

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

Tên Đồ án:

HỆ THỐNG QUẢN LÝ TRƯỜNG HỌC

SVTH: Võ Minh Tân

MSSV: 2274802010799

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 bờ ngỗ. 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 NV A, 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.

Nhóm thực hiện báo cáo

MỤC LỤC

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	5
1.1 Lý do thực hiện dự án	5
1.2 Mục tiêu của dự án	5
1.3 Phạm vi và đối tượng sử dụng hệ thống	5
Chương 2: Tổng quan về công nghệ sử dụng	6
2.1 Ngôn ngữ lập trình và framework	6
2.2 Cơ sở dữ liệu	6
2.3 Các công nghệ khác	6
Chương 3: Thiết kế và triển khai hệ thống	6
3.1 Cấu trúc hệ thống	6
3.1.1 Module Quản lý giáo viên:	6
3.1.2 Module Quản lý môn học:	7
3.1.3 Module Quản lý lớp học:	8
3.1.4 Module Quản lý học sinh:	9
3.1.5 Module Đăng nhập và phân quyền:	10
3.2 Quy trình triển khai	11
3.2.1 Giai đoạn 1: Xây dựng cơ sở dữ liệu	11
3.2.2 Giai đoạn 2: Phát triển Backend	12
3.2.3 Giai đoạn 3: Phát triển Frontend	12
3.2.4 Giai đoạn 4: Kiểm thử và triển khai	12
3.3 Giao diện	12
3.3.1 Trang login	12
3.3.2 Trang register	13
3.3.3 Trang dashboard	13
3.3.4 Mục Sinh Viên	14
3.3.5. Mục lớp	14
3.3.6 Mục học phần	14

3.3.5. Mục giảng viên	15
3.3.6 Mục môn học	15
Chương 4: Kết luận	15

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Lý do thực hiện dự án

Trong môi trường giáo dục hiện đại, việc quản lý thông tin học sinh, giáo viên, môn học, lớp học và các hoạt động liên quan đóng một vai trò quan trọng trong việc nâng cao hiệu quả giảng dạy và học tập. Các hệ thống quản lý truyền thống thường gặp phải các vấn đề về tính năng, độ ổn định và khả năng mở rộng. Chính vì vậy, việc xây dựng một hệ thống quản lý giáo dục trên nền tảng web có thể giúp các trường học, cơ sở giáo dục, các tổ chức đào tạo dễ dàng theo dõi và quản lý học sinh, giáo viên, các môn học, lớp học, cũng như các hoạt động khác.

1.2 Mục tiêu của dự án

Mục tiêu chính của dự án là xây dựng một hệ thống quản lý giáo dục với các chức năng cơ bản bao gồm:

- Quản lý thông tin học sinh: Thêm mới, sửa đổi, xóa và hiển thị thông tin học sinh.
- Quản lý thông tin giáo viên: Quản lý thông tin cá nhân của giáo viên, gán môn học cho giáo viên.
- Quản lý môn học và lớp học: Thêm mới, sửa đổi và xóa các môn học, các lớp học.
- Quản lý phân công giảng dạy: Phân công giáo viên cho các lớp học, theo dõi các lớp học của mỗi giáo viên.
- Hệ thống đăng nhập và phân quyền người dùng: Người dùng có thể đăng nhập và quản lý hệ thống dựa trên quyền hạn của họ (ví dụ: quản trị viên, giáo viên, học sinh).

1.3 Phạm vi và đối tượng sử dụng hệ thống

Hệ thống này sẽ được sử dụng chủ yếu trong các trường học, các tổ chức giáo dục lớn, nơi có nhu cầu quản lý nhiều lớp học, môn học và học sinh. Các đối tượng sử dụng hệ thống bao gồm:

- **Quản trị viên:** Quản lý toàn bộ hệ thống, bao gồm việc thêm mới, chỉnh sửa và xóa các thông tin giáo viên, học sinh, lớp học và môn học.
- **Giáo viên:** Có thể theo dõi lớp học, sinh viên của mình và thực hiện các tác vụ giảng dạy.
- **Học sinh:** Xem thông tin về lớp học, giáo viên, môn học mà mình đang tham gia.

Chương 2: Tổng quan về công nghệ sử dụng

2.1 Ngôn ngữ lập trình và framework

Hệ thống được xây dựng dựa trên nền tảng Python với Flask là framework web. Flask là một framework nhẹ và dễ sử dụng, đặc biệt phù hợp với các dự án nhỏ đến vừa và giúp tiết kiệm thời gian phát triển.

