ДОДАТОК А

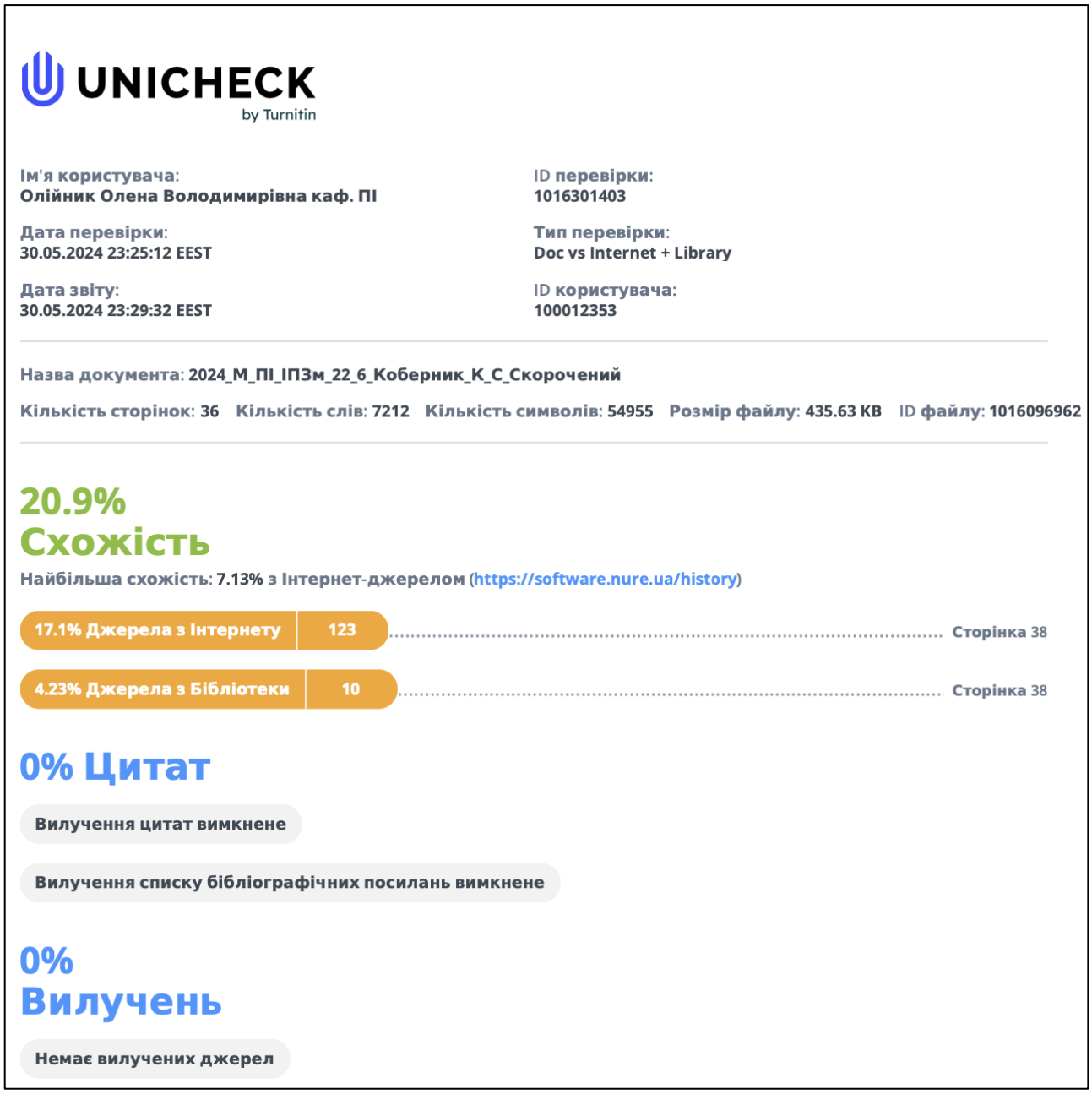
Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії

10. Igor Shubin, Andrii Kozyriev, Method for Solving Quantifier Linear Equations for Formation of Optimal Queries to Databases in: Computational Linguistics and Intelligent Systems 2023, Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems. vol. 449-459

11. Victoria Vysotska, Ihor Shubin, Maksym Mezentsev, Karen Kobernyk, Grygoryy Chetverikov, Ukrainian Big Data: The Problem of Databases Localization in: Intelligent Systems Workshop on 8th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2024), vol. 122-133.

