TRƯỜNG ĐẠI HỌC HỌC VĂN LANG

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC

PYTHON NÂNG CAO

Giang viên: Huỳnh Thái Học

Sinh viên thực hiện:

PHIMMACHACK Lattakone - 227480201is001

TP. Hồ Chí Minh – năm 2024

Mục lục

MỞ ĐẦU

1.GIỚI THIỆU

Mục tiêu của đồ án: Xây dựng một ứng dụng web quản lý sách sử dụng Python Flask và PostgreSQL. Ứng dụng này giúp người dùng dễ dàng thêm, sửa, xóa, và xem thông tin sách trong một cơ sở dữ liệu.

2. CHỨC NĂNG CODE

2.1 Trang chính

- **Chức năng: Hiển thị danh sách các sách có trong cơ sở dữ liệu.**
- **Cách thức hoạt động: Lấy dữ liệu từ bảng books trong PostgreSQL và gửi về index.html để hiển thị. Sử dụng Bootstrap để tạo giao diện bảng (table) với bố cục rõ ràng.**
- **Công nghệ sử dụng: HTML, CSS và Bootstrap giúp giao diện thân thiện và dễ nhìn.**

2.2 Thêm sách (add_book):

- **Chức năng: Cho phép người dùng thêm sách mới vào cơ sở dữ liệu.**
- **Cách thức hoạt động: Người dùng nhập thông tin sách (tiêu đề, tác giả, năm xuất bản, thể loại) vào form. Flask kiểm tra nếu tất cả các trường đã được nhập đầy đủ, sau đó lưu vào bảng books.**
- **Xử lý lỗi: Nếu thông tin bị thiếu, thông báo lỗi sẽ hiển thị yêu cầu người dùng điền đầy đủ các trường.**

2.3 Sửa sách (update_book):

- **Chức năng: Cập nhật thông tin sách có sẵn.**
- **Cách thức hoạt động: Người dùng nhập thông tin mới cho một sách đã có và nhấn nút "Update". Flask cập nhật dữ liệu trong PostgreSQL với thông tin mới.**
- **Xử lý lỗi: Kiểm tra nếu tất cả các trường đã được điền, nếu không sẽ thông báo cho người dùng.**

2.4 Xóa sách (delete_book):

- **Chức năng: Xóa một cuốn sách khỏi cơ sở dữ liệu.**

- **Cách thức hoạt động: Người dùng nhấn nút "Delete" bên cạnh cuốn sách cần xóa. Flask xử lý yêu cầu và xóa sách khỏi bảng books trong PostgreSQL.**

2.5 Các hàm kết nối và tạo bảng trong mã nguồn:

- **Connect_db(): Kết nối tới PostgreSQL. Sử dụng thông tin đăng nhập như tên cơ sở dữ liệu, người dùng, mật khẩu và cổng kết nối.**
- **Create_table(): Tạo bảng books nếu bảng chưa tồn tại, với các cột:**
    - **id: Tự động tăng, là khóa chính.**
    - **title: Tên sách, không được để trống.**
    - **author: Tác giả, không được để trống.**
    - **year: Năm xuất bản, phải là số nguyên.**
    - **genre: Thể loại sách, không được để trống.**

3. MÃ NGUỒN

- **Mã nguồn chính của ứng dụng (Python và Flask):**
    - **app.py:**

```
from flask import Flask, render_tem-
plate, request, redirect, url_for, flash
import psycopg2

app = Flask(__name__)
app.secret_key = 'your_secret_key'

# Database connection
def connect_db():
    try:
        conn = psycopg2.connect(
```

```
            dbname="Lattakone",
            user="postgres",
            password="1234",
            host="localhost",
            port='5432'
        )
        return conn
    except Exception as e:
        print("Error while connecting to
database:", e)
        raise

def create_table():
    conn = connect_db()
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS books
(
            id SERIAL PRIMARY KEY,
            title VARCHAR(255) NOT NULL,
            author VARCHAR(255) NOT
NULL,
            year INTEGER NOT NULL,
            genre VARCHAR(100) NOT NULL
        );
```

```
    """)
    conn.commit()
    cursor.close()
    conn.close()

@app.route('/')
def index():
    conn = connect_db()
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM
books")
    books = cursor.fetchall()
    cursor.close()
    conn.close()
    return render_template('index.html',
books=books)

@app.route('/add', methods=['POST'])
def add_book():
    title = request.form['title']
    author = request.form['author']
    year = request.form['year']
    genre = request.form['genre']
```

```
    if title and author and year and
genre:
        conn = connect_db()
        cursor = conn.cursor()
        cursor.execute("INSERT INTO
books (title, author, year, genre) VAL-
UES (%s, %s, %s, %s)",
                       (title, author,
year, genre))
        conn.commit()
        cursor.close()
        conn.close()
        flash('Book added success-
fully!', 'success')
    else:
        flash('Please fill in all
fields!', 'error')

    return redirect(url_for('index'))

@app.route('/update/<int:book_id>',
methods=['POST'])
def update_book(book_id):
    title = request.form['title']
    author = request.form['author']
```

```
    year = request.form['year']
    genre = request.form['genre']

    if title and author and year and
genre:
        conn = connect_db()
        cursor = conn.cursor()
        cursor.execute("UPDATE books SET
title=%s, author=%s, year=%s, genre=%s
WHERE id=%s",
                       (title, author,
year, genre, book_id))
        conn.commit()
        cursor.close()
        conn.close()
        flash('Book updated success-
fully!', 'success')
    else:
        flash('Please fill in all
fields!', 'error')

    return redirect(url_for('index'))

@app.route('/delete/<int:book_id>')
def delete_book(book_id):
```

```
    conn = connect_db()
    cursor = conn.cursor()
    cursor.execute("DELETE FROM books WHERE id=%s", (book_id,))
    conn.commit()
    cursor.close()
    conn.close()
    flash('Book deleted successfully!', 'success')
    return redirect(url_for('index'))

if __name__ == '__main__':
    create_table()
    app.run(debug=True)

