Ввід додаткової інформації для пошуку

ПІБ викладача англійською мовою потрібне для того, щоб розширити пошук робіт, так як деякі наукові роботи опубліковані під англійським прізвищем. Це поле не обов'язкове, без нього парсинг даних відбудеться, але

якщо залишити поля, де вводиться ПІБ українською мовою, то програма повідомить користувача про відповідну помилку (рисунок 4.13).

The screenshot shows a search form with three columns: "Прізвище" (Surname), "Ім'я" (Name), and "По-батькові" (Patronymic). Below these columns are two rows of input fields labeled "УКР" (Ukrainian) and "АНГЛ" (English). At the bottom left, there are dropdown menus for "Зібрати роботи науковця з" (Collect works of the scientist from) and "по" (from) with the value "2000", followed by "рік" (year). A red error message "Поля пусті. Ви не ввели дані" (Fields are empty. You did not enter data) is displayed on the right. A large black button labeled "Зібрати дані" (Collect data) is at the bottom center.

Рисунок 4.13. Помилка про пусті поля

Ввівши дані, користувач натискає кнопку "Зібрати дані". Після чого відбувається парсинг даних даного викладача, збираються наукові роботи з міжнародного джерела CORE методом Selenium WebDriver. Система автоматично відкриває браузер і починає переходити по сторінках сайту, для того, щоб зібрати необхідну інформації про викладачів (рисунок 4.14).

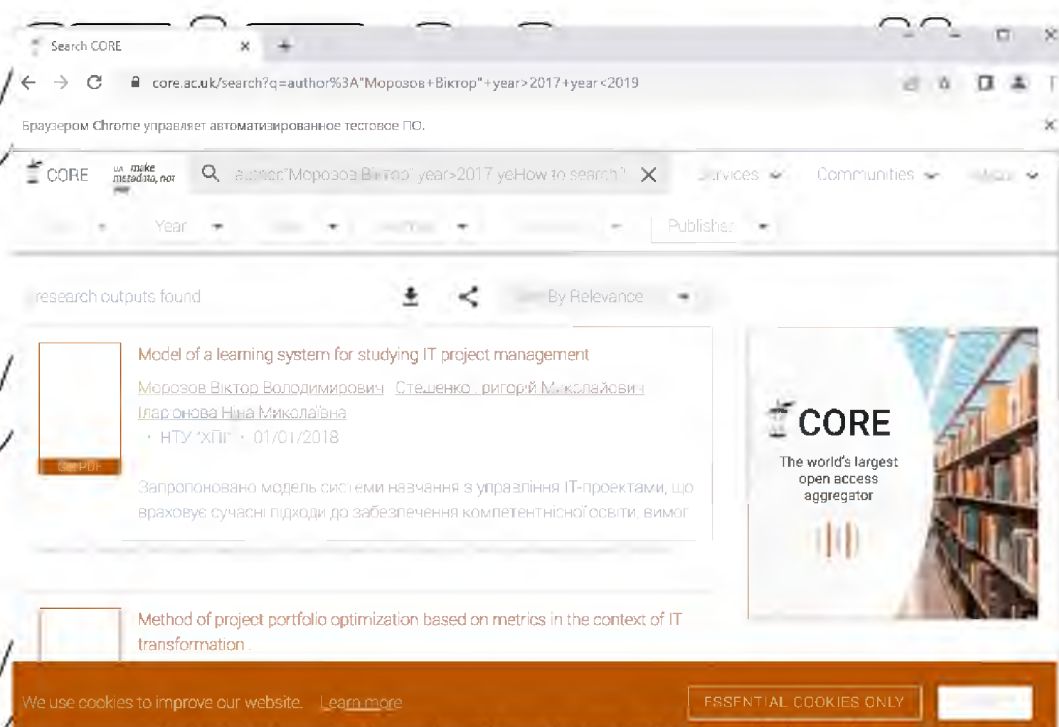


Рисунок 4.14. Збір інформації з міжнародного джерела CORE

Пошук наукових робіт відбувається по прізвищу і імені. А також створюються окремі запити, які шукають роботу по прізвищу і першій букві імені викладача. Це збільшує вірогідність не загубити наукову роботу викладача, тому що в сайту є свої алгоритми пошуку інформації по прізвищу та імені, вони можуть бути не ідеальними і загубити якусь роботу. У полі, де відображені автори/роботи перевіряється чи точно даний сайт відобразив роботу даного викладача (рисунк 4.15).