- **Python:** Python là ngôn ngữ lập trình mạnh mẽ, dễ học và có một cộng đồng phát triển rộng lớn. Python cung cấp nhiều thư viện hỗ trợ cho việc phát triển web, khoa học dữ liệu và nhiều lĩnh vực khác.
- **Flask:** Flask là một micro-framework cho Python, giúp việc xây dựng ứng dụng web nhanh chóng mà không cần phải cấu hình phức tạp. Flask cho phép người phát triển dễ dàng tùy chỉnh và mở rộng ứng dụng của mình.

2.2 Cơ sở dữ liệu

Để lưu trữ dữ liệu của học sinh, giáo viên, môn học và lớp học, hệ thống sử dụng MySQL làm cơ sở dữ liệu. MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến, có khả năng xử lý lượng dữ liệu lớn và thực thi các truy vấn SQL nhanh chóng.

2.3 Các công nghệ khác

- **HTML, CSS, JavaScript:** Được sử dụng để xây dựng giao diện người dùng. Các giao diện người dùng được thiết kế thân thiện và dễ sử dụng.
- **Bootstrap:** Một framework CSS giúp phát triển giao diện web nhanh chóng và responsive, dễ dàng hiển thị trên nhiều loại thiết bị khác nhau.

Chương 3: Thiết kế và triển khai hệ thống

3.1 Cấu trúc hệ thống

Hệ thống được chia thành nhiều module như sau:

3.1.1 Module Quản lý giáo viên:

- Cho phép thêm mới, sửa đổi và xóa thông tin giáo viên.
- Phân công môn học cho giáo viên.

```

1  @app.route("/teacher", methods=['GET', 'POST'])
2  def teacher():
3      if 'loggedin' in session:
4          teachers = db.get_all_teachers()
5          subjects = db.get_all_subjects()
6          return render_template("teacher.html", teachers=teachers, subjects=subjects)
7      return redirect(url_for('login'))
8
9  @app.route("/edit_teacher", methods=['GET'])
10 def edit_teacher():
11     if 'loggedin' in session:
12         teacher_id = request.args.get('teacher_id')
13
14         teacher = db.get_teacher_by_id(teacher_id)
15
16         if teacher:
17             teacher_info = teacher[0]
18
19             subjects = db.get_all_subjects()
20
21             return render_template("edit_teacher.html", teacher=teacher_info, subjects=subjects)
22         else:
23             return "Teacher not found", 404
24
25     return redirect(url_for('login'))
26
27
28 @app.route("/save_teacher", methods=['POST'])
29 def save_teacher():
30     if 'loggedin' in session:
31         if request.method == 'POST' and 'teacher_name' in request.form and 'specialization' in request.form:
32             teacher_name = request.form['teacher_name']
33             specialization = request.form['specialization']
34             action = request.form['action']
35
36             if action == 'updateTeacher':
37                 teacher_id = request.form['teacherid']
38                 db.save_teacher(teacher_name, specialization, teacher_id)
39             else:
40                 db.save_teacher(teacher_name, specialization)
41
42             return redirect(url_for('teacher'))
43
44         elif request.method == 'POST':
45             msg = 'Please fill out the form fields!'
46             return redirect(url_for('teacher'))
47
48     return redirect(url_for('login'))
49
50 @app.route("/delete_teacher", methods=['GET'])
51 def delete_teacher():
52     if 'loggedin' in session:
53         teacher_id = request.args.get('teacher_id')
54         db.delete_teacher(teacher_id)
55         return redirect(url_for('teacher'))
56     return redirect(url_for('login'))