```

- **Giao diện người dùng (HTML, CSS):**
  - **index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Book Management</title>
    <link href="https://cdn.jsde-
livr.net/npm/bootstrap@5.3.0-al-
pha1/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        body {
            background-color: #f0f4f8;
            font-family: 'Arial', sans-
serif;
            color: #333;
        }
        .container {
            margin-top: 50px;
            border-radius: 8px;
            box-shadow: 0 2px 10px
rgba(0, 0, 0, 0.1);
            background-color: white;
            padding: 20px;
        }
        h1 {
            font-size: 3rem;
            margin-bottom: 20px;
            background-color: #e71628;
            text-align: center;
```

```
            color: #ffffff;
        }
        h1 img {
            width: 40px;
            vertical-align: middle;
            margin-right: 10px;
        }
        .btn-light-red {
            background-color: #f8d7da;
            color: #721c24;
            border: none;
            transition: background-color
 0.3s, color 0.3s;
        }
        .btn-light-red:hover {
            background-color: #f5c6cb;
            color: #721c24;
        }
        .btn-green {
            background-color: #28a745;
            color: white;
            border: none;
            transition: background-color
 0.3s, transform 0.3s;
        }
```

```
        .btn-green:hover {
            background-color: #218838;
            transform: scale(1.05);
        }
        table {
            width: 100%;
            margin-top: 20px;
        }
        th, td {
            text-align: center;
            padding: 15px;
        }
        th {
            background-color: #007bff;
            color: rgb(0, 0, 0);
        }
        .alert {
            text-align: center;
            margin-top: 20px;
        }
        input[type="text"], in-
  put[type="number"] {
            margin-bottom: 10px;
            width: calc(100% - 22px);
            padding: 10px;
```

```
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        form {
            display: flex;
            flex-direction: column;
            align-items: center;
            margin-bottom: 20px;
        }
        .actions {
            display: flex;
            gap: 5px;
            justify-content: center;
        }
        .action-buttons {
            width: 120px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1><img src="https://cdn-icons-
png.flaticon.com/512/2436/2436702.png"
alt="Book Icon"> Book Management</h1>
```

```
        {% with messages =
get_flashed_messages(with_catego-
ries=true) %}
            {% if messages %}
                {% for category, message
in messages %}
                    <div class="alert
alert-{{ category }} alert-dismissible
fade show" role="alert">
                        {{ message }}
                        <button
type="button" class="btn-close" data-
dismiss="alert" aria-la-
bel="Close"></button>
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}

