

TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CUỐI KỲ
MÔN HỌC LẬP TRÌNH PYTHON NÂNG CAO

Bài tập 03:

Website Quản lý sinh viên

SVTH: Trần Ngọc Phước – 2274802010699

LỚP: 241_71ITSE31003_02

GVHD: Huỳnh Thái Học

TP. Hồ Chí Minh – 11/2024

MỤC LỤC

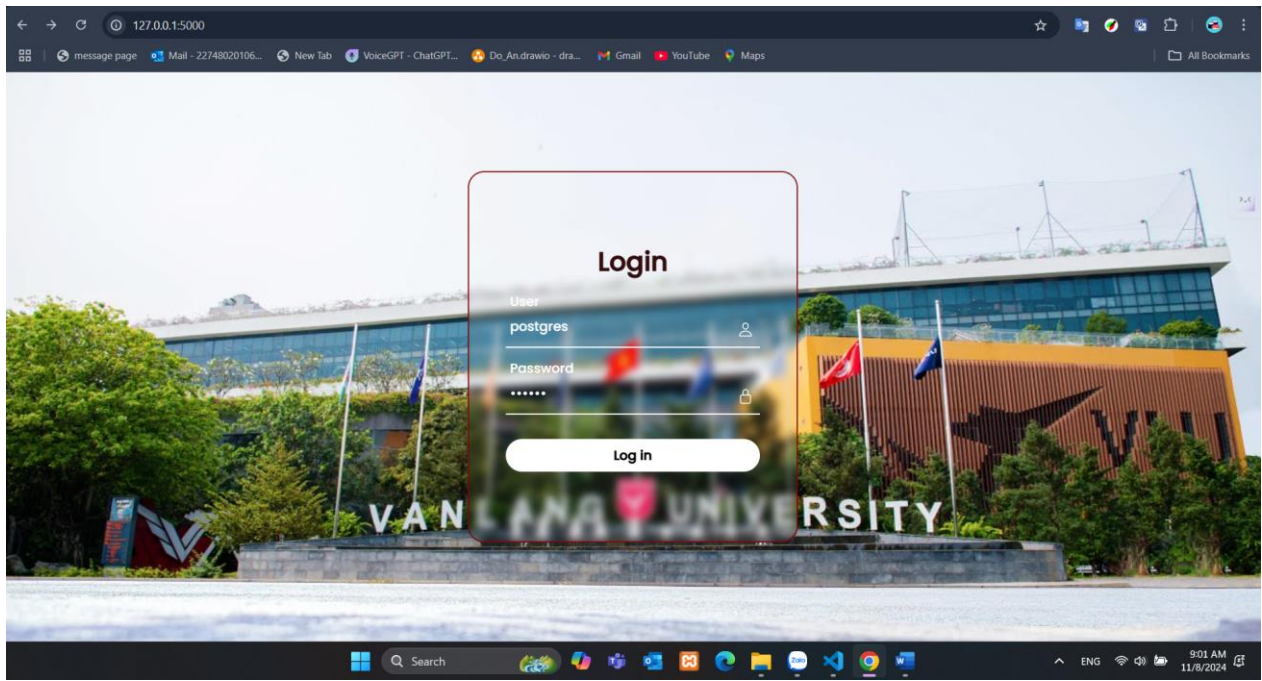
1. Giới thiệu về flask.....	3
2. Giao diện Website Quản lý sinh viên.....	4
3. Chức năng.....	6
3.1 Login vào cơ sở dữ liệu:	6
3.2 Lọc thông tin sinh viên dựa vào lớp học:	6
3.3 Thêm sinh viên mới vào cơ sở dữ liệu:.....	6
4. Mã chương trình.	7
5. Đường link github.....	11
--- HẾT---	11