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



UNICHECK
by Turnitin

Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016301403

Дата перевірки:
30.05.2024 23:25:12 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
30.05.2024 23:29:32 EEST

ID користувача:
100012353

Назва документа: 2024_М_ПІ_ІПЗм_22_6_Коберник_К_С_Скорочений

Кількість сторінок: 36 **Кількість слів:** 7212 **Кількість символів:** 54955 **Розмір файлу:** 435.63 KB **ID файлу:** 1016096962

20.9%
Схожість

Найбільша схожість: 7.13% з Інтернет-джерелом (https://software.nure.ua/history)

17.1% Джерела з Інтернету | 123 ..... Сторінка 38

4.23% Джерела з Бібліотеки | 10 ..... Сторінка 38

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

ДОДАТОК В

Слайди презентації

ДОСЛІДЖЕННЯ МЕТОДІВ СТВОРЕННЯ ТА СУПРОВОДУ МЕРЕЖЕВИХ БАЗ ДАНИХ

Виконав:
ст. гр. ІПЗм-22-6 Коберник К. С.

Керівник:
проф. Шубін І. Ю.

АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

- Сучасні додатки генерують та обробляють величезні обсяги даних, що потребує ефективного зберігання і швидкого доступу. NoSQL бази даних, такі як MongoDB, Cassandra та Redis, забезпечують горизонтальне масштабування і високу продуктивність для великих наборів даних. Python, завдяки своїм численним бібліотекам та простоті інтеграції з різними базами даних, є ідеальним інструментом для роботи з такими технологіями.
- NoSQL бази даних пропонують гнучкість у моделюванні даних, що дозволяє розробникам швидко адаптувати структуру бази даних до змін у вимогах проекту. Python, з його легкою читабельністю та широким спектром фреймворків (наприклад, Django, Flask), сприяє швидкій розробці та тестуванню прототипів. Це особливо важливо у стартапах та динамічних проектах, де швидкість виходу на ринок має вирішальне значення.
- Python є однією з найпопулярніших мов програмування у світі, з великою та активною спільнотою розробників. Це забезпечує постійну підтримку, обмін знаннями та швидкий розвиток інструментів і бібліотек. Використання NoSQL архітектури також стає все більш поширеним у сучасних додатках, від великих корпорацій до стартапів. Така популярність обох технологій гарантує наявність численних ресурсів для навчання, обговорення проблем та обміну досвідом.

АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ

Мета - Мета даного дослідження полягає в розробці нових методів та інструментів для створення та супроводу мережевих баз даних, які відповідають вимогам сучасного інформаційного середовища

Ось ключові аспекти, що варто розглядати:

- мови запитань: Декларативні та імперативні мови: Важливо вибрати мову запитань, яка забезпечить зручний і ефективний доступ до даних. В багатьох мережевих СУБД використовують декларативні мови, які описують, які дані потрібно отримати, тому дуже важливим рішенням буде створити мову запитів аналогічну тим що існують на ринці але додати розширенні функції налаштування методів запиту;
- масштабованість: Горизонтальна та вертикальна масштабованість: Забезпечення ефективної роботи з базою даних при збільшенні обсягу даних є критичним. Важливо вибрати рішення, яке дозволяє масштабувати базу даних горизонтально (додавання нових серверів) та вертикально (збільшення потужності наявних серверів);
- моделювання взаємозв'язків: Використання вказівників у обидві сторони: Однією з ключових особливостей мережевої моделі є здатність висловлювати складні взаємозв'язки між записами. Важливо враховувати, наскільки ефективно створена СУБД буде їх встановлювати;

