

[illegible]

Giáo viên hướng dẫn
(Ký tên và ghi rõ họ tên)

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Kính gửi: thầy Nguyễn Bảo Ân,

Trước hết, em xin chân thành cảm ơn thầy vì sự giúp đỡ nhiệt tình trong quá trình làm đề tài “Nghiên cứu Node.js và xây dựng ứng dụng đăng ký đồ án của sinh viên công nghệ thông tin”. Nhờ sự hỗ trợ và chỉ dẫn từ cô, em đã hoàn thành công việc một cách thuận lợi hơn.

Bản thân em còn thiếu kinh nghiệm và kiến thức chuyên môn, do đó em rất mong nhận được sự thông cảm và góp ý từ thầy. Điều này sẽ giúp em rút kinh nghiệm và nâng cao khả năng thực hiện các dự án trong tương lai. Em rất biết ơn mọi hướng dẫn và góp ý từ thầy để có thể phát triển và làm tốt hơn. Xin chân thành cảm ơn sự hỗ trợ và sự chỉ dẫn của cô trong suốt thời gian qua.

Em xin hứa sẽ tiếp tục nỗ lực, không ngừng học hỏi và phấn đấu để trở thành một người có ảnh hưởng tích cực trong ngành nghề của mình.

Trân trọng,

Nguyễn Thanh Lý

MỤC LỤC

| | |
|---|----|
| CHƯƠNG 1: TỔNG QUAN | |
| 1.1 Mô tả bài toán | 8 |
| 1.2 Công cụ và cách thức thực hiện..... | 8 |
| 1.3 Xác định mô hình dữ liệu | 8 |
| 1.4 Kết quả nghiên cứu..... | 8 |
| 1.5 Kết quả đạt được..... | 8 |
| CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT | 9 |
| 2.1 Nghiên cứu RESTfull API..... | 9 |
| 2.1.1 Các thành phần của RESTfull API..... | 9 |
| 2.1.2 RESTful hoạt động như thế nào? | 10 |
| 2.2 Kiến thức về framework Express.js | 10 |
| 2.3 Tailwind CSS | 10 |
| 2.4 Hệ quản trị cơ sở dữ liệu MongoDB | 11 |
| CHƯƠNG 3 HIỆN THỰC HÓA NGHIÊN CỨU..... | 12 |
| 3.1 Đặc tả các chức năng hệ thống | 12 |
| 3.1.1 Chức năng cho giảng viên | 12 |
| 3.1.2 Chức năng cho sinh viên | 12 |
| 3.2 Thiết kế hệ thống | 13 |
| 3.2.1 Lược đồ mô tả hệ thống | 13 |
| 3.2.2 Lược đồ cơ sở dữ liệu..... | 13 |
| CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU | 25 |
| 4.1 Sản phẩm đạt được sau quá trình nghiên cứu..... | 25 |
| 4.2 Về hiệu năng..... | 25 |
| 4.3 Về trải nghiệm người dùng..... | 25 |
| 4.4 Kết quả thử nghiệm | 26 |
| CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 33 |
| 5.1 Kết luận..... | 33 |
| 5.1.1 Kết quả đạt được | 33 |
| 5.1.2 Những đóng góp mới | 33 |
| 5.1.3 Hướng phát triển..... | 33 |

DANH MỤC HÌNH

| | |
|--|----|
| Hình 1. RESTful API..... | 9 |
| Hình 2. Mô hình MVC | 13 |
| Hình 3. Lược đồ cơ sở dữ liệu..... | 13 |
| Hình 4. Mã nguồn kết nối mongodb..... | 14 |
| Hình 5. Mã nguồn xử lý yêu cầu đăng nhập..... | 15 |
| Hình 6. Mã nguồn xử lý trang đăng nhập và xử lý yêu cầu đăng ký giáo viên mới..... | 16 |
| Hình 7. Mã nguồn kiểm tra sinh viên đã đăng nhập chưa và hiển thị thông tin cá nhân của sinh viên. | 16 |
| Hình 8. Mã nguồn hiển thị trang để sinh viên đăng ký các dự án | 17 |
| Hình 9. Mã nguồn xử lý việc đăng xuất của sinh viên và xử lý việc đăng ký dự án cho sinh viên..... | 18 |
| Hình 10. Mã nguồn hiển thị trang thay đổi mật khẩu | 18 |
| Hình 11. Mã nguồn kiểm tra giáo viên đã đăng nhập chưa và hiển thị thông tin giáo viên | 19 |
| Hình 12. Mã nguồn hiển thị trang tạo tài khoản cho sinh viên..... | 19 |
| Hình 13. Mã nguồn xử lý tạo tài khoản sinh viên mới. | 20 |
| Hình 14. Mã nguồn hiển thị danh sách sinh viên và hiển thị trang thêm đề tài. | 21 |
| Hình 15. Mã nguồn xử lý tạo đề tài mới và hiển thị danh sách đề tài. | 21 |
| Hình 16. Mã nguồn xử lý đăng xuất giáo viên và hiển thị danh sách các đề tài cần phê duyệt..... | 22 |
| Hình 17. Mã nguồn hiển thị chi tiết của một dự án cần phê duyệt. | 22 |
| Hình 18. Mã nguồn xử lý phê duyệt đề tài cho sinh viên. | 23 |
| Hình 19. Mã nguồn hiển thị trang chỉnh sửa thông tin sinh viên và xử lý chỉnh sửa thông tin của một sinh viên..... | 24 |
| Hình 20. Mã nguồn hiển thị trang chỉnh sửa thông tin dự án và xử lý chỉnh sửa đồ án. | 24 |
| Hình 21. Trang chủ..... | 26 |
| Hình 22. Trang đăng nhập..... | 26 |
| Hình 23. Chức năng của giảng viên..... | 27 |
| Hình 24. Giao diện tạo đề tài | 27 |
| Hình 25. Danh sách đề tài..... | 28 |
| Hình 26. Duyệt đề tài..... | 28 |
| Hình 27. Chi tiết duyệt đề tài..... | 29 |
| Hình 28. Tạo tài khoản sinh viên..... | 29 |
| Hình 29. Danh sách sinh viên..... | 30 |
| Hình 30. Chỉnh sửa sinh viên | 30 |
| Hình 31. Chức năng cho sinh viên | 31 |
| Hình 32. Đăng ký đề tài..... | 31 |
| Hình 33. Đổi mật khẩu | 32 |

TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH

Đề tài: Nghiên cứu Node.js và xây dựng ứng dụng đăng ký đồ án cho sinh viên công nghệ thông tin

Mục tiêu: Node.js là một nền tảng phát triển ứng dụng web và di động được xây dựng chủ yếu trên ngôn ngữ lập trình JavaScript. Trong dự án xây dựng ứng dụng đăng ký đồ án cho sinh viên công nghệ thông tin, tôi đã sử dụng các công cụ như Node.js, Express.js, Tailwind css và MongoDB để tạo ra một hệ thống linh hoạt, hiệu quả và dễ bảo trì.