```

3.1.2 Module Quản lý môn học:

- Cho phép thêm mới, sửa đổi và xóa thông tin môn học.

```

1  @app.route("/subject", methods=['GET', 'POST'])
2  def subject():
3      if 'loggedin' in session:
4          subjects = db.get_all_subjects()
5          return render_template("subject.html", subjects = subjects)
6      return redirect(url_for('login'))
7
8  @app.route("/edit_subject", methods=['GET'])
9  def edit_subject():
10     if 'loggedin' in session:
11         subject_id = request.args.get('subject_id')
12
13         subject = db.get_subject_by_id(subject_id)
14
15         if subject:
16             subject_info = subject[0]
17             return render_template("edit_subject.html", subject=subject_info)
18         else:
19             return "Subject not found", 404
20
21     return redirect(url_for('login'))
22
23
24 @app.route("/save_subject", methods=['POST'])
25 def save_subject():
26     if 'loggedin' in session:
27
28         if request.method == 'POST' and 'subject' in request.form and 's_type' in request.form and 'code' in request.form:
29             subject = request.form['subject']
30             s_type = request.form['s_type']
31             code = request.form['code']
32             action = request.form['action']
33
34             if action == 'updateSubject':
35                 subject_id = request.form['subjectid']
36                 db.save_subject(subject, s_type, code, subject_id)
37             else:
38                 db.save_subject(subject, s_type, code)
39             return redirect(url_for('subject'))
40         elif request.method == 'POST':
41             msg = 'Please fill out the form field !'
42             return redirect(url_for('subject'))
43     return redirect(url_for('login'))
44
45 @app.route("/delete_subject", methods=['GET'])
46 def delete_subject():
47     if 'loggedin' in session:
48         subject_id = request.args.get('subject_id')
49         db.delete_subject(subject_id)
50         return redirect(url_for('subject'))
51     return redirect(url_for('login'))

```

3.1.3 Module Quản lý lớp học:

- Quản lý các lớp học, bao gồm thông tin về lớp học, giáo viên giảng dạy và số lượng học sinh.


```

1 @app.route("/classes", methods=['GET', 'POST'])
2 def classes():
3     if 'loggedin' in session:
4
5         classes = db.get_all_classes()
6         sections = db.get_all_sections()
7         teachers = db.get_all_teachers()
8
9         return render_template("class.html", classes=classes, sections=sections, teachers=teachers)
10
11     return redirect(url_for('login'))
12
13 @app.route("/edit_class", methods=['GET'])
14 def edit_class():
15     if 'loggedin' in session:
16         class_id = request.args.get('class_id')
17
18         class_data = db.get_class_by_id(class_id)
19
20         if class_data:
21             class_info = class_data[0]
22
23             sections = db.get_all_sections()
24             teachers = db.get_all_teachers()
25
26             return render_template("edit_class.html", class_data=class_info, sections=sections, teachers=teachers)
27         else:
28             return "Class not found", 404
29
30     return redirect(url_for('login'))
31
32 @app.route("/save_class", methods=['POST'])
33 def save_class():
34     if 'loggedin' in session:
35
36         if request.method == 'POST' and 'cname' in request.form and 'sectionid' in request.form and 'teacherid' in request.form:
37             cname = request.form['cname']
38             sectionid = request.form['sectionid']
39             teacherid = request.form['teacherid']
40             action = request.form['action']
41
42             if action == 'updateClass':
43                 class_id = request.form['classid']
44                 db.save_class(cname, sectionid, teacherid, class_id)
45             else:
46                 db.save_class(cname, sectionid, teacherid)
47
48             return redirect(url_for('classes'))
49
50         elif request.method == 'POST':
51             msg = 'Please fill out the form fields!'
52             return redirect(url_for('classes'))
53
54     return redirect(url_for('login'))
55
56
57 @app.route("/delete_class", methods=['GET'])
58 def delete_class():
59     if 'loggedin' in session:
60         class_id = request.args.get('class_id')
61
62         db.delete_class(class_id)
63
64         return redirect(url_for('classes'))
65
66     return redirect(url_for('login'))

```

3.1.4 Module Quản lý học sinh:

- Quản lý thông tin học sinh, bao gồm mã số học sinh, họ tên, lớp học, môn học đang theo học.

```

1 @app.route("/student", methods=['GET', 'POST'])
2 def student():
3     if 'loggedin' in session:
4
5         students = db.get_all_students()
6         classes = db.get_all_classes()
7         sections = db.get_all_sections()
8
9         return render_template("student.html", students=students, classes=classes, sections=sections)
10    return redirect(url_for('login'))
11
12 @app.route("/edit_student", methods=['GET'])
13 def edit_student():
14     if 'loggedin' in session:
15         student_id = request.args.get('student_id')
16
17         student = db.get_student_by_id(student_id)
18
19         classes = db.get_all_classes()
20         sections = db.get_all_sections()
21
22
23         if student:
24             return render_template("edit_student.html", student=student, classes=classes, sections=sections)
25         else:
26             return "Student not found", 404
27    return redirect(url_for('login'))
28
29
30 @app.route("/save_student", methods=['POST'])
31 def save_student():
32     if 'loggedin' in session:
33
34         if request.method == 'POST':
35             student_data = {
36                 'name': request.form['name'],
37                 'gender': request.form['gender'],
38                 'dob': request.form['dob'],
39                 'mobile': request.form['mobile'],
40                 'email': request.form['email'],
41                 'current_address': request.form['current_address'],
42                 'class_id': request.form['class_id'],
43                 'section_id': request.form['section_id'],
44                 'stream': request.form.get('stream'),
45                 'admission_date': request.form['admission_date'],
46                 'academic_year': request.form['academic_year']
47             }
48             action = request.form['action']
49
50             if action == 'updateStudent':
51                 student_id = request.form['student_id']
52                 db.save_student(student_data, student_id)
53             else:
54                 db.save_student(student_data)
55
56             return redirect(url_for('student'))
57
58             return redirect(url_for('student'))
59    return redirect(url_for('login'))
60
61 @app.route("/delete_student", methods=['GET'])
62 def delete_student():
63     if 'loggedin' in session:
64         student_id = request.args.get('student_id')
65
66         db.delete_student(student_id)
67
68         return redirect(url_for('student'))
69    return redirect(url_for('login'))