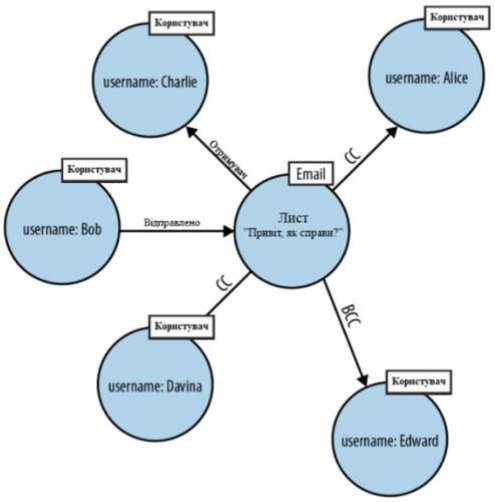
ПОСТАНОВКА ЗАДАЧІ

- Провести дослідження існуючих мережевих баз даних та обрати технології і програмні рішення які будуть актуальними для реалізації гнучкого програмного застосунку та аналіз методів інтеграції мережевих баз даних
- Збудувати архітектуру програмного забезпечення з урахуванням визначеної бізнес моделі, дослідження методів створення нереляційних баз даних у вигляді застосунків
- Створити програмне забезпечення орієнтоване на реалізацію нереляційних баз даних за мережевою моделлю даних

ВИКОРИСТАННЯ МБД З NOSQL

У сучасному середовищі noSQL бази даних стають все популярнішими, особливо для роботи з великими обсягами неструктурованих даних.

Мережеві бази даних у noSQL використовують графову модель для представлення та зберігання даних



ВИБІР БАЗОВОЇ ТЕХНОЛОГІЇ ДЛЯ СТВОРЕННЯ МБД

Язики Програмування на Бекенді, де відбувається обробка даних та їх взаємодія з базами даних, часто використовуються такі мови програмування, як:

- JavaScript (Node.js): Популярний через свою універсальність, оскільки JavaScript також використовується на фронтенді та для нього написано багато бібліотек та фреймворків для візуалізації та надання контенту інтерактивності або для обробки даних на бекенді;
- Python: Завдяки своїй читабельності та великому набору бібліотек майже для будь-якої програмної системи, Python є відмінним вибором для розробки веб-додатків;
- Java: Використовується в багатьох великих корпоративних середовищах завдяки своїй масштабованості та надійності;
- PHP: Широко використовується для розробки веб-сайтів та веб-додатків, часто у поєднанні з MySQL;
- Ruby (Ruby on Rails): Відомий своєю швидкістю розробки та чистотою коду але вже дуже застарілий.

ПРИКЛАДИ NOSQL БАЗ ДАНИХ

- MongoDB: Це одна з найпопулярніших документоорієнтованих NoSQL баз даних. Вона зберігає дані у форматі, схожому на JSON, що робить її дуже гнучкою для роботи з великими обсягами даних;
- Cassandra: Ця база даних відома своєю високою масштабованістю та надійністю. Cassandra є колоноорієнтованою системою, яка добре підходить для обробки великих обсягів даних з високою швидкістю запису та читання;
- Redis: Це система управління базами даних типу "ключ-значення", яка часто використовується як система кешування та зберігання сесій у веб-додатках через свою високу швидкість та ефективність;
- Couchbase: Подібно до MongoDB, Couchbase є документоорієнтованою NoSQL базою даних. Вона пропонує гнучкість та масштабованість для розподілених веб-додатків і часто використовується в мобільних застосунках;
- Apache HBase: Ця база даних є нереляційною колоноорієнтованою та розподіленою, розробленою на основі Google's Bigtable. HBase часто використовується для великих обсягів даних, особливо у випадках, коли необхідний випадковий доступ до даних.

ОПИС АЛГОРИТМІВ ПОШУКУ У МБД

Алгоритми пошуку в NoSQL базах даних відрізняються від традиційних реляційних баз даних через їхню унікальну структуру та способи зберігання даних. Оскільки NoSQL бази даних часто використовують неструктуровані або напів-структуровані дані, алгоритми пошуку мають бути гнучкими та ефективними, щоб впоратися з різноманітністю форматів даних.