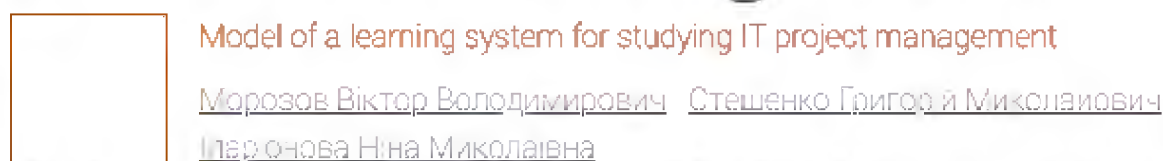


Рисунок 4.15. Автори наукової роботи

Для даного процесу потрібне стабільне підключення до інтернету, інакше є вірогідність знайти не всі роботи, які є в базі бібліографічного джерела. На рисунку 4.16 відображена помилка, яка може виникнути при проблемах з інтернетом. Дана помилка не впливає на роботу програми, вона буде далі пробувати зібрати ті дані, які необхідні. Але в результаті число кількості робіт може виявитися меншим, ніж є насправді.

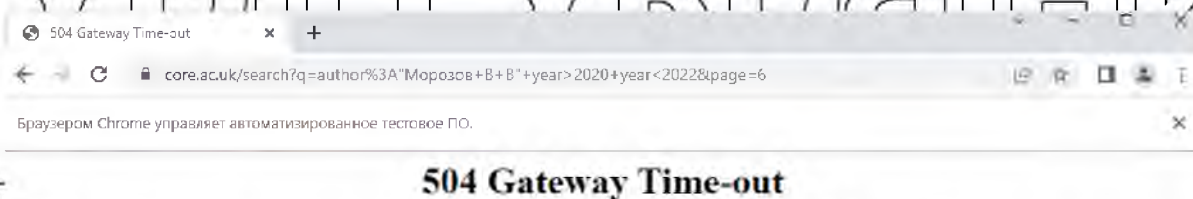


Рисунок 4.16. Можлива помилка при зборі даних

При зборі даних програма інформує користувача про те, що потрібно почекати (рисунк 4.17).

Після натиснення кнопки потрібно почекати, відбудеться збір інформації...

Рисунок 4.17. Інформаційний текст про збір інформації

Після успішного збору інформації браузер закривається і з'являється вікно, на якому відображаються дані про викладача і кількість його робіт по рокам (рисунок 4.14).

Далі користувач може переглянути всі роботи даного викладача, натиснувши кнопку “Показати всі роботи”. Програма відобразила вікно, де можна побачити всі роботи даного викладача, а саме назву роботи, автори наукових робіт, абстракт (для того, щоб далі аналізувати слова з ключовими словами для оцінки діяльності) і рік публікації (рисунок 4.18). Дані в таблиці можна сортувати по різним критеріям.