1. Giới thiệu về flask.

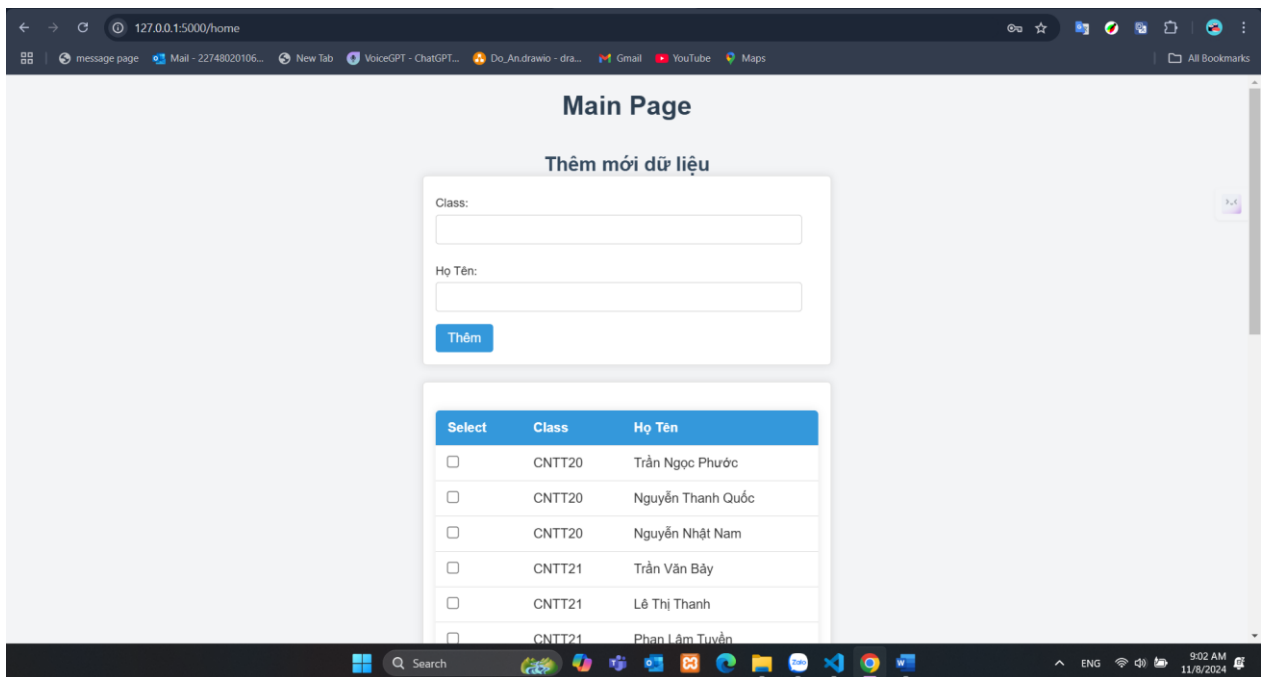
Flask là một microframework phổ biến trong Python, được thiết kế để phát triển các ứng dụng web nhanh chóng và dễ dàng. Ra mắt lần đầu vào năm 2010 bởi Armin Ronacher, Flask nổi bật với triết lý “đơn giản nhưng mở rộng”, mang lại cho các nhà phát triển sự linh hoạt trong việc lựa chọn các công cụ và thư viện tùy ý mà không bị ràng buộc vào cấu trúc cứng nhắc. Với thiết kế nhẹ và tối giản, Flask không đi kèm với các tính năng như ORM hay xác thực, giúp người dùng dễ dàng kiểm soát và mở rộng theo nhu cầu dự án. Flask cũng hỗ trợ hệ thống định tuyến, hệ thống mẫu Jinja2, và khả năng tích hợp dễ dàng với các tiện ích mở rộng, làm cho nó trở thành lựa chọn lý tưởng cho cả các dự án nhỏ và các ứng dụng web phức tạp.