Основні Особливості Пошуку у NoSQL Базах Даних є Індексація: Як і в реляційних БД, індексація грає ключову роль у пошуку NoSQL.

Індекс працює подібно до індексу в книзі. Він створюється для одного або декількох полів у таблиці, що значно скорочує час пошуку, оскільки база даних може швидко знаходити місце розташування даних, на які є посилання в індексі, замість того, щоб переглядати кожен рядок таблиці.

АРХІТЕКТУРА ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ

Управління Конфігурацією: Правильне управління конфігурацією мережевих БД включає налаштування параметрів зберігання даних, мережевих налаштувань та параметрів безпеки. Наведені Підсхеми мають примітивне уявлення про структуру БД і це дозволяє програмній системі відправляти добре сформований запит далі, саме ці підсхеми потрібно регулярно оновлювати для корректної роботи застосунку.

ОПИС ПРИЙНЯТИХ ПРОЄКТНИХ РІШЕНЬ

Мова програмування: Вибір Python був обумовлений його простотою, широким спектром бібліотек і модулів, а також великою спільнотою розробників, що забезпечує підтримку і швидке вирішення проблем.

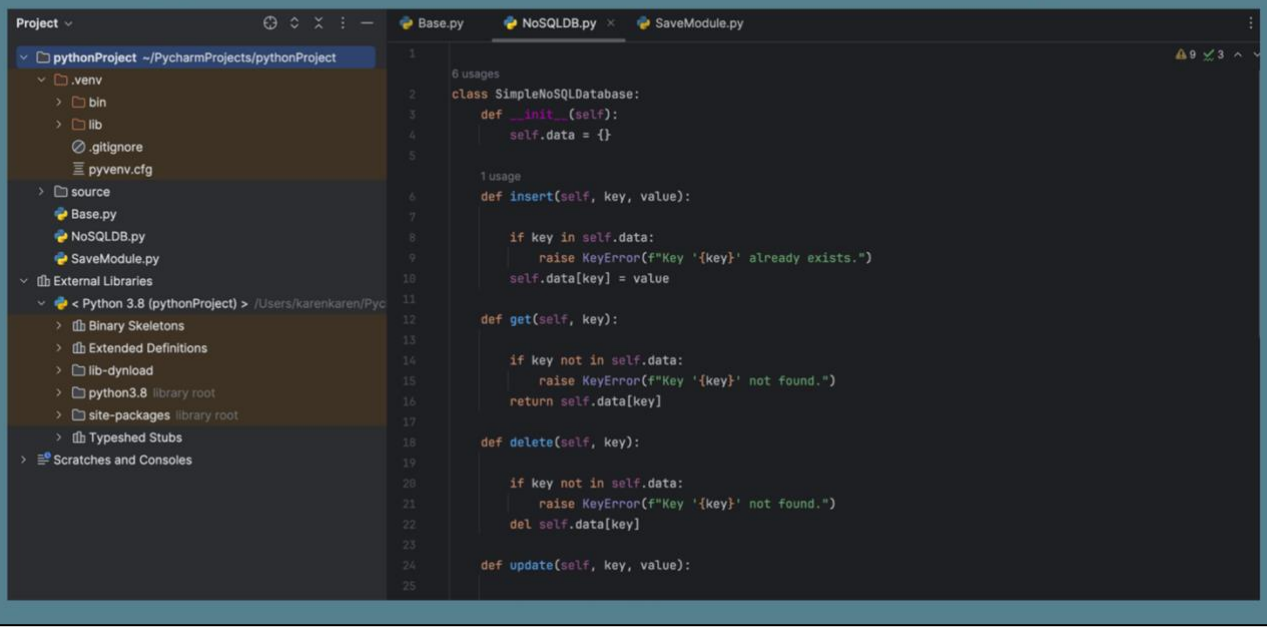
Requests: Обраний для роботи з HTTP-запитами, оскільки забезпечує простий і зручний спосіб взаємодії з веб-ресурсами, підтримуючи всі методи HTTP та пропонуючи зручний інтерфейс для відправки запитів і обробки відповідей.