Всі роботи

| Прізвище | Ім'я | По-батькові | Surname | Name | Middlename | |
|----------|---------|-------------|---------------|------|------------|------|
| 1 | Морозов | Віктор | Володимирович | None | None | None |

НАЗАД

| № | Назва роботи | Автори | Абстракт | Рік |
|---|---|---|---|------|
| 1 | Model of a learning system for studying IT project management | Морозов Віктор Володимирович Стешенко Григорій Миколайович Ляріонова Ніна Миколаївна | Запропоновано модель системи навчання з управління IT-проектами, що враховує сучасні підходи до забезпечення компетентної освіти, виног міжнародних стандартів з управління проектами та нормативної бази ... | 2018 |
| 2 | Method of project portfolio optimization based on metrics in the context of IT transformation . | Артикульна Надія Францізна Морозов Віктор Володимирович Селіванов Володимир Вікторович | У статті досліджуються причини необхідності застосування сучасних підходів до впровадження нових продуктів та послуг для клієнтів банківських установ, розглядаються підходи до оптимізації та перформативання портфелів проєктів шляхом застосування метрик та ключових показників ефективності впровадження нових продуктів та послуг. На ... | 2018 |
| 3 | Applying the antisipative project management for development of distributed information systems | Кальніченко Олена Володимирівна Морозов Віктор Володимирович Чрутьба Андрій Сергійович | Проаналізовані сучасні тенденції розвитку інформаційних технологій та їх вплив на трансформацію бізнесу, що в свою чергу призводить до створення складних та потужних IT-продуктів з не стандартними характеристиками. Традиційні методології у повній мірі не дозволяють ефективно проводити управління такими проектами, що ... | 2018 |
| 4 | The study of processes interaction models in IT projects based on proactive approach | Кальніченко Олена Володимирівна Морозов Віктор Володимирович | Авторами статті окреслені основні проблемні аспекти, характерні для реалізації IT-проектів, які визначаються сучасними тенденціями середовища, в якому виконуються проєкти. Зазначена актуальність пошуку дієвих механізмів реагування на неконформовані позитивні/негативні впливи з боку зовнішнього ... | 2019 |

Провести аналіз та оцінку

Пошук іншого викладача

Оновити дані

Рисунок 4.18. Роботи певного викладача

У даному вікні є декілька кнопок. Якщо натиснути “Оновити дані”, то програма поверне користувача до вікна, що відображене на рисунку 4.12, після чого можна оновити дані викладача, додавши ПІБ англійською мовою, а також обрати роки, які потрібно додати для пошуку наукових робіт.

“Пошук іншого викладача” перекине користувача на початкову сторінку.

Дані кнопки дублюються з попереднього вікна для більш зручного користування програмою.

Для того, щоб перейти до вікна, яке дозволить провести аналіз та оцінку діяльності викладачів або групи викладачів, можна використати декілька

варіантів:

- натиснути кнопку “Провести оцінку діяльності та порівняти з іншими науковцями” (у вікні, де розміщені дані науковця і кількість робіт по рокам (рисунок 4.19);
- натиснути відповідну кнопку у вікні, де відображаються всі роботи викладачів (рисунок 4.20);
- натиснути “Аналіз групи науковців” на початковому вікні даної програми (рисунок 4.21).

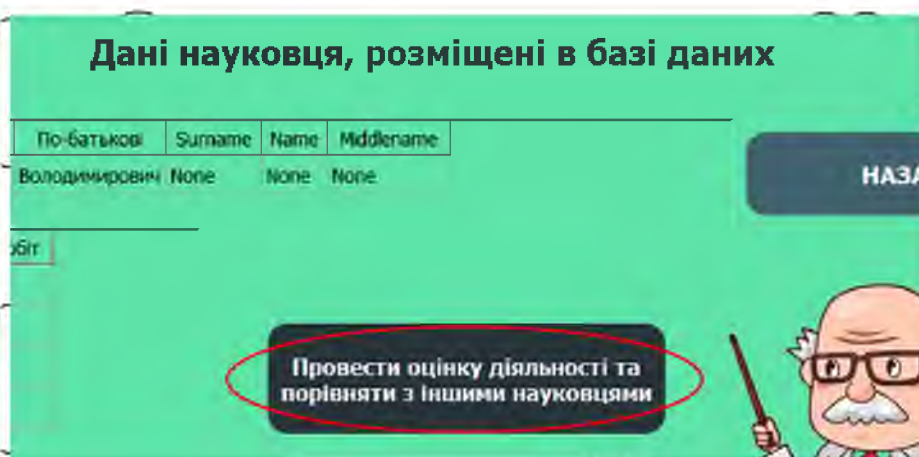


Рисунок 4.19. Оцінка діяльності викладачів у вікні “Дані науковця, розміщені в базі даних”



