

TRƯỜNG ĐẠI HỌC HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC

LẬP TRÌNH PYTHON NÂNG CAO

NGÀNH: CÔNG NGHỆ THÔNG TIN

Đồ án:

QUẢN LÝ SINH VIÊN BẰNG FLASK

SVTH: Nguyễn Thành Nhân

MSSV: 2274802010600

GVHD: Huỳnh Thái Học

Tp. Hồ Chí Minh – năm 2024

LỜI CẢM ƠN

Viết một báo cáo đồ án môn học là một trong những việc khó nhất mà chúng em phải hoàn thành trong quá trình học một môn học. Trong quá trình thực hiện đề tài chúng em đã gặp rất nhiều khó khăn và ngỡ ngàng. Nếu không có những sự giúp đỡ và lời động viên chân thành của nhiều người có lẽ chúng em khó có thể hoàn thành tốt tiểu luận này. Đầu tiên chúng em xin gửi lời biết ơn chân thành đến thầy Huỳnh Thái Học, người trực tiếp hướng dẫn chúng em hoàn thành tiểu luận này. Những ý kiến đóng góp của thầy là vô cùng hữu ích, nó giúp chúng em nhận ra các khuyết điểm của đồ án. Cảm ơn thầy và các bạn trường Đại học Văn Lang là những người đã cùng nhóm em sát cánh và trải nghiệm để hoàn thành đồ án môn học.

Mục lục

1. Giới Thiệu.....	4
2. Mục Tiêu	4
3. Phân Tích Yêu Cầu	4
3.1 Yêu Cầu Chức Năng.....	4
3.2 Yêu Cầu Phi Chức Năng	4
4. Thiết Kế Hệ Thống.....	4
4.1 Công Nghệ Sử Dụng	4
4.2 Thiết Kế Giao Diện.....	5
5. Cấu Trúc Mã Nguồn	6
5.1 Cấu Trúc Thư Mục	6
5.2 Các Lớp và Hàm Chính	7
6. Chi Tiết Mã Nguồn.....	7
6.1 Định nghĩa mô hình dữ liệu.....	7
6.2 Xử lý thêm sinh viên.....	8
6.3 Giao diện HTML - Danh sách sinh viên	8
7. Kết Quả Đạt Được.....	9
8. Kết Luận.....	10
9. Hướng Phát Triển Trong Tương Lai.....	10
10. Tài Liệu Tham Khảo.....	10
11. Github.....	10

1. Giới Thiệu

Bài toán đặt ra là xây dựng một hệ thống quản lý sinh viên đơn giản, sử dụng Python với Flask để tạo giao diện web và PostgreSQL làm cơ sở dữ liệu. Hệ thống cho phép quản lý thông tin sinh viên như mã số sinh viên (MSSV), họ tên, ngày sinh, và lớp học. Dữ liệu được lưu trữ trong PostgreSQL để đảm bảo tính bền vững và an toàn.

2. Mục Tiêu

Tạo ra một ứng dụng giúp quản lý thông tin sinh viên dễ dàng và tiện lợi qua giao diện web.

Sử dụng Flask để xây dựng ứng dụng web đơn giản nhưng mạnh mẽ.

Tích hợp PostgreSQL làm hệ quản trị cơ sở dữ liệu, đảm bảo ổn định và bảo mật dữ liệu.

Áp dụng Flask-SQLAlchemy để quản lý ORM (Object Relational Mapping), đơn giản hóa các thao tác với cơ sở dữ liệu.

3. Phân Tích Yêu Cầu

3.1 Yêu Cầu Chức Năng

Đăng nhập: Người dùng phải đăng nhập để truy cập hệ thống.

Xem danh sách sinh viên: Hiển thị danh sách sinh viên dưới dạng bảng.

Thêm sinh viên: Người dùng có thể thêm thông tin sinh viên mới.

Chỉnh sửa thông tin sinh viên: Cập nhật thông tin của sinh viên trong hệ thống.

Xóa sinh viên: Xóa thông tin sinh viên khi không còn cần thiết.

3.2 Yêu Cầu Phi Chức Năng

Bảo mật: Hệ thống yêu cầu đăng nhập và bảo vệ thông tin người dùng.

Trực quan: Giao diện web dễ sử dụng và thân thiện với người dùng.

Hiệu quả: Hệ thống phải phản hồi nhanh và xử lý dữ liệu hiệu quả.

4. Thiết Kế Hệ Thống

4.1 Công Nghệ Sử Dụng

Python Flask: Framework chính để xây dựng ứng dụng web.

Flask-SQLAlchemy: Thư viện hỗ trợ ORM để tương tác với PostgreSQL.

PostgreSQL: Hệ quản trị cơ sở dữ liệu quan hệ.

Werkzeug Security: Hỗ trợ mã hóa mật khẩu và xác thực thông tin người dùng.

4.2 Thiết Kế Giao Diện

Màn hình đăng nhập: Người dùng nhập tài khoản và mật khẩu.

Danh sách sinh viên: Hiển thị danh sách sinh viên với các nút chỉnh sửa và xóa.

Hệ Thống Quản Lý Sinh Viên						
Danh Sách Sinh Viên						
<button>Thêm Sinh Viên</button>						
STT	MSSV	Họ và Tên	Ngày Sinh	Lớp	Hành động	
1	2274802010600	Nguyễn Thành Nhân	2004-06-12	CNTT02	<button>Sửa</button>	<button>Xóa</button>
2	adacc	adasd	2004-07-25	qurdoq	<button>Sửa</button>	<button>Xóa</button>
3	qudfq3.KGFEH	LKJEBSFCL s	2024-11-09	DQÚADCÁV ÁEV	<button>Sửa</button>	<button>Xóa</button>
4	qudqurd	123	2024-11-03	dZfdqqr	<button>Sửa</button>	<button>Xóa</button>

Thêm sinh viên: Biểu mẫu nhập liệu để thêm thông tin sinh viên.