Hướng tiếp cận: Nghiên cứu lý thuyết: Tiếp cận vào thế giới của Node.js, Express.js, Tailwind css, MongoDB và đăng ký đồ án thông qua việc tìm hiểu sâu vào tài liệu chuyên sâu, tham gia cộng đồng nghiên cứu. Nghiên cứu thực nghiệm: Xây dựng ứng dụng quản lý lịch đối của bộ môn công nghệ thông tin.

Cách giải quyết vấn đề: Tận dụng sức mạnh của Node.js, một nền tảng linh hoạt và dễ sử dụng. Sử dụng framework Express xây dựng phần back-end một cách nhanh chóng và dễ dàng mở rộng. Sự linh hoạt và hiệu quả của MongoDB, một hệ quản trị cơ sở dữ liệu NoSQL, sẽ giúp đáp ứng tốt các yêu cầu của dự án.

Kết quả đạt được: Ứng dụng quản lý lịch đối tác được xây dựng thành công, đáp ứng được các yêu cầu đề ra. Ứng dụng có giao diện thân thiện, dễ sử dụng. Ứng dụng có khả năng mở rộng, dễ dàng tích hợp với các hệ thống khác.

Bài học kinh nghiệm: Qua trình thực hiện dự án, tôi đã nhận ra sự quan trọng của việc đọc hiểu kỹ và nghiên cứu trước về đề tài. Cần lập kế hoạch thực hiện chi tiết và khoa học. Cần kiên trì và nỗ lực để hoàn thành đề tài.

MỞ ĐẦU

1. Lý do chọn đề tài:

Hiện nay vấn đề quản lý đăng ký đề tài đồ án cho sinh viên đang gặp nhiều khó khăn quản lý và tốn rất nhiều thời gian vậy nên tôi muốn xây dựng một ứng dụng đăng ký đồ án để giải quyết vấn đề.

2. Mục đích nghiên cứu:

Để giải quyết vấn đề trên tôi đã nghiên cứu và xây dựng một ứng dụng đăng ký đồ án. Việc xây dựng một ứng dụng đăng ký đồ án giúp giải quyết một vấn đề thực tế trong hệ thống giáo dục. Nó cung cấp một cách hiệu quả để quản lý và đăng ký đồ án, làm giảm bớt công đoạn thủ công và tăng tính tự động hóa trong quá trình quản lý.

CHƯƠNG 1: TỔNG QUAN

1.1 Mô tả bài toán

Hiện nay, việc quản lý đăng ký đồ án rất phức tạp, không chỉ vậy còn tốn rất nhiều thời gian cho việc làm thủ công. Vì thế tôi đã nghiên cứu và tiến hành xây dựng ứng dụng đăng ký đồ án để giải quyết tất cả vấn đề này. Đây là giúp người dùng tối ưu hóa việc quản lý đồ án, giúp tiết kiệm thời gian và công sức.

1.2 Công cụ và cách thức thực hiện

Để có thể giải quyết được bài toán trên, em đã nghiên cứu và tìm hiểu về NodeJS, ExpressJS, Tailwind css và MongoDB, đây là một công cụ có thể thực hiện một cách tốt nhất ứng dụng đăng ký đồ án của sinh viên công nghệ thông tin.

1.3 Xác định mô hình dữ liệu

Mô hình dữ liệu gồm 3 collection: teachers, students và projects

1.4 Kết quả nghiên cứu

Sau khi hoàn thành quá trình nghiên cứu và triển khai dự án xây dựng ứng dụng đăng ký đồ án cho sinh viên ngành Công nghệ Thông tin, tôi đã thu thập được nhiều kiến thức mới và trải nghiệm sâu hơn về việc làm việc với Node.js, Express.js, Tailwind CSS, và cơ sở dữ liệu MongoDB. Những kinh nghiệm này không chỉ giúp tôi nắm vững hơn về công nghệ mà còn là bước đệm quan trọng để phát triển các dự án tiếp theo một cách hiệu quả và chuyên nghiệp hơn.

1.5 Kết quả đạt được

Sau quá trình nghiên cứu và tìm hiểu thì ứng dụng đã được thành công, với đầy đủ các chức năng cơ bản như thêm đề tài, xem danh sách đề tài, sửa đề tài, đăng ký đề tài, duyệt đề tài và đăng ký tài khoản cho sinh viên. Ứng dụng chạy nhanh chóng, giao diện thân thiện với người dùng và dễ sử dụng.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Nghiên cứu RESTfull API

RESTfull API là viết tắt của Representational State Transferful Application Programming Interface. RESTful API là một tiêu chuẩn được sử dụng để thiết kế các dịch vụ web dựa trên các phương thức HTTP như GET, POST, PUT và DELETE. Đây là một kiểu thiết kế giao diện lập trình ứng dụng (API) cho phép các ứng dụng web giao tiếp với nhau.



Hình 1. RESTful API

2.1.1 Các thành phần của RESTfull API

RESTful API được thiết kế dựa trên kiến trúc client-server, stateless và cacheable. Bản thân kiến trúc này rất đơn giản và dễ hiểu, trong đó client gửi yêu cầu đến server, server xử lý yêu cầu và trả về kết quả cho client. Tuy nhiên, để thiết kế một RESTful API tốt thì cần phải biết rõ các thành phần chính của nó.

- **Resource:** Đây là đối tượng mà chúng ta muốn truy xuất từ server. Resource được định danh bởi một URI (Uniform Resource Identifier) và có thể được truy xuất bằng HTTP methods như GET, POST, PUT, DELETE.
- **HTTP Method:** Đây là cách client gửi yêu cầu đến server. RESTful API hỗ trợ 4 phương thức chính: GET, POST, GET, DELETE. Những phương thức này phù hợp với các tác vụ cụ thể như lấy thông tin từ server (GET), tạo mới resource (POST), cập nhật resource (PUT) hay xóa resource (DELETE).