Рисунок 4.20. Оцінка діяльності викладачів у вікні “Всі роботи”

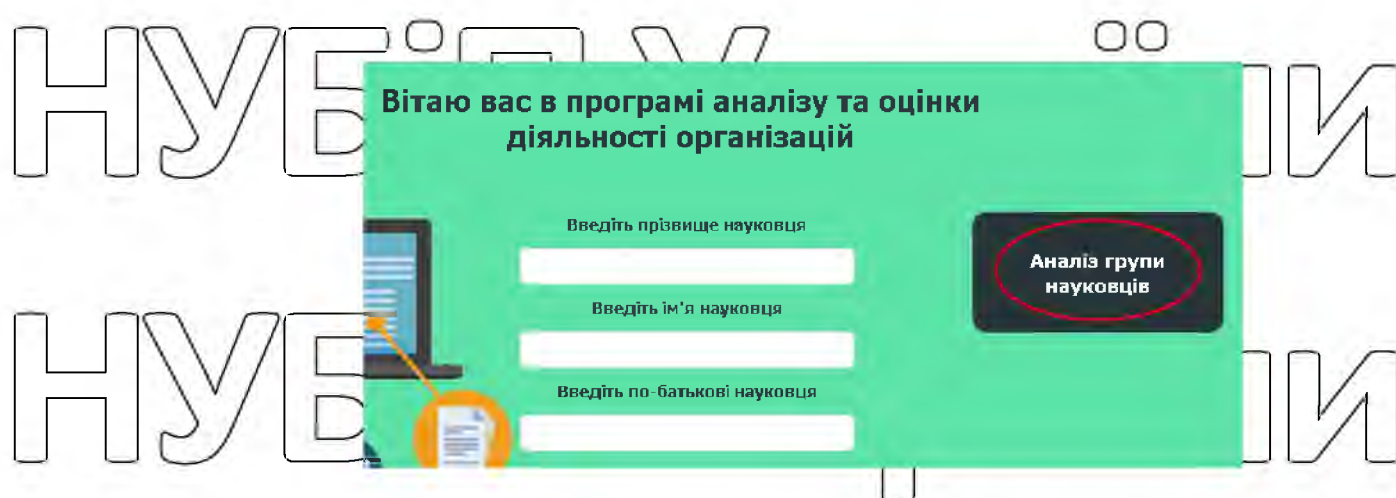


Рисунок 4.21. Оцінка діяльності викладачів на початковому вікні

Після натискання однієї з вищеперелічених кнопок, програма відкриє відповідне вікно, де уже автоматично підтягнеться ПІБ викладача, дані якого збиралися (рисунок 4.22). В центрі вікна знаходиться велике текстове поле, де потрібно ввести ПІБ одного викладача або зразу декількох викладачів, залежить від того, які цілі в користувача. Для того, щоб інформація про викладачів була коректно записана і збережена в БД, в програмі наведений приклад заповнення тексту (рисунок 4.23).

НУБІП України

Введіть через крапку з комою повністю ПІБ науковців, по можливості розділивши комою українську і англійську мову

Приклад:

Один автор - Петренко Петро Петрович, Petrenko Petro Petrovich

Деякі автори - Петренко Петро Петрович, Petrenko Petro Petrovich; Іванов Іван Іванович, Ivanov Ivan Ivanovich

Морозов Вктор Володимирович

Назва групи (кафедри):

Проаналізувати роботи з 2000 по 2000 рік

НАЗАД Аналіз по ключовим словам Аналіз по роботам

Після натиснення кнопки потрібно почекати, відбуватиметься збір інформації...