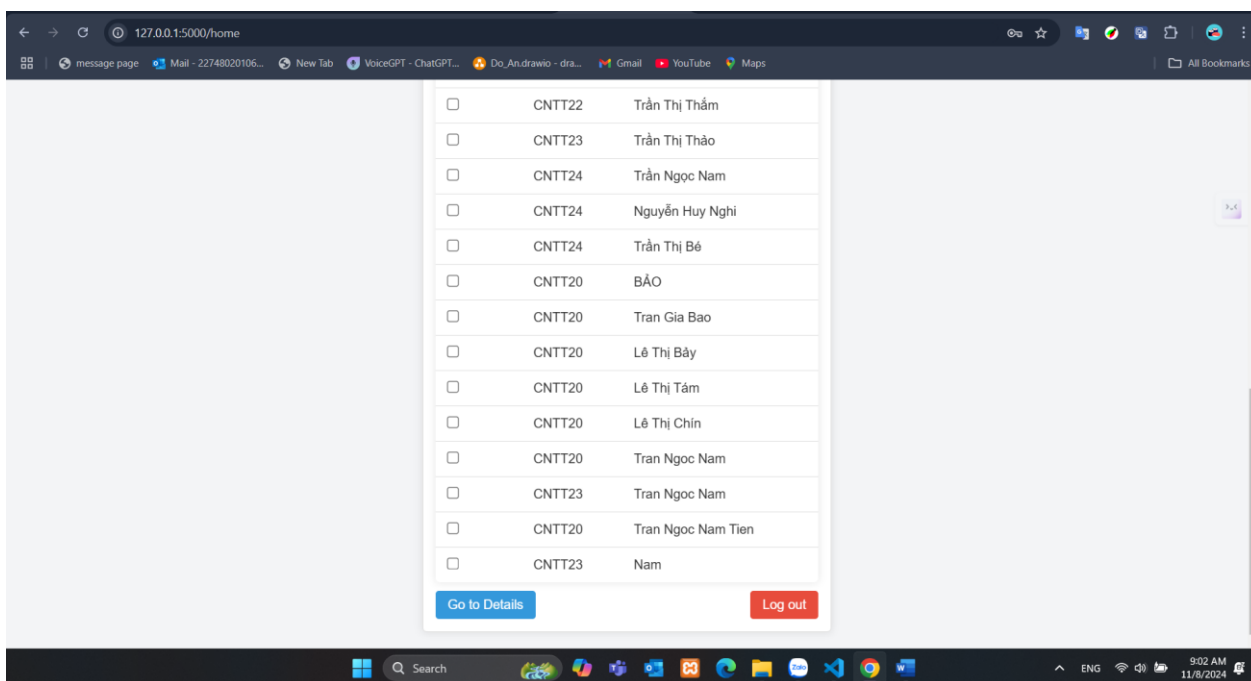
2. Giao diện Website Quản lý sinh viên.