- **Representation:** Mỗi resource có thể có nhiều dạng representation khác nhau như HTML, XML, JSON,... Chúng ta có thể yêu cầu server trả về representation của resource bằng cách sử dụng HTTP header Accept.
- **Status code:** Là mã trạng thái mà server trả về sau khi xử lý yêu cầu từ client. RESTful API sử dụng mã trạng thái HTTP để chỉ ra kết quả của yêu cầu, ví dụ như 200 OK cho thành công, 404 Not Found cho không tìm thấy resource,...
- **Hypermedia:** Là các liên kết giữa các resource. Nó giúp cho việc điều hướng giữa các resource dễ dàng hơn, cũng như giúp cho client có thể tự động tìm kiếm và truy xuất các resource liên quan. Hypermedia có thể được định dạng bằng các định dạng như HTML, Atom hay JSON.

2.1.2 RESTful hoạt động như thế nào?

RESTful API hoạt động bằng cách sử dụng phương thức HTTP để truyền tải dữ liệu. Khi có yêu cầu từ ứng dụng web, API sẽ xử lý yêu cầu này và trả về kết quả dưới dạng chuẩn như **XML** hoặc **JSON**. Kiến trúc này đã được sử dụng rộng rãi trong việc phát triển các ứng dụng di động và web. RESTful API được xây dựng dựa trên các nguyên tắc thiết kế đơn giản và dễ hiểu.

2.2 Kiến thức về framework Express.js

Express là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Express hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Các tính năng nổi bật của Express:

- Thiết lập router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Hỗ trợ xây dựng theo mô hình MVC
- Cho phép định nghĩa middleware giúp tổ chức và tái sử dụng code
- Hỗ trợ RESTful API

2.3 Tailwind CSS

Tailwind CSS là một utility-first CSS framework, nó cũng giống như Bootstrap, nó có những class built-in mà chúng ta có thể dùng. Tailwind CSS có nhiều các class bao

gồm các thuộc tính CSS khác nhau và quan trọng, chúng ta có thể dễ dàng mở rộng tạo mới ra những class bằng chính những class của nó.

2.4 Hệ quản trị cơ sở dữ liệu MongoDB

MongoDB là một cơ sở dữ liệu tài liệu đa nền tảng. Được phân loại là cơ sở dữ liệu NoSQL, MongoDB sử dụng lược đồ giống tài liệu JSON. MongoDB được phát triển bởi MongoDB Inc. và được cấp phép theo Giấy phép Công cộng phía Máy chủ (SSPL). MongoDB lưu trữ dữ liệu trong các tài liệu linh hoạt, giống như JSON, có nghĩa là các trường có thể thay đổi từ tài liệu này sang tài liệu khác và cấu trúc dữ liệu có thể được thay đổi theo thời gian.

Mô hình tài liệu ánh xạ tới các đối tượng trong mã ứng dụng của bạn, giúp dữ liệu dễ dàng làm việc với.

- Truy vấn đặc biệt, lập chỉ mục và tổng hợp thời gian thực cung cấp các cách mạnh mẽ để truy cập và phân tích dữ liệu của bạn.
- MongoDB là một cơ sở dữ liệu phân tán ở cốt lõi của nó, vì vậy tính sẵn sàng cao, tỷ lệ ngang và phân phối địa lý được xây dựng và dễ sử dụng.

MongoDB là miễn phí và nguồn mở. Các phiên bản được phát hành trước ngày 16 tháng 10 năm 2018 được xuất bản theo AGPL. Tất cả các phiên bản được phát hành sau ngày 16 tháng 10 năm 2018.

CHƯƠNG 3 HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Đặc tả các chức năng hệ thống

Hệ thống có các chức năng đăng nhập nếu giảng viên đăng nhập vào thì có thể thêm đề tài, sửa đề tài, xem danh sách đề tài, duyệt đề tài, tạo tài khoản cho sinh viên và Sinh viên đăng nhập có quyền đăng ký đề tài và thay đổi mật khẩu. Giao diện người dùng sử dụng công cụ handlebars engine kết hợp với Tailwind CSS để thiết kế các form và các nút. Giao tiếp giữa backend và giao thức HTTPS được đảm bảo thông qua RESRful API. Sử dụng MongoDB để lưu trữ thông tin, hệ thống này đạt được sự linh hoạt và hiệu quả bằng cách kết hợp NodeJS với cơ sở dữ liệu.

3.1.1 Chức năng cho giảng viên

Khi giảng viên đăng nhập hệ thống sẽ chuyển đến trang quản lý đề tài trong này giảng viên có thể thêm một đề tài mới với các thông tin như giáo viên hướng dẫn, tên đề tài, chú ý và thời hạn đăng ký.

Xem danh sách đề tài: Giảng viên có thể xem danh sách đề tài của mình đã tạo và bao gồm tất cả các đề tài của giáo viên khác.

Duyệt đề tài: Giảng viên có thể duyệt các đề tài do mình tạo ra khi có sinh viên muốn đăng ký.

Tạo tài khoản cho sinh viên: Tài khoản của sinh viên đăng nhập do giáo viên cung cấp, sinh viên không thể tự đăng ký tài khoản cho mình. Thông tin của tài khoản sinh viên bao gồm: Mã số sinh viên, mật khẩu, họ tên sinh viên, giới tính và năm sinh.

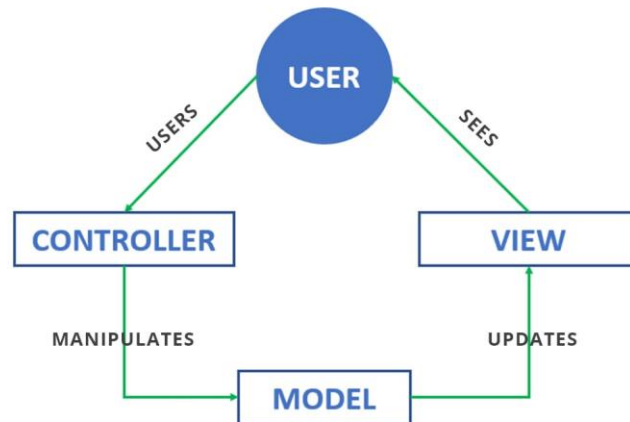
3.1.2 Chức năng cho sinh viên

Khi sinh viên đăng nhập hệ thống sẽ chuyển trang đến trang sinh viên trong này sinh viên có thể lựa chọn đăng ký đề tài do các giảng viên đề ra.