Рисунок 4.22. Аналіз та оцінка викладача або групи викладачів

Один автор - Петренко Петро Петрович, Petrenko Petro Petrovich
 Деякі автори - Петренко Петро Петрович, Petrenko Petro Petrovich; Іванов Іван Іванович, Ivanov Ivan Ivanovich

Рисунок 4.23. Вимоги до введення даних

Після вводу інформації про викладачів, користувач вводить назву групи (кафедри) і обирає роки, по яких програма буде збирати дані (рисунок 4.23).

І далі у користувача є вибір, провести аналіз по ключовим словам або по роботам.

Якщо вибрати кнопку “Аналіз по роботам”, починається перевірка інформації в базі даних, якщо цих даних недостатньо, то починається додатковий збір даних і збирається інформація, якої бракує. Під час збору інформації вікно недоступне для користувача. Система інформує користувача про те, що потрібно зачекати, оскільки інформація збирається.

НУБІП України

Введіть через крапку з комою повністю ПІБ науковців, по можливості розділивши комою українську і англійську мову

Приклади:
 Один автор - Петренко Петро Петрович, Petrenko Petro Petrovich
 Декілька авторів - Петренко Петро Петрович, Petrenko Petro Petrovich; Іванов Іван Іванович, Ivanov Ivan Ivanovich

Білощицький Андрій Олександрович; Палій Сергій Володимирович; Бронін Сергій Вадимович

Назва групи (кафедри): Інформаційні системи та технології

Проаналізувати роботи з 2009 по 2018 рік

НАЗАД Аналіз по ключовим словам Аналіз по роботам

Після натиснення кнопки потрібно почекати, відбуватиметься збір інформації...

Рисунок 4.23. Ввід групи викладачів

Після успішного збору інформації відкривається вікно, у якому відображається таблиця. У ній відображається інформація про викладачів та кількість їх робіт за обраний користувачем період часу (рисунок 4.24). Дані в таблиці можна сортувати за різними критеріями.

Аналіз науковців по кількості робіт за певний проміжок часу

| | Провище | Ім'я | По-батькові | Surname | Name | Middlename | 2009 - 2018 |
|---|-------------|--------|---------------|---------|------|------------|-------------|
| 1 | Білощицький | Андрій | Олександрович | None | None | None | 14 |
| 2 | Палій | Сергій | Володимирович | None | None | None | 4 |
| 3 | Бронін | Сергій | Вадимович | None | None | None | 2 |

НАЗАД Обрати іншу групу осіб

Рисунок 4.24. Результат оцінки викладачів по кількості робіт

Дана інформація є корисною, тому що за допомогою неї можна проаналізувати чи має викладач достатню кількість робіт, чи можливо взагалі немає. Це допоможе направити окремі організації на збільшення наукових праць.

Якщо повернутися до попереднього вікна і обрати кнопку “Аналіз по ключовим словам”, то почнеться той самий процес перевірки наявності викладачів і їх робіт в базі даних, якщо чогось буде не вистачати, то включиться парсинг даних.

Після успішної перевірки даних в БД, відкривається вікно, де потрібно ввести ключові слова напряму діяльності викладачів (рисунок 4.25).

Рисунок 4.25. Ввід ключових слів

Рекомендована кількість введених слів — не менше 7. Якщо наукові роботи викладачів збиралися і англійською мовою, то рекомендовано ще дописати ключові слова англійською мовою.

Після введення слів і натиснення кнопки “Аналіз” програма відображає таблицю, на якій можна побачити дані про викладачів, кількість робіт кожного

викладача за один проміжок часу і відсоток схожості ключових слів до наукових робіт викладачів (рисунк 4.26). Знизу відображений відеоток співпадіння робіт всієї групи викладачів.