Hệ Thống Quản Lý Sinh Viên	
Thêm Sinh Viên	
MSSV:	<input type="text" value="2274802010357"/>
Họ và Tên:	<input type="text" value="Trần Văn C"/>
Ngày Sinh:	<input type="text" value="15/11/2024"/>
Lớp:	<input type="text" value="CNTT12"/>
<button>Thêm Sinh Viên</button>	

Chỉnh sửa sinh viên: Biểu mẫu cập nhật thông tin sinh viên.

Hệ Thống Quản Lý Sinh Viên

MSSV:

Họ và Tên:

Ngày Sinh:

Lớp:

Cập Nhật

5. Cấu Trúc Mã Nguồn

5.1 Cấu Trúc Thư Mục

static/: Chứa tệp CSS (style.css) để định dạng giao diện.

templates/: Chứa các tệp HTML cho giao diện ứng dụng.

login.html

index.html

add_student.html

edit_student.html

app.py: Tập chính điều phối ứng dụng.

requirements.txt: Danh sách các thư viện cần thiết.

5.2 Các Lớp và Hàm Chính

Lớp Student: Đại diện cho bảng sinh viên trong cơ sở dữ liệu.

Thuộc tính: mssv, hoten, ngaysinh, lop.

Lớp User: Quản lý tài khoản đăng nhập.

Phương thức: set_password(), check_password().

Các route chính:

/login: Xử lý đăng nhập.

/: Trang danh sách sinh viên (chỉ truy cập sau khi đăng nhập).

/add_student: Thêm sinh viên mới.

/edit_student/<mssv>: Chỉnh sửa thông tin sinh viên.

/delete_student/<mssv>: Xóa sinh viên.

6. Chi Tiết Mã Nguồn

Dưới đây là một số đoạn mã quan trọng:

6.1 Định nghĩa mô hình dữ liệu

python

Sao chép mã

```
class Student(db.Model):
```

```
    __tablename__ = 'sinhvien'
```

```
    mssv = db.Column(db.String(13), primary_key=True)
```

```
    hoten = db.Column(db.String(200), nullable=False)
```

```
    ngaysinh = db.Column(db.Date, nullable=False)
```

```
    lop = db.Column(db.String(50), nullable=False)
```

6.2 Xử lý thêm sinh viên

python

Sao chép mã

```
@app.route('/add_student', methods=['GET', 'POST'])
def add_student():
    if 'username' not in session:
        return redirect(url_for('login'))
    if request.method == 'POST':
        mssv = request.form['mssv']
        hoten = request.form['hoten']
        ngaysinh = request.form['ngaysinh']
        lop = request.form['lop']
        student = Student(mssv=mssv, hoten=hoten, ngaysinh=ngaysinh, lop=lop)
        db.session.add(student)
        db.session.commit()
        flash("Thêm sinh viên thành công!", "success")
        return redirect(url_for('index'))
    return render_template('add_student.html')
```

6.3 Giao diện HTML - Danh sách sinh viên

html

Sao chép mã

```
<table>
  <thead>
    <tr>
      <th>MSSV</th>
      <th>Họ và Tên</th>
      <th>Ngày sinh</th>
      <th>Lớp</th>
```



```

        <th>Hành động</th>

    </tr>

</thead>

<tbody>

    {% for student in students %}

    <tr>

        <td>{{ student.mssv }}</td>

        <td>{{ student.hoten }}</td>

        <td>{{ student.ngaysinh }}</td>

        <td>{{ student.lop }}</td>

        <td>

            <a href="{{ url_for('edit_student', mssv=student.mssv) }}">Chỉnh sửa</a>

            <form action="{{ url_for('delete_student', mssv=student.mssv) }}"
method="POST">

                <button type="submit">Xóa</button>

            </form>

        </td>

    </tr>

    {% endfor %}

</tbody>

</table>

```

7. Kết Quả Đạt Được

Giao diện: Xây dựng giao diện web đơn giản nhưng đầy đủ chức năng.

Chức năng: Đáp ứng đầy đủ các yêu cầu: đăng nhập, thêm, chỉnh sửa, và xóa sinh viên.

Tích hợp cơ sở dữ liệu: Dữ liệu được lưu trữ và quản lý hiệu quả trong PostgreSQL.

8. Kết Luận

Hệ thống quản lý sinh viên đã được xây dựng thành công với Flask và PostgreSQL. Hệ thống không chỉ đáp ứng nhu cầu quản lý cơ bản mà còn dễ dàng mở rộng thêm các chức năng.

9. Hướng Phát Triển Trong Tương Lai

Tìm kiếm sinh viên: Bổ sung chức năng tìm kiếm nhanh.

Phân quyền: Thêm các cấp độ người dùng và phân quyền truy cập.

Báo cáo: Tích hợp các báo cáo thống kê sinh viên theo lớp hoặc độ tuổi.

10. Tài Liệu Tham Khảo

Flask Documentation

SQLAlchemy Documentation

PostgreSQL Documentation

11. Github.

[NGUYENTHANHNHAN/baitap3 at main · NhanDet/NGUYENTHANHNHAN](#)