```

3.1.5 Module Đăng nhập và phân quyền:

- Cho phép người dùng đăng nhập và phân quyền dựa trên vai trò người dùng (quản trị viên, giáo viên, học sinh).

```

1 @app.route('/')
2 @app.route('/login', methods=['GET', 'POST'])
3 def login():
4     message = ''
5     if request.method == 'POST' and 'email' in request.form and 'password' in request.form:
6         email = request.form['email']
7         password = request.form['password']
8         user = db.get_user_by_email_and_password(email, password)
9         if user:
10             session['loggedin'] = True
11             session['userid'] = user[0]
12             session['name'] = user[1]
13             session['email'] = user[3]
14             session['role'] = user[6]
15             message = 'Logged in successfully!'
16             return redirect(url_for('dashboard'))
17         else:
18             message = 'Please enter correct email/password!'
19     return render_template('login.html', message=message)
20
21 @app.route('/logout')
22 def logout():
23     session.clear()
24     return redirect(url_for('login'))
25
26 @app.route("/dashboard", methods=['GET', 'POST'])
27 def dashboard():
28     if 'loggedin' in session:
29         return render_template("dashboard.html")
30     return redirect(url_for('login'))
31
32 @app.route('/register', methods=['GET', 'POST'])
33 def register():
34     message = ''
35     if request.method == 'POST':
36
37         first_name = request.form['first_name']
38         last_name = request.form['last_name']
39         email = request.form['email']
40         password = request.form['password']
41         confirm_password = request.form['confirm_password']
42         mobile = request.form['mobile']
43         user_type = request.form['type']
44
45         if password != confirm_password:
46             message = 'Passwords do not match!'
47         else:
48             if db.get_user_by_email_and_password(email, password):
49                 message = 'Email is already registered!'
50             else:
51                 hashed_password = password
52
53                 db.create_user(first_name, last_name, email, hashed_password, mobile, user_type)
54                 message = 'Registration successful! Please log in.'
55                 return redirect(url_for('login'))
56
57     return render_template('register.html', message=message)

```

3.2 Quy trình triển khai

3.2.1 Giai đoạn 1: Xây dựng cơ sở dữ liệu

- Thiết kế các bảng trong MySQL: sms_teacher, sms_student, sms_class, sms_subject, sms_section.
- Cài đặt các quan hệ giữa các bảng: mỗi giáo viên có thể dạy nhiều môn học, mỗi lớp học có thể có nhiều học sinh, v.v.

3.2.2 Giai đoạn 2: Phát triển Backend

- Cài đặt Flask, kết nối với MySQL và xây dựng các route để xử lý các yêu cầu từ người dùng (thêm mới, sửa đổi, xóa).
- Tạo các API để giao tiếp giữa frontend và backend.