Оцінка діяльності

| | Прізвище | Ім'я | По-батькові | Surname | Name | Middlename | К-сть робіт(2009 - 2018 роки) | % |
|---|-------------|--------|---------------|---------|------|------------|-------------------------------|-------|
| 1 | Білощицький | Андрій | Олександрович | None | None | None | 14 | 69.57 |
| 2 | Палій | Сергій | Володимирович | None | None | None | 4 | 71.75 |
| 3 | Бронін | Сергій | Вадимович | None | None | None | 2 | 68.75 |

Співпадіння робіт до напрямку діяльності науковців дорівнює 70.02%

НАЗАД

Рисунок 4.26. Результат оцінки діяльності викладачів по ключовим словам

Дана програма розроблена таким чином, що якщо в одного з викладачів не буде взагалі робіт, то система про це повідомить і цей викладач не буде впливати на відсоток збігу всієї групи викладачів (рисунк 4.27).

| | Прізвище | Ім'я | По-батькові | Surname | Name | Middlename | К-сть робіт(2018 - 2021 роки) | % |
|---|--------------|---------|---------------|---------------|---------|------------|-------------------------------|-------------|
| 1 | Морозов | Віктор | Володимирович | None | None | None | 5 | 89.64 |
| 2 | Лукова-Чуйко | Наталія | Вікторівна | Lukova-Chuiko | Natalia | Viktorivna | 0 | Немає робіт |

Співпадіння робіт до напрямку діяльності науковців дорівнює 89.64%

НАЗАД

Рисунок 4.27. Результат оцінки діяльності, якщо у одного з викладачів немає наукових робіт

Програмний продукт проаналізував викладачів за ключовими словами та абстрактів робіт цих авторів. Якщо деякі роботи не мали абстракту, то додатково проводив аналіз за назвами наукових праць.

4.3 Експеримент у реальному часі

Проведемо експеримент розробленої системи, взявши за приклад одного викладача. Програма здійснить збір робіт за певний проміжок часу і на основі цих даних проведе оцінку діяльності викладача по ключових словах.

Отже, для аналізу візьмемо викладача Малашонка Геннадія Івановича, на початку роботи з програмою вводимо ПІБ автора. Після того, як програма перевірила наявність даного викладача додаємо інформацію англійською мовою.

Вписуємо “Malaschonok Gennadi Ivanovich”. Проведемо аналіз з 2016 по 2021 роки. Після парсингу даних програма знайшла і зберегла 10 робіт. Перевіривши роботи, можна побачити, що майже всі вони англійською мовою (рисунок 4.28).

тому аналіз доцільніше всього провести з ключовими словами англійською мовою.

НУБІП України

Всі роботи

| | Прізвище | Ім'я | По-батькові | Surname | Name | Middlename |
|---|-----------|----------|-------------|-------------|---------|------------|
| 1 | Малашонко | Геннадій | Іванович | Malaschonok | Gennadi | Ivanovich |

| | Назва роботи | Автори | |
|----|---|---|---|
| 1 | A Basic Result on the Theory of Subresultants | Akritas Alkiviadis G, Malaschonok Gennadi I, Vigiaklas Panagiotsis S. | Given the polynomials $f, g \in \mathbb{Z}[x]$ th |
| 2 | Subresultant Polynomial Remainder Sequences Obtained by Polynomial Divisions in $\mathbb{Q}[x]$ or in $\mathbb{Z}[x]$ | Akritas Alkiviadis G, Malaschonok Gennadi I, Vigiaklas Panagiotsis S. | In this paper we present two new m |
| 3 | Effective Matrix Methods in Commutative Domains | Malaschonok Gennadi | Effective matrix methods for solving |
| 4 | Algorithms for the Computing Determinants in Commutative Rings | Malaschonok Gennadi | Two known computation methods a |
| 5 | Algorithms for the solution of systems of linear equations in commutative ring | Malaschonok Gennadi | Solution methods for linear equation |
| 6 | Recursive Method for the Solution of Systems of Linear Equations | Malaschonok Gennadi | New solution method for the system |
| 7 | Розподілені обчислення: ДАП-технологія розпаралелювання рекурсивних алгоритмів | Malaschonok Gennadi Sidko Alla | This article addresses the description |
| 8 | Recursive Matrix Algorithms in Commutative Domain for Cluster with Distributed Memory | Ilchenko Eugeni Malaschonok Gennadi | We give an overview of the theoreti |
| 9 | LDU factorization | Malaschonok Gennadi | LU-factorization of matrices is one of |
| 10 | Computing with quasiseparable matrices | Chan T. F, Malaschonok G. I. +3 MORE | International audience The class of q |

