

BÀI BÁO CÁO

TÊN BÀI: XÂY DỰNG WEB QUẢN LÝ SINH VIÊN KẾT NỐI VỚI DATABASE TRÊN POSTGRES

MSSV: 2274802010083

HỌ TÊN: ĐOÀN TRUNG CƯỜNG

Nội dung:

1. Giao diện:

1.1 Mô tả giao diện: Giao diện web quản lý sinh viên được thiết kế đơn giản, trực quan, và thân thiện với người dùng, bao gồm các thành phần chính:

Tiêu đề chính: "Student Management" hiển thị ở đầu trang.

Form thêm sinh viên:

Một form nhập liệu gồm:

Trường Full Name (Họ tên sinh viên).

Trường Department ID (Mã ngành học).

Nút bấm Add Student để thêm sinh viên mới.

Danh sách sinh viên:

Bảng hiển thị danh sách sinh viên gồm các cột:

- **ID:** Mã định danh sinh viên (tự động tăng).
- **Full Name:** Tên đầy đủ của sinh viên.
- **Department ID:** Mã ngành học của sinh viên.
- **Actions:** Nút Delete để xóa sinh viên khỏi cơ sở dữ liệu.

1.2 Hình ảnh giao diện

Form nhập liệu và bảng danh sách sinh viên.

Form Thêm Sinh Viên.:

- Trường Full Name để nhập tên sinh viên.
- Trường Department ID để nhập mã ngành học.
- Nút Add Student để thêm dữ liệu.

Bảng hiển thị danh sách sinh viên.

- Hiển thị thông tin sinh viên theo dạng bảng với các cột như ID, Tên đầy đủ, Mã ngành học, và hành động.

Là 1 giao diện lọc các dữ liệu data trong PostgreSQL.

Student Management

Full Name

Department ID

Add Student

Student List

ID	Full Name	Department ID	Actions
1	Nguyen Van A	1	Delete
2	Tran Thi B	2	Delete
3	Le Van C	3	Delete
4	Doan Trung Cuong	2258	Delete
5	Đoàn Trung Nam	376641580	Delete

2. Chức năng:

2.1 Danh sách các chức năng

Kết nối với cơ sở dữ liệu PostgreSQL:

- Sử dụng thư viện psycopg2 để kết nối với cơ sở dữ liệu PostgreSQL.
- Đảm bảo an toàn và hiệu quả khi truy vấn dữ liệu.

Thêm sinh viên: Cho phép thêm một sinh viên mới với thông tin

- Họ tên (Full Name).
- Mã ngành học (Department ID)

Xóa sinh viên: Xóa một sinh viên khỏi cơ sở dữ liệu dựa trên studentid.

Hiển thị danh sách sinh viên:

- Hiển thị toàn bộ danh sách sinh viên từ cơ sở dữ liệu PostgreSQL.
- Cập nhật danh sách ngay sau khi thêm hoặc xóa dữ liệu.

2.2 Quy trình hoạt động

Người dùng truy cập trang chủ.

Giao diện hiển thị danh sách sinh viên hiện có trong cơ sở dữ liệu.

Người dùng có thể:

- Nhập thông tin sinh viên mới và nhấn nút Add Student để thêm sinh viên.
- Nhấn nút Delete để xóa sinh viên khỏi danh sách.

Sau mỗi thao tác, giao diện sẽ cập nhật dữ liệu mới nhất từ cơ sở dữ liệu.