Giao diện để login vào Website

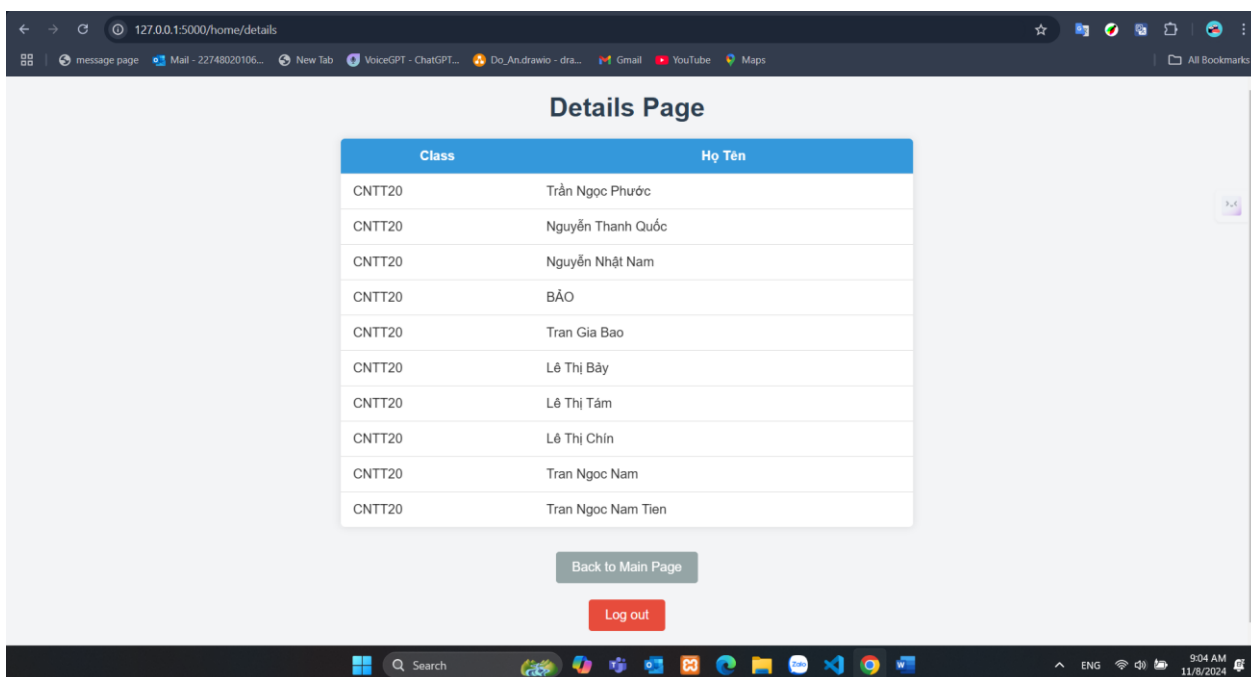


Giao diện khi login thành công vào website





Giao diện khi Lọc thông tin sinh viên Khi người dùng chọn vào mã lớp -> Go to Details



Và khi người dùng nhấn **“Log out”** thì sẽ quay lại trang login

3.Chức năng.

Website Quản lý sinh viên được xây dựng bằng Python và thư viện flask có các chức năng chính sau:

3.1 Login vào cơ sở dữ liệu:

- Người dùng nhập user: “postgres”
- Người dùng nhập Password: “123456”
- Phần này người dùng cung cấp dữ liệu đầu vào là user và Pass và nhấn nút “Log in”, website sẽ tải toàn bộ danh sách sinh viên có trong cơ sở dữ liệu và hiển thị vào trang “Main Page”

3.2 Lọc thông tin sinh viên dựa vào lớp học:

- Người dùng chọn lớp cần xem ở các ô “Select”
- Phần này người dùng đã chọn “mã lớp” và nhấn nút “Go to Details”, Website sẽ hiển thị toàn bộ danh sách sinh viên của lớp học đó vào trang “Details Page”.

3.3 Thêm sinh viên mới vào cơ sở dữ liệu:

- Người dùng nhập tên và lớp vào hai ô nhập liệu tương ứng.
- Phần này người dùng cung cấp dữ liệu đầu vào là “tên sinh viên mới” và “Mã lớp học” khi nhấn nút “Thêm”, ứng dụng sẽ thêm sinh viên mới đó vào mã lớp học tương ứng.

4. Mã chương trình.

Về giao diện:

- Trang Index.html để phục vụ cho việc Login

```
app.py 4, U index.html U x home.html U details.html U database.py U index.css U
Do_An_Cuoi_Ky > project_01 > test_03 > templates > index.html > html > body > section > div.form-box > div.form-value > form > p.messages
2 <html lang="en">
9 <body>
10 {% block content %}
11 <section>
12
13     <div class="form-box">
14         <div class="form-value">
15             <form action="#" method="post">
16                 <h2>Login</h2>
17                 {% with messages = get_flashed_messages() %}
18                     {% if messages %}
19                         {% for m in messages %}
20                             <p class="messages">{{ m|safe }}</p>
21                         {% endfor %}
22                     {% endif %}
23                 {% endwith %}
24                 <div class="inputbox">
25                     <ion-icon name="person-outline"></ion-icon>
26                     <input type="text" name="user" required> </input>
27                     <label for="">User</label>
28                 </div>
29                 <div class="inputbox">
30                     <ion-icon name="lock-closed-outline"></ion-icon>
31                     <input type="password" name="password" required> </input>
32                     <label for="">Password</label>
33                 </div>
34                 <button>Log in</button>
35             </form>
36         </div>
37     </div>
38 </section>
39 {% endblock %}
```