Tkinter: Використовується для створення графічного інтерфейсу користувача (GUI). Це стандартна бібліотека Python для створення GUI-додатків, яка забезпечує простоту у використанні та потужні можливості для побудови інтерфейсів.

OS: Застосовується для взаємодії з операційною системою, включаючи роботу з файловою системою, що дозволяє програмі виконувати такі дії, як зчитування та запис файлів, а також отримання інформації про файлову структуру.

Розробка в PyCharm: Інтегроване середовище розробки (IDE): Вибір PyCharm обумовлений його потужними інструментами для розробки Python-програм, такими як інтегрований налагоджувач, підтримка тестування, автозавершення коду та аналіз якості коду.

МОДУЛЬ РЕАЛІЗОВАНОЇ МЕРЕЖЕВОЇ БАЗИ ДАНИХ



ФУНКЦІЯ ОТРИМАННЯ ДАНИХ ЧЕРЕЗ API ДЛЯ ВНЕСЕННЯ В МБД

```
def call_api(var1, var2, var3):
    global drecieved
    url = var1
    db2 = NoSQLDB.SimpleNoSQLDatabase()
    querystring = var2
    headers = var3
    response = requests.get(url, headers=headers, params=querystring)
    recieved = response.text
    drecieved = response.text
    db2.load_text(drecieved)
    result = f"Your imported row is: {recieved}"
    messagebox.showinfo( title: "Result", result, parent=new_window)
```

ОПИС СФОРМОВАНИХ РЕКОМЕНДАЦІЙ

- Обирати мережеву базу даних потрібно тільки для проведення нестандартних операцій або задач для збору статистики, де ввони будуть найефективнішими у використанні
- Такі мовии програмування як Python дуже полегшують створення такого роду застосунки через наявність бібліотек та швидкість обчислення
- Мережеві NoSQL бази даних ідеально працюють у завданнях статистики та аналізу

ВИСНОВКИ

- Масштабованість: NoSQL бази даних, як правило, краще масштабуються ніж традиційні реляційні бази даних. Вони можуть ефективно обробляти величезні обсяги даних та високі навантаження на читання/запис, що особливо важливо для великих мережевих систем;
- Гнучкість схеми: МБД часто не вимагають жорсткої схеми, що дозволяє більш гнучко працювати з різними типами даних. Це особливо корисно в динамічних середовищах, де структура даних може часто змінюватися;
- Висока доступність та відмовостійкість: багато МБД пропонують вбудовані рішення для реплікації даних та відновлення після збоїв, що забезпечує високу доступність та відмовостійкість даних;
- Оптимізація для специфічних випадків використання: різні типи NoSQL баз даних (документ-орієнтовані, графові, ключ-значення) оптимізовані для конкретних випадків використання, що дозволяє вибрати найбільш підходящу систему для конкретних задач;
- Простота горизонтального масштабування: МБД майже усі розроблені з урахуванням горизонтального масштабування, що дозволяє додавати більше серверів для обробки збільшеного обсягу даних;
- Ефективність для неструктурованих та напівструктурованих даних: МБД ефективно працюють з неструктурованими та напівструктурованими даними, що є важливим у сучасних додатках, де часто використовуються дані такого типу;
- Швидкість обробки запитів: для деяких типів запитів МБД можуть пропонувати кращу продуктивність порівняно з традиційними реляційними системами.

ДОДАТОК Г

Апробація результатів роботи

Victoria Vysotska, Ihor Shubin, Maksym Mezentsev, Karen Kobernyk, Grygoryy Chetverikov, Ukrainian Big Data: The Problem of Databases Localization in Intelligent Systems Workshop on 8th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2024), сторінки 122-133. (https://ceur-ws.org/Vol-3688/paper9.pdf)

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент (посада) | програмної інженерії (кафедра) | ІПЗм-22-6 (група)

Коберник Карен Сергійович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні познаки	
	7.15 Додатки	

зауважень немає

Експерт _____ (підпис) Олена ОЛІЙНИК (прізвище, ініціали)

14.06.2024