Рисунок 4.28. Роботи Малашонка Г.І. за 2016 – 2021 роки

Наступним кроком є введення ключових слів діяльності, якою займається автор. Даний викладач працює на кафедрі “Математика та інформатика”.

Введемо такі ключові слова: “mathematics, math, computer science, informatics, technology, method, algorithms, matrix, theorem, IT, systems, web, project”.

Після чого отримуємо результат (рисунок 4.29)

| | | | | | | |
|---|---------------------------|--|--|-------------|----|-------|
| 4 | Математика та Інформатика | Малашонко Геннадій Іванович, Malaschonok Gennadi Ivanovich | mathematics, math, computer science, informatics, technology, method, algorithms, matrix, theorem, IT, systems, web, project | 2016 - 2021 | 10 | 86.14 |
|---|---------------------------|--|--|-------------|----|-------|

Рисунок 4.29. Результат оцінки діяльності Малашонка Г.І.

Відсоток схожості напрямку діяльності викладача до його робіт дорівнює 86.14%. Це досить високий результат, який показує, що викладач має роботи, які відносяться до того напрямку, яким він займається.

Програма довела, що вона може працювати, як з текстами українською мовою, так і англійською. Це допомагає розширити можливості даного програмного продукту.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВИСНОВКИ

НУБІП України

У ході виконання даного дипломного проекту був спроектований програмний додаток, який може збирати інформацію про викладачів з бібліографічного джерела, зберігати дані, аналізувати і проводити оцінку діяльності викладачів за двома критеріями: за кількістю робіт протягом років і за ключовими словами. Завдання, які були вирішені:

- проведений аналіз подібних систем, дана система є унікальною, такий аналіз і оцінку діяльності викладачів на даний момент не проводить жодна програма;
- за допомогою Selenium WebDriver налаштований швидкий збір інформації з міжнародного бібліографічного джерела;
- розроблена реляційна база даних, яка зберігає в собі всю потрібну інформацію;
- проведений аналіз різних метрик порівняння слів. Подібність Джаро-Вінклера найбільше підійшла для даного проекту;
- використовуючи метрику подібності слів, розроблений алгоритм порівняння кожного слова абстракту наукових робіт з кожним ключовим словом напрямку діяльності;
- отримані результати представлені у вигляді програми, з функціональним інтерфейсом, який містить в собі зручні таблиці і приємний дизайн.

Результати роботи над даним проектом показали, що всі початкові цілі були досягнуті. Також в дипломній роботі детально розписана інструкція запуску програми і подальшого користування з нею.

Дана програма допоможе проконтролювати викладачів і направити їх на зміну тем наукових робіт або ж взагалі дасть зрозуміти, що даний викладач не підходить для даної спеціальності чи спеціалізації.