- Trang home.html để phục vụ cho việc hiển thị cơ sở dữ liệu và chức năng thêm sinh viên mới

```
app.py 4, U index.html U home.html U x details.html U database.py U index.css U
Do_An_Cuoi_Ky > project_01 > test_03 > templates > home.html > html > head > style > ahover
2 <html lang="en">
162 <body>
173 <h2>Thêm mới dữ liệu</h2>
174 <form action="{{ url_for('add data') }}" method="post">
175     <label for="class">Class:</label>
176     <input type="text" name="class" required>
177     <label for="name">Họ Tên:</label>
178     <input type="text" name="name" required>
179     <button type="submit">Thêm</button>
180 </form>
181
182 <!-- Bảng dữ liệu -->
183 <form action="{{ url_for('details') }}" method="post">
184     <table>
185         <tr>
186             <th>Select</th>
187             <th>Class</th>
188             <th>Họ Tên</th>
189         </tr>
190         {% for row in data %}
191             <tr>
192                 <td><input type="checkbox" name="selected" value="{{ row[0] }}"></td>
193                 <td>{{ row[0] }}</td>
194                 <td>{{ row[1] }}</td>
195             </tr>
196         {% endfor %}
197     </table>
198     <div class="form-buttons">
199         <button type="submit">Go to Details</button>
200         <a href="{{ url_for('login') }}">Log out</a>
201     </div>
202 </form>
203 </body>
```

- Trang details.html để phục vụ cho việc lọc các sinh viên theo mã lớp học mà người dùng chọn “Select”

```

2 <html lang="en">
4 <head>
98 </head>
99
100 <body>
101     {% with messages = get_flashed_messages() %}
102     {% if messages %}
103     {% for m in messages %}
104     <p class="message">{{ m }}</p>
105     {% endfor %}
106     {% endif %}
107     {% endwith %}
108
109     <h1>Details Page</h1>
110
111     <table>
112     <tr>
113         <th>Class</th>
114         <th>Họ Tên</th>
115     </tr>
116     {% for row in data %}
117     <tr>
118         <td>{{ row[0] }}</td>
119         <td>{{ row[1] }}</td>
120     </tr>
121     {% endfor %}
122     </table>
123
124     <a href="{{ url_for('home') }}" class="back-link">Back to Main Page</a>
125     <a href="{{ url_for('login') }}">Log out</a>
126 </body>
127

```

Về các chức năng chính:

- Kết nối vào database

```

1 # db_connection.py
2 import psycopg2
3
4 class Database:
5     def __init__(self, db_name, user, password, host, port):
6         self.db_name = db_name
7         self.user = user
8         self.password = password
9         self.host = host
10        self.port = port
11
12    def connect(self):
13        try:
14            self.conn = psycopg2.connect(
15                dbname=self.db_name,
16                user=self.user,
17                password=self.password,
18                host=self.host,
19                port=self.port
20            )
21            self.cur = self.conn.cursor()
22            return True
23        except Exception as e:
24            print(f"Error connecting to the database: {e}")
25            return False
26
27
28

```



```
app.py 4, U X index.html U home.html U details.html U database.py 1, U index.css U
Do_An_Cuoi_Ky > project_01 > test_03 > app.py > login
1 from flask import Flask, render_template, request, redirect, url_for, make_response, flash, session
2 import psycopg2
3 from psycopg2 import sql
4 from database import Database
5
6 app = Flask(__name__)
7 app.secret_key = 'your_secret_key' # Thêm khóa bảo mật cho session
8
9 # Route trang đăng nhập (mặc định)
10 @app.route('/', methods=['GET', 'POST'])
11 def login():
12     if request.method == 'POST':
13         # Lấy dữ liệu từ form
14         user = request.form['user']
15         password = request.form['password']
16
17         # Tạo kết nối với thông tin từ form
18         db = Database(db_name='dbtest', user=user, password=password, host='localhost', port=5432)
19         if db.connect():
20             # Lưu thông tin đăng nhập trong session
21             session['user'] = user
22             session['password'] = password
23             db.cur.close() # Đóng cursor
24             db.conn.close() # Đóng kết nối
25
26             # Kết nối thành công, chuyển đến trang home
27             return redirect(url_for('home'))
28         else:
29             # Kết nối thất bại, hiển thị thông báo lỗi
30             flash("Kết nối không thành công. <br> Vui lòng kiểm tra lại thông tin đăng nhập.", 'error')
31
32     return render_template('index.html')
33 # Trang chính khi kết nối thành công
```

