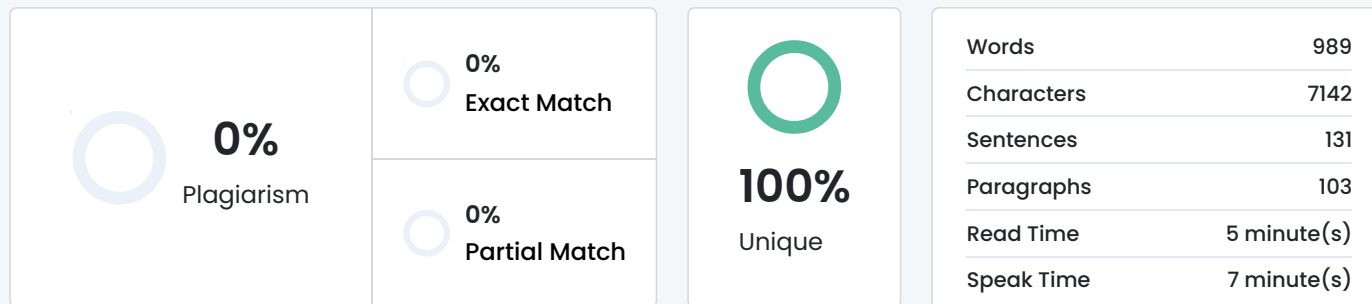


Plagiarism Scan Report



Content Checked For Plagiarism

4.2 Налаштування підключення до бази даних

- `server` – змінна, яка має містити назву сервера бази даних (може бути локальний сервер або хмарний).
- `database` – назва бази даних, у якій зберігається інформація про книги.

Рис.4.2.1 – Назва бази даних та серверу

4.3 Функція для встановлення підключення до бази даних

Функція `get_connection()` створює та повертає об'єкт підключення до бази даних.

Основні параметри підключення:

- `DRIVER={{ODBC Driver 17 for SQL Server}}` – вказує драйвер ODBC для SQL Server.
- `SERVER={server}` – змінна, яка містить адресу або ім'я сервера.
- `DATABASE={database}` – змінна, яка містить назву бази даних (BooksDB).
- `Trusted_Connection=yes` – дозволяє використовувати автентифікацію Windows для підключення.
- `Encrypt=yes` – активує шифрування з'єднання для підвищення безпеки.
- `TrustServerCertificate=yes` – дозволяє використовувати самопідписані сертифікати при підключенні.

Ця функція використовується в усіх інших частинах програми для встановлення зв'язку з базою даних.

Рис.4.3.1 – Підключення до бази даних

4.4 Функція для додавання книги до бази даних

Перевірка введених даних:

- Перед виконанням запиту перевіряється, чи всі необхідні дані передані у функцію.
- Якщо хоча б одне поле не заповнене, програма виведе повідомлення про помилку і припинить виконання функції.

Рис.4.4.1 – Функція додавання книги

Виконання SQL-запиту:

- Використовується контекстний менеджер `with`, щоб забезпечити автоматичне закриття підключення після виконання операції.
- Формується SQL-запит для вставки нової книги до таблиці Books.
- Використовуються параметризовані запити (?), які допомагають уникнути SQL-ін'єкцій.
- `conn.commit()` – підтверджує внесення змін у базу даних.

Рис.4.4.2 – Вигляд SQL-запиту

Обробка помилок:

- Якщо виникає помилка, вона виводиться у консоль.

Рис.4.4.3 – Обробка помилок

4.5 Функція для отримання рекомендованих книг із бази даних

Імпорти та клас:

- Ця функція шукає книги у базі за вказаним жанром.
- Якщо вказано `min_rating`, запит доповнюється фільтром за мінімальним рейтингом.

Рис.4.5.1 – Функція отримання рекомендованих книг

Фільтрація за рейтингом:

- Якщо `min_rating` не передано, беруться всі книги вказаного жанру.

Рис.4.5.2 – Фільтрація за рейтингом

Виконання запиту та отримання результату:

- `cursor.fetchall()` повертає всі знайдені книги у вигляді списку.

Рис.4.5.3 – Отримання результату

4.6 Функція для отримання всіх книг із бази даних

Функція `get_all_books_from_db()` виконує запит до бази даних і повертає список усіх книг, які є в таблиці `Books`.