Ngoài ra, sinh viên có thể thay đổi mật khẩu khi giáo viên cấp tài khoản.

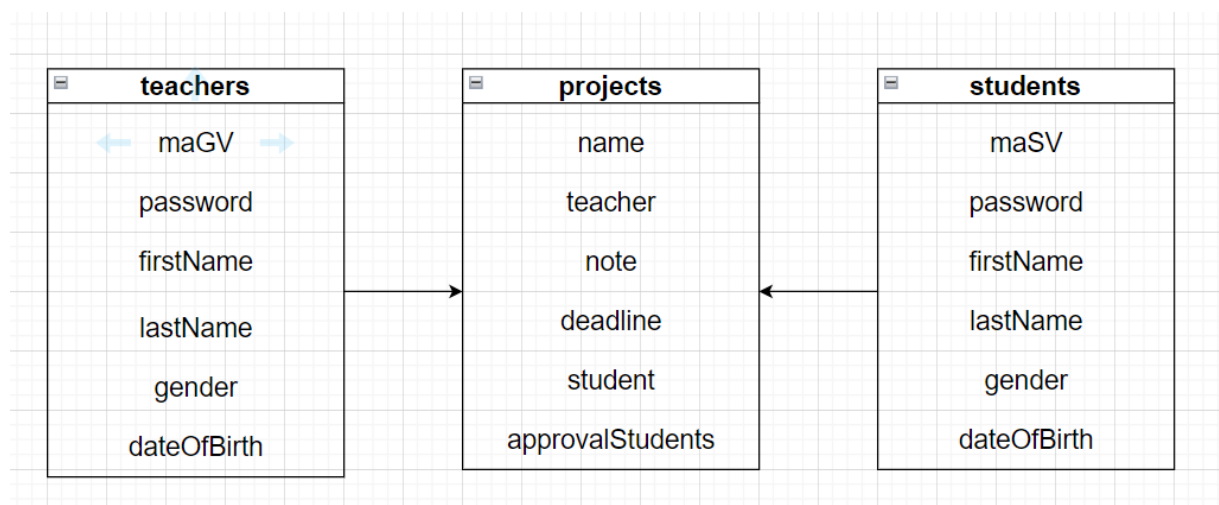
3.2 Thiết kế hệ thống

3.2.1 Lược đồ mô tả hệ thống



Hình 2. Mô hình MVC

3.2.2 Lược đồ cơ sở dữ liệu



Hình 3. Lược đồ cơ sở dữ liệu

Bảng "teachers" sẽ lưu trữ thông tin đăng nhập của giảng viên.

Bảng "students" sẽ lưu trữ thông tin đăng nhập của sinh viên.

Bảng "projects" sẽ lưu các thông tin của đề tài.

Mã nguồn kết nối mongodb

```
async function connectDB(url) {
  try {
    await mongoose.connect(url)
    const teachers = await Teacher.find({})
    if (teachers.length == 0) {
      const newTeacher1 = new Teacher({
        maGV: 'GV001',
        password: '123456',
        firstName: 'Nguyễn',
        lastName: 'Bảo Ân',
        gender: 'Nam',
        dateOfBirth: '07/07/1999',
      })
      await newTeacher1.save()
      const newTeacher2 = new Teacher({
        maGV: 'GV002',
        password: '123456',
        firstName: 'Kim',
        lastName: 'Linh',
        gender: 'Nữ',
        dateOfBirth: '07/07/1979',
      })
      await newTeacher2.save()
    }
  } catch (err) {
    console.error('Could not connect to MongoDB', err)
  }
}

connectDB('mongodb://127.0.0.1:27017/dkdoan')
```

Hình 4. Mã nguồn kết nối mongodb

Mã nguồn này khi kết nối với mongodb và khởi động dự án hệ thống sẽ tự tạo ra 2 tài khoản dành cho giáo viên.

Mã nguồn xử lý yêu cầu đăng nhập

```
5 //xử lý yêu cầu đăng nhập
6 exports.login = async function(req, res) {
7   const username = req.body.username
8   const password = req.body.password
9   const isTeacher = req.body.isTeacher
10
11   if (isTeacher == 'on') {
12     const teacher = await Teacher.findOne({
13       maGV: username,
14       password: password
15     })
16     if (teacher) {
17       req.session.teacher = teacher
18       res.redirect('/teachers')
19     }
20     else {
21       res.redirect('login?error=invalid')
22     }
23   }
24   else {
25     const student = await Student.findOne({
26       maSV: username,
27       password: password
28     })
29     if (student) {
30       req.session.student = student
31       res.redirect('/students')
32     }
33     else {
34       res.redirect('login?error=invalid')
35     }
36   }
37 }
38
39
40 }
41
```

Hình 5. Mã nguồn xử lý yêu cầu đăng nhập

Mã nguồn xử lý trang đăng nhập và xử lý yêu cầu đăng ký giáo viên mới.

```
42
43 //xử lý hiển thị trang đăng nhập
44 exports.getLogin = async function(req, res) {
45   const error = req.query.error // check xem có error không.
46   if (error == 'invalid') {
47     res.render('login', {error: 'invalid login!'})
48   }
49   else {
50     res.render('login')
51   }
52 }
53
54
55 //xử lý yêu cầu đăng ký giáo viên mới
56 exports.register = async function(req, res) {
57   // const username = req.body.username
58   // const password = req.body.password
59
60   const newTeacher = new Teacher({
61     maGV: 'GV001',
62     password: '123',
63     firstName: 'Alice',
64     lastName: 'Smith',
65     gender: 'Female',
66     dateOfBirth: '1985-05-10',
67     classIds: []
68   })
69   try {
70     const result = await newTeacher.save()
71     res.json(result)
72   }
73   catch(error) {
74     res.json(error)
75   }
76
77 }
```

Hình 6. Mã nguồn xử lý trang đăng nhập và xử lý yêu cầu đăng ký giáo viên mới.