• Load Data

```
33 # Trang chính khi kết nối thành công
34 @app.route('/home', methods=['GET', 'POST'])
35 def home():
36     # Kiểm tra session trước khi truy cập trang home
37     if 'user' in session and 'password' in session:
38         db = Database(db_name='dbtest', user=session['user'], password=session['password'], host='localhost', port=5432)
39         if db.connect():
40             db.cur.execute('SELECT * FROM khoacntt')
41             data = db.cur.fetchall()
42             db.cur.close()
43             db.conn.close()
44             return render_template('home.html', data=data)
45
46     # Nếu không có thông tin đăng nhập, chuyển về trang đăng nhập
47     flash("Bạn cần đăng nhập để truy cập trang này.", 'error')
48     return render_template('index.html')
49
```

• Insert data

```
50 # Route để thêm dữ liệu mới vào database
51 @app.route('/home/add', methods=['POST'])
52 def add_data():
53     # Kiểm tra session trước khi cho phép thêm dữ liệu
54     if 'user' in session and 'password' in session:
55         # Lấy dữ liệu từ form
56         new_class = request.form['class']
57         new_name = request.form['name']
58
59         db = Database(db_name='dbtest', user=session['user'], password=session['password'], host='localhost', port=5432)
60         if db.connect():
61             try:
62                 # Thêm dữ liệu vào bảng khoacntt
63                 insert_query = "INSERT INTO khoacntt (class, hoten) VALUES (%s, %s)"
64                 db.cur.execute(insert_query, (new_class, new_name))
65                 db.conn.commit()
66                 flash("Thêm dữ liệu thành công.", 'success')
67             except Exception as e:
68                 flash(f"Lỗi khi thêm dữ liệu: {e}", 'error')
69             finally:
70                 db.cur.close()
71                 db.conn.close()
72
73         return redirect(url_for('home'))
74
```

• Select data

```
75 # Xem chi tiết với dữ liệu từ cookie
76 @app.route('/home/details', methods=['GET', 'POST'])
77 def details():
78     if request.method == 'POST':
79         selected_data = request.form.getlist('selected')
80         resp = make_response(redirect(url_for('details')))
81         # Cookies: Lưu class đã chọn từ trang chính
82         resp.set_cookie('selected_data', ','.join(selected_data))
83         return resp
84
85     # Lấy dữ liệu từ cookie
86     selected_data = request.cookies.get('selected_data', '').split(',')
87     if selected_data == ['']:
88         selected_data = []
89
90     # Kết nối cơ sở dữ liệu với thông tin từ session
91     db = Database(db_name='dbtest', user=session['user'], password=session['password'], host='localhost', port=5432)
92     if db.connect():
93         if selected_data:
94             query = sql.SQL("SELECT * FROM khoacntt WHERE class IN ({})").format(
95                 sql.SQL(', ').join(map(sql.Literal, selected_data))
96             )
97             db.cur.execute(query)
98             data = db.cur.fetchall()
99         else:
100             data = []
101             db.cur.close()
102             db.conn.close()
103     else:
104         flash("Kết nối cơ sở dữ liệu không thành công.", 'error')
105         return redirect(url_for('login'))
106     return render_template('details.html', data=data)
```

5. Đường link github.

<https://github.com/phuocsasc/PythonNC/tree/master/Baitap01>

--- HẾT ---