3.2.3 Giai đoạn 3: Phát triển Frontend

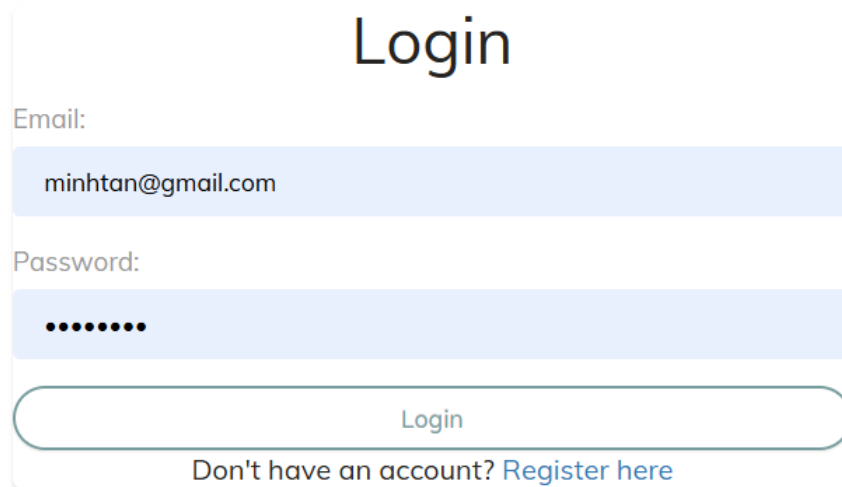
- Xây dựng giao diện người dùng bằng HTML, CSS và Bootstrap.
- Tạo các form để nhập liệu cho các thông tin như giáo viên, học sinh, môn học, lớp học.

3.2.4 Giai đoạn 4: Kiểm thử và triển khai

- Kiểm thử tính năng của hệ thống.
- Triển khai lên môi trường server để sử dụng thực tế.

3.3 Giao diện

3.3.1 Trang login



The image shows a login form with a light gray background. At the top, the word "Login" is centered in a large, dark font. Below it, there are two input fields. The first is labeled "Email:" and contains the text "minhtan@gmail.com". The second is labeled "Password:" and contains a series of dots representing a masked password. Below these fields is a rounded rectangular button with the text "Login". At the bottom of the form, there is a link that says "Don't have an account? Register here".

3.3.2 Trang register

Register

First Name:

Last Name:

Email:

Password:

Confirm Password:

Mobile:

User Type:


General


Register


Already have an account? [Login here](#)


3.3.3 Trang dashboard


SCHOOL MANAGEMENT


 DASHBOARD

 STUDENT


 CLASSES

 SECTIONS


 TEACHER

 SUBJECTS


School Management System




Welcome to your Admin Dashboard!!

 Students


→ View

 Teachers


→ View

 Classes

→ View

 Subjects

→ View

 Sections

→ View

3.3.4 Mục Sinh Viên

Student Section							
Student Admission							
ID	Name	Gender	DOB	Mobile	Class	Section	
1	minh tan	male	10/11/2024	123456	1	1	<div>EditDelete</div>

3.3.5. Mục lớp

Classes Section					
Add New Class					
ID	Name	Sections	Class Teacher		
1	CLASS01	Math0101	TanDavid	<div>Edit</div>	<div>Delete</div>
2	CLASS02	Computer0102	TanDavid	<div>Edit</div>	<div>Delete</div>

3.3.6 Mục học phần

Classes Section			
Add New Section			
ID	Name		
1	Math0101	<div>Edit</div>	<div>Delete</div>
2	Computer0102	<div>Edit</div>	<div>Delete</div>
3	Hitory02	<div>Edit</div>	<div>Delete</div>

3.3.5. Mục giảng viên

Teachers Section				
+Add				
id	Name	Subject		
1	TanDavid	Computer	Edit	Delete
2	TanHeri	Mathematics	Edit	Delete
3	TanShery	History	Edit	Delete

3.3.6 Mục môn học

Subjects Section					
					Add
ID	Subject	Code	Subject Type		
1	Computer	101	Theory	Edit	Delete
2	Mathematics	102	Theory	Edit	Delete
3	History	103	Theory	Edit	Delete

3.4 Link GitHub

https://github.com/VMT200104/SchoolManagementSystem_Flask

Chương 4: Kết luận

Hệ thống quản lý giáo dục đã đạt được mục tiêu ban đầu là cung cấp một công cụ dễ sử dụng, hiệu quả và dễ dàng mở rộng để quản lý thông tin học sinh, giáo viên, lớp học và môn học. Tuy nhiên, để đáp ứng được nhu cầu ngày càng cao của các tổ chức giáo dục, cần phát triển thêm nhiều tính năng mới và cải thiện các vấn đề về bảo mật và hiệu suất. Việc bổ sung các tính năng như báo cáo thống kê, quản lý điểm số, tích hợp học trực tuyến và nâng cao bảo mật sẽ giúp hệ thống trở thành một công cụ mạnh mẽ và hoàn thiện hơn cho quản lý giáo dục trong tương lai.