        <form action="{{
url_for('add_book') }}" method="POST">
            <input type="text" name="ti-
tle" placeholder="Title" required>
            <input type="text" name="au-
thor" placeholder="Author" required>
```

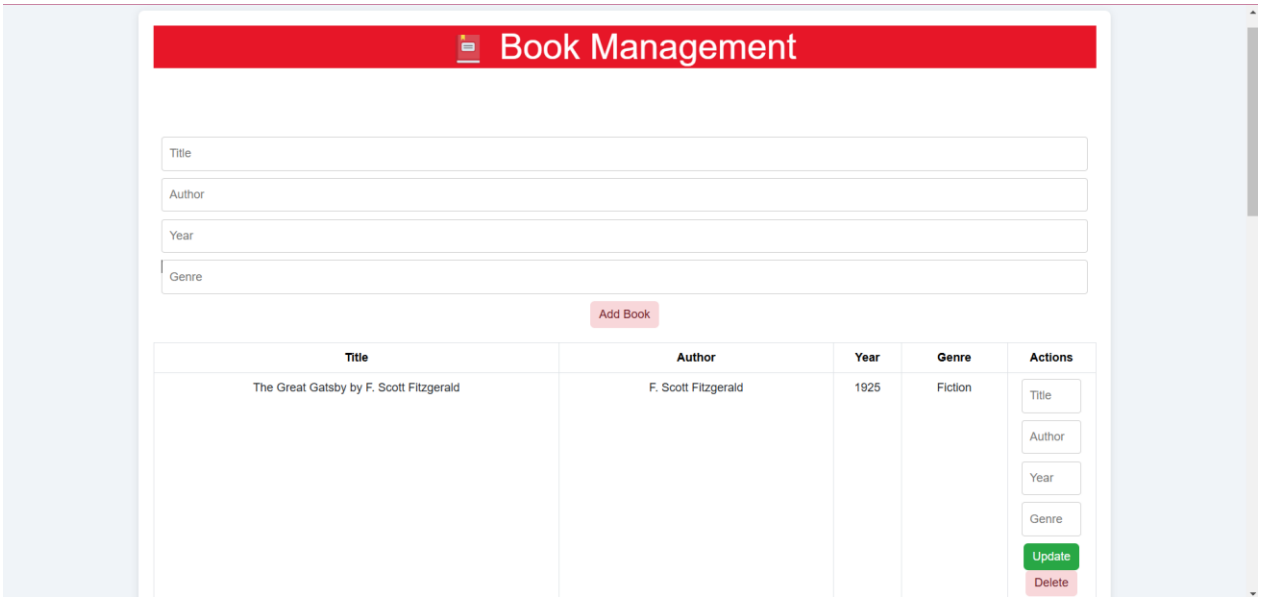
```
            <input type="number"
name="year" placeholder="Year" required>
            <input type="text"
name="genre" placeholder="Genre" re-
quired>
            <button type="submit"
class="btn btn-light-red">Add Book</but-
ton>
        </form>

        <table class="table table-bor-
dered">
            <thead>
                <tr>
                    <th>Title</th>
                    <th>Author</th>
                    <th>Year</th>
                    <th>Genre</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                {% for book in books %}
                    <tr>
```

```
                        <td>{{ book[1] }}</td>
                        <td>{{ book[2] }}</td>
                        <td>{{ book[3] }}</td>
                        <td>{{ book[4] }}</td>
                        <td class="action-buttons">
                            <form action="{{ url_for('update_book', book_id=book[0]) }}" method="POST" class="d-inline">
                                <input type="text" name="title" placeholder="Title" required>
                                <input type="text" name="author" placeholder="Author" required>
                                <input type="number" name="year" placeholder="Year" required>
```

```
                                    <input
type="text" name="genre" place-
holder="Genre" required>
                                    <button
type="submit" class="btn btn-green">Up-
date</button>
                                </form>
                                <a href="{{
url_for('delete_book', book_id=book[0])
}}" class="btn btn-light-red">Delete</a>
                            </td>
                        </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
        <script src="https://cdn.jsde-
livr.net/npm/bootstrap@5.3.0-al-
pha1/dist/js/bootstrap.bun-
dle.min.js"></script>
        <script>
            setTimeout(() => {
                const alerts = docu-
ment.querySelectorAll('.alert');
                alerts.forEach(alert => {
```

```
                alert.classList.re-
move('show');
                alert.classList.add('fad
e');
            });
        }, 3000);
    </script>
</body>
</html>
```

Trang Web



Trong Cơ sơ Database

Query | Query History

```
1  SELECT * FROM Books;
```

Data Output | Messages | Notifications

	id [PK] integer	title character varying (255)	author character varying (255)	year integer	genre character varying (100)
1	24	The Great Gatsby by F. Scott Fitzgerald	F. Scott Fitzgerald	1925	Fiction
2	25	Ulysses	by James Joyce	1922	Fiction
3	27	The Catcher in the Rye by J. D. Salinger	by J. D. Salinger	1951	Fiction
4	26	In Search of Lost Time by Marcel Proust	by Marcel Proust	1913	Fiction
5	28	Nineteen Eighty Four by George Orwell	by George Orwell	1949	Fiction
6	29	Harry Potter and the Order of the Phoenix	by J K Rowling (Hardcover)	2021	Synopsis

4. ĐƯA LÊN GITHUB

Python_NangCao/Project3 at main · Lattakone1/Python_NangCao