Mã nguồn kiểm tra sinh viên đã đăng nhập chưa và hiển thị thông tin cá nhân của sinh viên.

```
3 //kiểm tra xem một sinh viên đã đăng nhập chưa
4 function checkAuthStudent(req, res) {
5   const student = req.session.student
6   if (!student) return false
7   return true
8 }
9
10 //hiển thị thông tin cá nhân của sinh viên
11 exports.getStudents = async (req, res) => {
12   if (!checkAuthStudent(req, res)) {
13     res.render('unauthorized')
14     return
15   }
16   res.render('student/profile', {student: req.session.student})
17
18 }
```

Hình 7. Mã nguồn kiểm tra sinh viên đã đăng nhập chưa và hiển thị thông tin cá nhân của sinh viên.

Mã nguồn hiển thị trang để sinh viên đăng ký các dự án

```
19 //hiển thị trang để sinh viên đăng ký các dự án
20 exports.getRegisterProject = async (req, res) => {
21   if (!checkAuthStudent(req, res)) {
22     res.render('unauthorized')
23     return
24   }
25
26   const student = req.session.student
27   const idStudent = student._id
28
29
30   const projectApproval = await Project.findOne({
31     student: idStudent // + tìm kiếm
32   }).populate('student').populate('teacher').exec()
33
34
35
36
37   // đề tài đã được duyệt ( có student )
38   const projectsApproached = await Project.find({ student: { $ne: null } })
39     .populate('student') // lấy ra thông tin của sinh viên được duyệt
40     .populate('teacher')
41     .exec()
42
43   // đề tài chưa được đăng ký ( không có student )
44   const projectsNotResgiter= await Project.find({
45     student: null,
46     approvalStudents: { $nin: [idStudent] }
47   })
48     .populate('teacher')
49     .exec()
50
51   // lấy ra danh sách đề tài sinh viên đăng ký
52   const projectsResgiter= await Project.find({
53     student: null,
54     approvalStudents: idStudent
55   })
56     .populate('teacher')
57     .exec()
58
59
60   // tiếp theo kiểm tra xem sinh viên này đã được duyệt đề tài hay chưa. nếu được duyệt rồi -> không cho đăng ký.
61
62
63   if (projectApproval) {
64     for (let i = 0; i < projectsNotResgiter.length; i++) {
65       projectsNotResgiter[i].isApproached = true
66     }
67   }
68
69
70
71
72   res.render('student/registerProject', {projectApproval, projectsApproached, projectsNotResgiter, projectsResgiter})
73
74
75 }
```

Hình 8. Mã nguồn hiển thị trang để sinh viên đăng ký các dự án

Mã nguồn xử lý việc đăng xuất của sinh viên và xử lý việc đăng ký dự án cho sinh viên.

```
122 //xử lý việc đăng xuất sinh viên
123 exports.handleLogout = async (req, res) => {
124
125     // xóa đi và chuyển về người trang chủ.
126     if (!checkAuthStudent(req, res)) {
127         res.render('unauthorized')
128         return
129     }
130     req.session.student = null
131     res.redirect('/')
132 }
133 //xử lý việc đăng ký dự án cho sinh viên
134 exports.registerProject = async (req, res) => {
135     if (!checkAuthStudent(req, res)) {
136         res.render('unauthorized')
137         return
138     }
139
140     const idProject = req.body.idProject
141     const project = await Project.findById(idProject).populate('student')
142     const student = req.session.student
143     project.approvalStudents.push(student._id)
144     await project.save()
145     res.redirect('/students/projects') // chuyển người dùng về lại giao diện đăng ký
146 }
```

Hình 9. Mã nguồn xử lý việc đăng xuất của sinh viên và xử lý việc đăng ký dự án cho sinh viên.

Mã nguồn hiển thị trang thay đổi mật khẩu

```
//hiển thị trang để sinh viên thay đổi mật khẩu
exports.getChangePassword = async(req, res) => {
    if (!checkAuthStudent(req, res)) {
        res.render('unauthorized')
        return
    }
    let message = ""
    if (req.query.error == 'password') {
        message = 'Password và confirm password không giống nhau!'
        console.log(message)
        res.render('student/changePassword', {message})
        return
    }
    else if (req.query.success == 'true') {
        message = 'Thay đổi mật khẩu thành công!'
        res.render('student/changePassword', {message})
        return
    }
    res.render('student/changePassword')
}
```

Hình 10. Mã nguồn hiển thị trang thay đổi mật khẩu

Mã nguồn kiểm tra giáo viên đã đăng nhập chưa và hiển thị thông tin giáo viên

```
6 //kiểm tra xem giáo viên đã đăng nhập hay chưa
7 function checkAuthTeacher(req, res) {
8     const teacher = req.session.teacher
9     if (!teacher) {
10         return false
11     }
12     return true
13 }
14
15 //hiển thị thông tin cá nhân của giáo viên
16 exports.getTeacher = async (req, res) => {
17
18     if (!checkAuthTeacher(req, res)) {
19         res.render('unauthorized')
20         return
21     }
22     const teacher = req.session.teacher
23     res.render('teacher/profile', {teacher})
24 }
```

Hình 11. Mã nguồn kiểm tra giáo viên đã đăng nhập chưa và hiển thị thông tin giáo viên

Mã nguồn hiển thị trang tạo tạo khoản cho sinh viên

```
25 //hiển thị trang để giáo viên tạo sinh viên mới
26 exports.createStudent = async (req, res) => {
27     if (!checkAuthTeacher(req, res)) {
28         res.render('unauthorized')
29         return
30     }
31
32     // lấy ra từ param
33     let message = null
34     if (req.query.coincide == 'student') {
35         message = "Trùng mã số sinh viên!"
36     }
37     else if (req.query.coincide == 'teacher') {
38         message = "Trùng mã số giáo viên!"
39     }
40
41     else if (req.query.error == 'password') {
42         message = "Mật khẩu mà mật khẩu xác nhận không giống nhau!"
43     }
44
45     const error = req.query.error
46     if (error == 'invalid') res.render('createStudent', {error: 'invalid credentials'})
47     else res.render('createStudent', {message})
48 }
49
```