НУБІП України

Відсоток збігу групи викладачів дає оцінку організації. Якщо цей відсоток низький, то потрібно переглянути процес роботи і направити його на подальшу зміну тем наукових робіт, щоб вони відповідали напрямку діяльності, якою займається та чи інша організація.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Google Scholar [Електронний ресурс] – Режим доступу до ресурсу: <https://scholar.google.com/>.
2. Scopus [Електронний ресурс] – Режим доступу до ресурсу: <https://www.scopus.com/>.
3. Solomon D. Types of Open Access Publishers in Scopus [Електронний ресурс] / David Solomon. – 2013. – Режим доступу до ресурсу: <https://doi.org/10.3390/publications1010016>.
4. ArXiv [Електронний ресурс] – Режим доступу до ресурсу: <https://arxiv.org/>
5. Core [Електронний ресурс] – Режим доступу до ресурсу: <https://core.ac.uk/>
6. Диго С. М. Базы данных. Проектирование и создание / С. М. Диго. – Москва: Изд. центр ЕАОИ, 2008. – 171 с.
7. Si Alhir S. Learning UML / Sinan Si Alhir., 2003. – 304 с.
8. What is MySQL? [Електронний ресурс] // MySQL – Режим доступу до ресурсу: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>.
9. Schwartz B. High Performance MySQL: Optimization, Backups, and Replication / B. Schwartz, P. Zaitsev, V. Tkachenko., 2012. – 826 с.
10. Rana Y. Python: Simple though an Important Programming language [Електронний ресурс] // Yogesh Rana // IRJET. – 2019. – Режим доступу до ресурсу: <https://www.irjet.net/archives/V6/i2/IRJET-V6I2367.pdf>.
11. Python Introduction [Електронний ресурс] // Google for Education. – 2022. – Режим доступу до ресурсу: <https://developers.google.com/edu/python/introduction?hl=ru>.
12. Lutz M. Learning Python, 3rd Edition [Електронний ресурс] // Mark Lutz // O'Reilly Media. – 2007. – Режим доступу до ресурсу: <https://www.oreilly.com/library/view/learning-python-3rd/9780596513986/ch01.html>.
13. PyCharm Pro vs PyCharm Community [Електронний ресурс] // JetBrains – Режим доступу до ресурсу:

<https://www.jetbrains.com/products/compare/?product=pycharm&product=pycharm-ce>.

14. Дронов В. А. Python 3 и PyQt 5. Разработка приложений / В. А. Дронов, Н. А. Прохоренок. – Санкт-Петербург: БХВ-Петербург, 2016. – 832 с.

15. Gundecha U. Selenium WebDriver 3 Practical Guide: End-to-end automation testing for web and mobile browsers with Selenium WebDriver / U. Gundecha, S. Avasarala., 2018. – 280 с.

16. Cahyono S. C. Comparison of document similarity measurements in scientific writing using Jaro-Winkler Distance method and Paragraph Vector method

[Електронний ресурс] / S. C. Cahyono. – 2019. – Режим доступу до ресурсу:

[https://www.researchgate.net/publication/337401647-Comparison_of_document_similarity_measurements_in_scientific_writing_using_Jaro-](https://www.researchgate.net/publication/337401647-Comparison_of_document_similarity_measurements_in_scientific_writing_using_Jaro-Winkler_Distance_method_and_Paragraph_Vector_method)

[Winkler_Distance_method_and_Paragraph_Vector_method.](https://www.researchgate.net/publication/337401647-Comparison_of_document_similarity_measurements_in_scientific_writing_using_Jaro-Winkler_Distance_method_and_Paragraph_Vector_method)

17. Rodichevski A. Approximate string-matching algorithms [Електронний

ресурс] / Alexandre Rodichevski. Режим доступу до ресурсу:

[https://www.morfoedro.it/doc.php?n=223&lang=en#Resources.](https://www.morfoedro.it/doc.php?n=223&lang=en#Resources)

18. Левенштейн В. І. Двійкові коди з виправленням випадань, вставок та заміщень символів. Доповіді Академії наук СРСР, 1965 (Levenshtein V. I.

Dvijkovij kodij z vypravleniam vypan, vstavok ta zamishchen symboliv.

Dopovidj Akademij nauk SRSR, 1965).

19. Damerau F. A technique for computer detection and correction of spelling errors, 1964. Commun. ACM 7, 3, 171–176.

20. Hamming R. S. Error detecting and error correcting codes. – Bell Syst. Tech, 1950, 147-160.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України