
Dokumentacja projektu: Aplikacja Biblioteka (Python + SQLite + Tkinter)

1 Opis funkcjonalności aplikacji

Aplikacja „Biblioteka” umożliwia zarządzanie bazą danych książek i autorów. Główne funkcje aplikacji to:

- Dodawanie nowych książek do bazy danych.
- Edycja istniejących książek.
- Usuwanie książek z bazy danych.
- Wyszukiwanie książek według tytułu lub autora.
- Sortowanie książek według tytułu lub autora.
- Eksport danych do pliku CSV.
- Import danych z pliku CSV.
- Prezentacja danych w graficznym interfejsie użytkownika (Tkinter).

Aplikacja wykorzystuje bazę danych SQLite i zapewnia operacje CRUD (Create, Read, Update, Delete) na tabelach **books** i **authors**.

2 Opis struktury bazy danych

Baza danych składa się z dwóch tabel:

Tabela authors

Kolumna	Typ	Opis
id	INTEGER PRIMARY KEY AUTOINCREMENT	Identyfikator autora
name	TEXT NOT NULL UNIQUE	Imię i nazwisko autora

Tabela books

Kolumna	Typ	Opis
id	INTEGER PRIMARY KEY AUTOINCREMENT	Identyfikator książki
title	TEXT NOT NULL	Tytuł książki
author_id	INTEGER	Klucz obcy do tabeli authors

Relacja:

Każda książka (**books.author_id**) wskazuje na jednego autora (**authors.id**).

Relacja typu wiele książek do jednego autora (N:1).

Diagram ERD



3 Kod SQL tworzący bazę danych

```
DROP TABLE IF EXISTS books;
DROP TABLE IF EXISTS authors;
```

```
CREATE TABLE authors (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT NOT NULL UNIQUE
);
```

```
CREATE TABLE books (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  title TEXT NOT NULL,
  author_id INTEGER,
  FOREIGN KEY (author_id) REFERENCES authors(id)
);
```

```
INSERT INTO authors (name) VALUES
```

```
('Henryk Sienkiewicz'),
('Adam Mickiewicz'),
('Bolesław Prus'),
('Maria Konopnicka'),
('Stanisław Lem');
```

```
INSERT INTO books (title, author_id) VALUES
('W pustyni i w puszczy', 1),
('Pan Tadeusz', 2),
('Lalka', 3),
('Rota', 4),
('Solaris', 5);
```

4 Kod źródłowy aplikacji (fragmenty)

Dodawanie książki

```
def add_book():
    title = entry_title.get()
    author_id = combo_author.get()
    if not title or not author_id:
        messagebox.showwarning("Brak danych", "Wpisz tytuł i wybierz autora.")
        return
    author_id = int(author_id.split(":")[0])
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute("INSERT INTO books (title, author_id) VALUES (?, ?)", (title,
author_id))
    conn.commit()
    conn.close()
    messagebox.showinfo("Sukces", f"Książka '{title}' została dodana.")
    show_records()
```

Eksport do CSV

```
def export_to_csv():
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute("""
        SELECT books.id, books.title, authors.name
        FROM books
        LEFT JOIN authors ON books.author_id = authors.id
    """)
    rows = cursor.fetchall()
    conn.close()
    with open('biblioteka_eksport.csv', 'w', newline="", encoding='utf-8') as f:
        writer = csv.writer(f)
```

```
writer.writerow(['ID', 'Tytuł', 'Autor'])
writer.writerows(rows)
messagebox.showinfo("Eksport zakończony", "Dane zostały zapisane do pliku
biblioteka_eksport.csv")
```

5 Testy jednostkowe

```
import sqlite3
import os
```

```
DB_NAME = 'biblioteka.db'
```

```
def test_database_connection():
    assert os.path.exists(DB_NAME), f"Plik bazy danych {DB_NAME} nie istnieje."
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute("SELECT name FROM sqlite_master WHERE type='table'")
    tables = cursor.fetchall()
    conn.close()
    assert len(tables) >= 2, "Baza powinna zawierać co najmniej 2 tabele."
```

6 Wygenerowana dokumentacja kodu

Dokumentacja wygenerowana za pomocą `pydoc` lub `help()` opisuje funkcje aplikacji:

- `add_book` — dodaje książkę do bazy,
 - `delete_book` — usuwa książkę,
 - `show_records` — wyświetla dane w GUI,
 - `export_to_csv` — eksportuje dane do pliku CSV,
 - `import_from_csv` — importuje dane z pliku CSV.
-

7 Instrukcja instalacji i uruchomienia

Wymagania

- Python 3.x

- Tkinter (wbudowany w Pythona)
- SQLite (wbudowany w Pythona)

Instalacja i uruchomienie

1 Sklonuj repozytorium:

```
git clone https://github.com/Vlodek/projekt_biblioteka.git
```

2 Wejdź do folderu:

```
cd projekt_biblioteka
```

3 Utwórz bazę danych (np. w konsoli sqlite3):

```
sqlite3 biblioteka.db < init_db.sql
```

4 Uruchom aplikację:

```
python app.py
```

5 Uruchom testy:

```
pytest test_app.py
```