Hình 12. Mã nguồn hiển thị trang tạo tạo khoản cho sinh viên

Mã nguồn xử lý tạo tài khoản sinh viên mới.

```
51 // xử lý việc tạo sinh viên mới
52 exports.createStudent = async (req, res) => {
53   if (!checkAuthTeacher(req, res)) {
54     res.render('unauthorized')
55     return
56   }
57
58   // trước tiên check với lỗi xem có trùng không đã.
59   const password = req.body.password
60   const confirmPassword = req.body.confirmPassword
61   if (password !== confirmPassword) {
62     res.redirect('/teachers/students/create?error=password')
63     return
64   }
65
66   // kiểm tra mã sinh viên.
67   const maSV = req.body.maSV
68   // kiểm tra xem có tồn tại mã sinh viên nào không.
69   const student = await Student.findOne({
70     maSV // tìm kiếm theo mã sinh viên
71   })
72   const teacher = await Teacher.findOne({
73     maGV: maSV
74   })
75
76   console.log(student + " student")
77
78
79   if (student !== null) {
80     console.log("Vào đây")
81     res.redirect('/teachers/students/create?coincide=student')
82     return
83   }
84   else if (teacher !== null) {
85     res.redirect('/teachers/students/create?coincide=teacher')
86     return
87   }
88
89   const newStudent = new Student({
90     maSV: req.body.maSV,
91     password: req.body.password,
92     firstName: req.body.firstName,
93     lastName: req.body.lastName,
94     gender: req.body.gender,
95     dateOfBirth: req.body.dateOfBirth,
96   })
97   await newStudent.save()
98   res.redirect('/teachers/students')
99 }
100
```

Hình 13. Mã nguồn xử lý tạo tài khoản sinh viên mới.

Mã nguồn hiển thị danh sách sinh viên và hiển thị trang thêm đề tài.

```
101 // hiển thị danh sách sinh viên
102 exports.getStudents = async (req, res) => {
103   if (!checkAuthTeacher(req, res)) {
104     res.render('unauthorized')
105     return
106   }
107   const students = await Student.find({})
108   res.render('students', {students: students})
109 }
110
111 //hiển thị trang để giáo viên tạo dự án mới
112 exports.getCreateProject = async (req, res) => {
113   if (!checkAuthTeacher(req, res)) {
114     res.render('unauthorized')
115   }
116   else {
117     const fullNameTeacher = req.session.teacher.firstName + " " + req.session.teacher.lastName
118     res.render('createProject', {fullNameTeacher})
119   }
120 }
121
```

Hình 14. Mã nguồn hiển thị danh sách sinh viên và hiển thị trang thêm đề tài.

Mã nguồn xử lý tạo đề tài mới và hiển thị danh sách đề tài.

```
122 //xử lý việc tạo dự án mới
123 exports.createProject = async (req, res) => {
124   if (!checkAuthTeacher(req, res)) {
125     res.render('unauthorized')
126   }
127   else {
128     await new Project({
129       name: req.body.name,
130       note: req.body.note,
131       teacher: req.session.teacher._id,
132       deadline: req.body.time + " " + req.body.date,
133       student: null,
134       approvalStudents: [] // chưa có sv nào !.
135     }).save()
136     res.redirect('/teachers/projects')
137   }
138 }
139
140
141 // hiển thị danh sách các dự án
142 exports.getProjects = async (req, res) => {
143   if (!checkAuthTeacher(req, res)) {
144     res.render('unauthorized')
145     return
146   }
147   try {
148     const projects = await Project.find({})
149       .populate('teacher')
150       .populate('student')
151       .exec()
152     res.render('projects', {projects})
153   } catch (err) {
154     res.render('error', {message: err.message})
155   }
156 }
```

Hình 15. Mã nguồn xử lý tạo đề tài mới và hiển thị danh sách đề tài.

Mã nguồn xử lý đăng xuất giáo viên và hiển thị danh sách các đề tài cần phê duyệt.

```
157 //xử lý việc đăng xuất giáo viên
158 exports.handleLogout = async (req, res) => {
159   if (!checkAuthTeacher(req, res)) {
160     res.render('unauthorized')
161     return
162   }
163   req.session.teacher = null
164   res.redirect('/')
165 }
166 //hiển thị danh sách các dự án cần phê duyệt cho giáo viên
167 exports.getApproval = async (req, res) => {
168   if (!checkAuthTeacher(req, res)) {
169     res.render('unauthorized')
170     return
171   }
172
173
174   const idTeacher = req.session.teacher._id
175   const projects = await Project.find({
176     teacher: idTeacher,
177     student: { $exists: true, $eq: null }
178   }).populate('approvalStudents').exec()
179   res.render('teacher/approvalProject', {projects})
180 }
181
```

Hình 16. Mã nguồn xử lý đăng xuất giáo viên và hiển thị danh sách các đề tài cần phê duyệt.

Mã nguồn hiển thị chi tiết của một dự án cần phê duyệt.

```
184 // hiển thị chi tiết của một dự án cần phê duyệt
185 exports.getApprovalDetail = async (req, res) => {
186   if (!checkAuthTeacher(req, res)) {
187     res.render('unauthorized')
188     return
189   }
190   let message = null
191   if (req.query.studentApproved == 'true') {
192     message = "Sinh viên đã được duyệt đồ án rồi, vui lòng chọn sinh viên khác!"
193   }
194   const idProject = req.params.id
195   const project = await Project.findById(idProject).populate('approvalStudents').exec()
196   res.render('teacher/detailApprovalProject', {project, message})
197
198
199
200
201 //   res.render('teacher/approvalProject', {projects})
202 }
```

Hình 17. Mã nguồn hiển thị chi tiết của một dự án cần phê duyệt.

Mã nguồn xử lý phê duyệt đề tài cho sinh viên.

```
203 //xử lý việc phê duyệt dự án cho sinh viên
204 exports.approvalDetail = async (req, res) => {
205   if (!checkAuthTeacher(req, res)) {
206     res.render('unauthorized')
207     return
208   }
209
210   const idStudent = req.body.idStudent
211   const idProject = req.params.id
212
213
214   // trước tiên kiểm tra xem sinh viên đó đã được duyệt trong đồ án nào chưa để khỏi bị nhầm lẫn.
215   // làm sao tìm kiếm => sử dụng findOne
216
217   const project = await Project.findOne({
218     student: idStudent
219   })
220
221   if (project) {
222     res.redirect('/teachers/projects/approval/' + idProject + "?studentApproved=true")
223     return
224   }
225
226
227   const projectApproval = await Project.findById(idProject).populate('student').exec()
228   projectApproval.student = idStudent
229   await projectApproval.save()
230
231
232
233   res.redirect('/teachers/projects/approval')
234 }
235
```

Hình 18. Mã nguồn xử lý phê duyệt đề tài cho sinh viên.