3. Mã chương trình:

3.1 Mã nguồn Flask (Python)

```
4. from flask import Flask, render_template, request, redirect, flash
5. import psycopg2
6. from psycopg2 import sql
7.
8. app = Flask(__name__)
9. app.secret_key = "secret_key" # Để sử dụng flash thông báo
10.
11. # Kết nối cơ sở dữ liệu
12. def get_db_connection():
13.     try:
14.         conn = psycopg2.connect(
15.             dbname='qlsv',
16.             user='postgres',
17.             password='123456',
18.             host='localhost',
19.             port='5432'
20.         )
21.         return conn
22.     except Exception as e:
```

```
23.         print(f"Error connecting to database: {e}")
24.         return None
25.
26. # Trang chủ - hiển thị danh sách sinh viên
27. @app.route("/")
28. def index():
29.     conn = get_db_connection()
30.     if not conn:
31.         flash("Cannot connect to the database", "danger")
32.         return render_template("index.html", students=[], error=True)
33.
34.     try:
35.         cur = conn.cursor()
36.         cur.execute("SELECT * FROM students")
37.         students = cur.fetchall()
38.         cur.close()
39.         conn.close()
40.         return render_template("index.html", students=students)
41.     except Exception as e:
42.         flash(f"Error loading data: {e}", "danger")
43.         return render_template("index.html", students=[], error=True)
44.
45. # Thêm sinh viên
46. @app.route("/add", methods=["POST"])
47. def add_student():
48.     fullname = request.form.get("fullname")
49.     departmentid = request.form.get("departmentid")
50.
51.     if not fullname or not departmentid:
52.         flash("All fields are required", "danger")
53.         return redirect("/")
54.
55.     conn = get_db_connection()
56.     if not conn:
57.         flash("Cannot connect to the database", "danger")
58.         return redirect("/")
59.
60.     try:
61.         cur = conn.cursor()
62.         cur.execute("INSERT INTO students (fullname, departmentid) VALUES
(%s, %s)", (fullname, departmentid))
63.         conn.commit()
64.         cur.close()
65.         conn.close()
66.         flash("Student added successfully", "success")
```

```

67.     except Exception as e:
68.         flash(f"Error adding student: {e}", "danger")
69.         return redirect("/")
70.
71. # Xóa sinh viên
72. @app.route("/delete/<int:studentid>")
73. def delete_student(studentid):
74.     conn = get_db_connection()
75.     if not conn:
76.         flash("Cannot connect to the database", "danger")
77.         return redirect("/")
78.
79.     try:
80.         cur = conn.cursor()
81.         cur.execute("DELETE FROM students WHERE studentid = %s",
            (studentid,))
82.         conn.commit()
83.         cur.close()
84.         conn.close()
85.         flash("Student deleted successfully", "success")
86.     except Exception as e:
87.         flash(f"Error deleting student: {e}", "danger")
88.         return redirect("/")
89.
90. if __name__ == "__main__":
91.     app.run(debug=True)
92.

```

3.2 File HTML (index.html)

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Management</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
</head>
<body>
    <div class="container mt-5">
        <h1 class="text-center">Student Management</h1>

        <!-- Thông báo -->
        {% with messages = get_flashed_messages(with_categories=true) %}

```

```

{% if messages %}
<div class="mt-3">
    {% for category, message in messages %}
    <div class="alert alert-{{ category }}">{{ message }}</div>
    {% endfor %}
</div>
{% endif %}
{% endwith %}

<!-- Form thêm sinh viên -->
<form method="POST" action="/add" class="mt-4">
    <div class="mb-3">
        <label for="fullname" class="form-label">Full Name</label>
        <input type="text" class="form-control" id="fullname"
name="fullname" required>
    </div>
    <div class="mb-3">
        <label for="departmentid" class="form-label">Department
ID</label>
        <input type="number" class="form-control" id="departmentid"
name="departmentid" required>
    </div>
    <button type="submit" class="btn btn-success">Add Student</button>
</form>

<!-- Danh sách sinh viên -->
<h2 class="mt-5">Student List</h2>
<table class="table table-bordered mt-3">
    <thead>
        <tr>
            <th>ID</th>
            <th>Full Name</th>
            <th>Department ID</th>
            <th>Actions</th>
        </tr>
    </thead>
    <tbody>
        {% for student in students %}
        <tr>
            <td>{{ student[0] }}</td>
            <td>{{ student[1] }}</td>
            <td>{{ student[2] }}</td>
            <td>
                <a href="/delete/{{ student[0] }}" class="btn btn-danger
btn-sm">Delete</a>
            </td>
        </tr>
        {% endfor %}
    </tbody>
</table>

```

```
        </td>
    </tr>
    {% endfor %}
</tbody>
</table>
</div>
</body>
</html>
```

Link GitHub: https://github.com/Cuong2k42004/BaiCuoiKy_PythonNangCao/tree/main