Алгоритм роботи:

1. Встановлення підключення до бази даних:
 - Використовується контекстний менеджер `with get_connection() as conn`, який гарантує, що з'єднання буде автоматично закрито після завершення роботи.
2. Відправлення SQL-запиту:
 - Виконується запит `SELECT`, який отримує всі записи з таблиці `Books`.
 - У запиті вибираються лише конкретні поля: `Title` (назва), `Author` (автор), `Genre` (жанр), `Year` (рік) і `Rating` (рейтинг).
3. Отримання та повернення результатів:
 - Метод `fetchall()` повертає список усіх рядків, отриманих у результаті запиту.
 - Функція повертає цей список, щоб можна було використовувати його в інших частинах програми.
4. Обробка помилок:

Функція включає два блоки `except` для перехоплення можливих помилок:

- `pyodbc.Error` – для обробки помилок, пов'язаних із SQL-запитами та підключенням до бази.
- 5. У разі помилки виводиться відповідне повідомлення, а функція повертає порожній список `[]`.
- `Exception` – загальний обробник для інших можливих винятків, не пов'язаних безпосередньо з `pyodbc`.

Функція `get_all_books_from_db()` дозволяє отримати список усіх книг із бази даних та забезпечує стабільність роботи завдяки обробці можливих помилок. Вона використовується для виведення повного каталогу книг у графічному інтерфейсі або в консольному виведенні.

Рис.4.6.1 – Функція для отримання всіх книг із бази даних

4.7 Функція для оновлення інформації про книгу в базі даних

Функція `update_book_in_db(book_id, title=None, author=None, genre=None, year=None, rating=None)`

дозволяє оновлювати інформацію про книгу в базі даних. Вона приймає `book_id` як обов'язковий параметр, а всі інші параметри є необов'язковими (`None` за замовчуванням), що дозволяє оновлювати лише певні поля.

Алгоритм роботи:

1. Встановлення підключення до бази даних
 - Використовується контекстний менеджер `with get_connection() as conn`, щоб автоматично закрити з'єднання після завершення операції.
2. Формування SQL-запиту:
 - Спочатку створюється базовий запит.
 - Створюється список параметрів `params = []`, який буде містити значення для оновлення.
3. Додавання змінних у SQL-запит:
 - Якщо передано значення для `title`, `author`, `genre`, `year`, `rating`, вони додаються до запиту.
 - Аналогічно додаються інші поля.
4. Видалення зайвої коми:
 - Оскільки додаємо змінні динамічно, у кінці запиту може бути зайва кома.
 - Додавання умови `WHERE` для конкретного запису
 - Вказуємо, що оновлення має стосуватися лише книги з певним `id`:

5. Виконання SQL-запиту

- Передаємо сформований запит і список параметрів у `cursor.execute()`.
- Використовується підстановка `?`, що захищає від SQL-ін'єкцій.

6. Збереження змін:

- Виконується `conn.commit()`, щоб зміни були застосовані в базі даних.

7. Обробка помилок:

- Якщо виникає помилка SQL-запиту, виводиться відповідне повідомлення.
- Якщо виникає інша невідома помилка, вона теж обробляється.

Функція `update_book_in_db()` дозволяє оновлювати дані про книги в базі без необхідності змінювати всі поля.

Завдяки цьому можна, наприклад, змінити тільки рейтинг книги або її автора, залишивши інші дані без змін.

Рис.4.7.1 – Функція для оновлення інформації про книгу в базі даних

4.8 Функція для видалення книги з бази даних

Функція `delete_book_from_db(book_id)` видаляє книгу з бази даних за її унікальним ідентифікатором `book_id`.

Вона встановлює з'єднання з базою, виконує SQL-запит для видалення відповідного запису, перевіряє успішність операції та обробляє можливі помилки.

Алгоритм роботи:

1. Встановлення з'єднання з базою даних:

- Викликається функція `get_connection()`, яка повертає підключення до бази.
- Використовується `conn.cursor()`, що дозволяє виконувати SQL-запити.

2. Формування та виконання SQL-запиту:

- Запит `DELETE FROM Books WHERE id = ?` використовується для видалення книги, де `?` замінюється на значення `book_id`.

- Передача параметрів окремо `((book_id,))` захищає від SQL-ін'єкцій.

3. Підтвердження змін у базі:

- Викликається `conn.commit()`, щоб зміни набули чинності.

4. Перевірка успішності операції:

- Якщо `cursor.rowcount == 0`, тобто жоден запис не був видалений, виникає помилка `ValueError`, що означає, що книги з таким `id` у базі не існує.

5. Закриття курсору та з'єднання:

- `cursor.close()` та `conn.close()` закривають об'єкти, щоб звільнити ресурси бази даних.

6. Обробка помилок:

- Якщо під час роботи функції виникає будь-яка помилка, вона перехоплюється у блоці `except`.
- Створюється новий виняток із повідомленням про помилку, щоб користувач отримав зрозумілу інформацію про проблему.

Функція забезпечує безпечне видалення книги з бази даних і гарантує, що користувач отримає повідомлення у випадку, якщо книгу не знайдено або виникла інша помилка.

Рис.4.8.1 – Функція для видалення книги з бази даних

Matched Source

No plagiarism found