Mã nguồn hiển thị trang chỉnh sửa thông tin sinh viên và xử lý chỉnh sửa thông tin của một sinh viên.

```
235
236 //hiển thị trang chỉnh sửa thông tin của một sinh viên
237 exports.getEditStudent = async (req, res) => {
238   if (!checkAuthTeacher(req, res)) {
239     res.render('unauthorized')
240     return
241   }
242   const idStudent = req.params.id
243   const student = await Student.findById(idStudent)
244   res.render('teacher/editStudent', {student})
245 }
246 //xử lý việc chỉnh sửa thông tin của một sinh viên
247 exports.editStudent = async(req, res) => {
248   if (!checkAuthTeacher(req, res)) {
249     res.render('unauthorized')
250     return
251   }
252   const idStudent = req.params.id
253   const student = await Student.findById(idStudent)
254   student.firstName = req.body.firstName
255   student.lastName = req.body.lastName
256   student.gender = req.body.gender
257   student.dateOfBirth = req.body.dateOfBirth
258   await student.save()
259   res.redirect('/teachers/students')
260 }
261
```

Hình 19. Mã nguồn hiển thị trang chỉnh sửa thông tin sinh viên và xử lý chỉnh sửa thông tin của một sinh viên.

Mã nguồn hiển thị trang chỉnh sửa thông tin dự án và xử lý chỉnh sửa đồ án.

```
262 //Hàm này hiển thị trang chỉnh sửa thông tin của một dự án
263 exports.getEditProject = async (req, res) => {
264   if (!checkAuthTeacher(req, res)) {
265     res.render('unauthorized')
266     return
267   }
268   const idProject = req.params.id
269   const project = await Project.findById(idProject).populate('teacher').exec()
270   // chia cắt chuỗi.
271   let timeDate = project.deadline.split(" ")
272   const time = timeDate[0]
273   const date = timeDate[1]
274   res.render('teacher/editProject', {
275     project, time, date
276   })
277 }
278 //xử lý việc chỉnh sửa thông tin của một dự án
279 exports.editProject = async (req, res) => {
280   if (!checkAuthTeacher(req, res)) {
281     res.render('unauthorized')
282     return
283   }
284   const idProject = req.params.id
285   const project = await Project.findById(idProject)
286   project.name = req.body.name
287   project.note = req.body.note
288   project.deadline = req.body.time + " " + req.body.date
289   await project.save()
290   res.redirect('/teachers/projects')
291 }
292
```

Hình 20. Mã nguồn hiển thị trang chỉnh sửa thông tin dự án và xử lý chỉnh sửa đồ án.

CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU

4.1 Sản phẩm đạt được sau quá trình nghiên cứu

Ứng dụng đăng kí đồ án được xây dựng thành công với đầy đủ các chức năng cơ bản như:

Chức năng cho giảng viên: Thêm đề tài, sửa đề tài, xem danh sách đề tài, duyệt đề tài, tạo tài khoản cho sinh viên.

Chức năng cho sinh viên: Đăng ký đề tài và thay đổi mật khẩu.

Đáp ứng đầy đủ các nhu cầu được đề ra. Ứng dụng chạy nhanh chóng, giao diện thân thiện dễ sử dụng.

4.2 Về hiệu năng

Để tối ưu hóa hiệu năng của ứng dụng đăng ký đồ án trên Node.js: Sử dụng cache, tối ưu hóa cơ sở dữ liệu và gọi API bất đồng bộ. Tối ưu hóa mã nguồn, sử dụng compression và minification. Kiểm soát lượng dữ liệu trả về và tối ưu hóa giao diện người dùng. Sử dụng công nghệ CDN và kiểm thử hiệu năng thường xuyên.

4.3 Về trải nghiệm người dùng

Tối ưu hóa các tính năng, giao diện thân thiện với người dùng, đảm bảo người dùng dễ dàng tiếp cận.

Hiện thị đầy đủ các thông tin của giáo viên và sinh viên.

4.4 Kết quả thử nghiệm

Trang chủ



Hình 21. Trang chủ

Trang này sẽ có giao diện là hình ảnh của Trường đại học Trà Vinh và phía bên trái là logo của trường, bên phải là nút đăng nhập.

Trang đăng nhập

Hình 22. Trang đăng nhập

Trang này sẽ hiển thị khi người dùng nhấn phím đăng nhập từ trang chủ, trang bao gồm có phần đăng nhập là Username (mã giảng viên hoặc mã sinh viên), password và có ô click chọn nếu người dùng đăng nhập là giảng viên thì chọn vào và người lại nếu là sinh viên thì không cần chọn.

Chức năng của giảng viên

The screenshot shows the 'Dashboard teacher' interface. On the left is a blue sidebar with a list of menu items: Profile, Trang chủ, Tạo đề tài, Duyệt đề tài, Danh sách đề tài, Tạo tài khoản sinh viên, Danh sách sinh viên, and Đăng xuất. The main content area is titled 'Profile giáo viên'. It contains several input fields: 'HỌ' (Last Name) with the value 'Nguyễn', 'TÊN' (First Name) with the value 'Bảo Ân', 'MÃ SINH VIÊN' (Student ID) with the value 'GV001', 'NGÀY SINH' (Date of Birth) with the value '07/07/1999', and 'GIỚI TÍNH' (Gender) with the value 'Nam'.

Hình 23. Chức năng của giảng viên

Khi người dùng đăng nhập vào là tài khoản giảng viên đầu tiên vào sẽ hiển thị ra thông tin của giảng viên đó. Trong trang của giảng viên sẽ có các chức năng tạo đề tài, duyệt đề tài, danh sách đề tài, tạo tài khoản sinh viên, danh sách sinh viên và đăng xuất.

Giao diện tạo đề tài

The screenshot shows the 'Tạo đề tài' (Create Topic) form. It includes the same blue sidebar as the previous page. The main content area has the title 'Tạo đề tài'. It contains the following fields: 'TÊN ĐỀ TÀI' (Topic Name) as a text input, 'GIÁO VIÊN HƯỚNG DẪN' (Supervising Teacher) as a dropdown menu with 'Nguyễn Bảo Ân' selected, 'GHI CHÚ' (Remarks) as a text area, 'GIỚI' (Gender) as a dropdown menu with a calendar icon, and 'NGÀY/THÁNG/NĂM' (Date/Month/Year) as a date picker with the format 'dd/mm/yyyy'. At the bottom of the form is a blue button labeled 'Tạo đề tài'.

Hình 24. Giao diện tạo đề tài

Khi giảng viên nhấn vào chức năng tạo đề tài hệ thống sẽ hiển thị ra các form bao gồm: Tên đề tài, Giảng viên hướng dẫn(tên của giảng viên đăng nhập), ghi chú, ngày và

giờ nộp báo cáo dự án khi giảng viên đã nhập đầy đủ thông tin vào form và nhấn tạo đề tài hiện thông sẽ lưu vào cơ sở dữ liệu và chuyển đến trang danh sách đề tài.

Danh sách đề tài

| Dashboard teacher | | | |
|--|---------------------|------------------|-----------------------------|
| Danh sách đề tài | | | |
| TÊN ĐỀ TÀI | GIÁO VIÊN HƯỚNG DẪN | THỜI HẠN | SINH VIÊN THỰC HIỆN |
| Nodejs xây dựng ứng dụng quản lý đối tác | Nguyễn Bảo Ân | 11:33 2024-04-24 | Nguyễn Thanh Lý - 110121061 |
| Nodejs xây dựng ứng dụng | Nguyễn Bảo Ân | 14:55 2024-04-03 | None |

Hình 25. Danh sách đề tài

Trong trang này sẽ hiển thị tất cả các đề tài do giảng viên tạo ra và có thể sửa thông tin của đề tài đó.

Duyệt đề tài

| | |
|--------------------------------------|----------|
| Dashboard teacher | |
| Duyệt đề tài | |
| Tên đề tài: Nodejs xây dựng ứng dụng | |
| Giảng viên hướng dẫn: | |
| Thời hạn: 14:55 2024-04-03 | |
| Ghi chú: học thêm nodejs | |
| Sinh viên chọn đề tài: 1 | Chi tiết |

Hình 26. Duyệt đề tài

Ở trang này sẽ hiển thị số lượng và thông tin muốn đăng ký đề tài của giảng viên đó đề ra và giảng viên chỉ có thể duyệt các đề tài do giảng viên đó tạo. Khi nhấn vào chi tiết trang sẽ hiển thị thông tin các sinh viên muốn đăng ký đề tài, nếu giáo viên duyệt thì

thông tin sinh viên sẽ thêm vào thông tin ở danh sách đề tài. Ví dụ minh họa như hình bên dưới.

The screenshot shows a web application interface. On the left is a blue sidebar with the text 'Dashboard teacher' and a graduation cap icon, followed by a list of menu items: Profile, Trang chủ, Tạo đề tài, Duyệt đề tài, Danh sách đề tài, Tạo tài khoản sinh viên, Danh sách sinh viên, and Đăng xuất. The main content area is titled 'Chi tiết duyệt đề tài'. It contains the following information: 'Tên đề tài: Nodejs xây dựng ứng dụng', 'Giảng viên hướng dẫn:', 'Thời hạn: 14:55 2024-04-03', and a red note 'Ghi chú: học thêm nodejs'. Below this, it says 'Sinh viên chọn đề tài: 1' and 'Trần Ngọc Mai - 110121062'. At the bottom right of the form is a green button labeled 'Duyệt'.

Hình 27. Chi tiết duyệt đề tài





Tạo tài khoản sinh viên

The screenshot shows the 'Tạo tài khoản sinh viên' form. It includes input fields for 'Mã số sinh viên' (with the value 110121061), 'Password', and 'Confirm password'. Below these are fields for 'First name', 'Last name', 'gender' (with a dropdown menu showing 'Nam'), and 'Ngày sinh' (with a date picker showing 'dd/mm/yyyy'). A blue button labeled 'Tạo tài khoản' is at the bottom.

Hình 28. Tạo tài khoản sinh viên


Tài khoản của sinh viên do giảng viên cấp, khi giảng viên nhấn vào tạo tài khoản cho sinh viên hệ thống sẽ hiển thị ra các form để giảng viên nhập bao gồm: mã số sinh viên, password, xác nhận password, họ và tên sinh viên, giới tính, năm sinh. Khi nhấn vào tạo tài khoản sẽ chuyển đến trang danh sách sinh viên.

Danh sách sinh viên

| Dashboard teacher  | | | | |
|---|-----------------|-----------|------------|---|
| Danh sách sinh viên | | | | |
| MÃ SINH VIÊN | HỌ VÀ TÊN | GIỚI TÍNH | NGÀY SINH | |
| 110121061 | Nguyễn Thanh Lý | Nam | 2003-06-26 |  |
| 110121062 | Trần Ngọc Mai | Nữ | 2003-01-01 |  |
| 110121063 | Nguyễn Văn A | Nam | 2003-08-13 |  |

Hình 29. Danh sách sinh viên

Trang này sẽ hiển thị thông tin của các tài khoản sinh viên và có thể chỉnh sửa thông tin, nếu muốn chỉnh sửa người dùng có thể nhấn vào nút chỉnh sửa ở bên phải mỗi dòng thông tin khi nhấn và sẽ hiển thị thông tin như hình bên dưới, khi đã chỉnh sửa và nhấn lưu hệ thống sẽ lưu lại vào cơ sở dữ liệu.

| Dashboard teacher  | |
|---|--|
| Profile | |
| Trang chủ | |
| Tạo đề tài | |
| Duyệt đề tài | |
| Danh sách đề tài | |
| Tạo tài khoản sinh viên | |
| Danh sách sinh viên | |
| Đăng xuất | |

Hiệu chỉnh student

HỌ

TÊN

Nguyễn


Thanh Lý

MÃ SINH VIÊN

110121061

NGÀY SINH

26/06/2003



GIỚI TÍNH

Nam

Lưu thay đổi

Hình 30. Chỉnh sửa sinh viên

Chức năng cho sinh viên

The screenshot shows a student dashboard with a blue sidebar on the left containing the following menu items: Dashboard student, Profile, Trang chủ, Đăng ký đề tài, Đổi mật khẩu, and Đăng xuất. The main content area is titled 'Profile sinh viên' and contains a form with the following fields: HO (Last Name) with the value 'Nguyễn', TÊN (First Name) with the value 'Thanh Lý', MÃ SINH VIÊN (Student ID) with the value '110121061', NGÀY SINH (Date of Birth) with the value '2003-06-26', and GIỚI TÍNH (Gender) with the value 'Nam'.

Hình 31. Chức năng cho sinh viên

Khi sinh viên đăng nhập vào hệ thống sẽ chuyển đến trang dành cho sinh viên có các chức năng như: Profile, đăng ký đề tài và đổi mật khẩu.

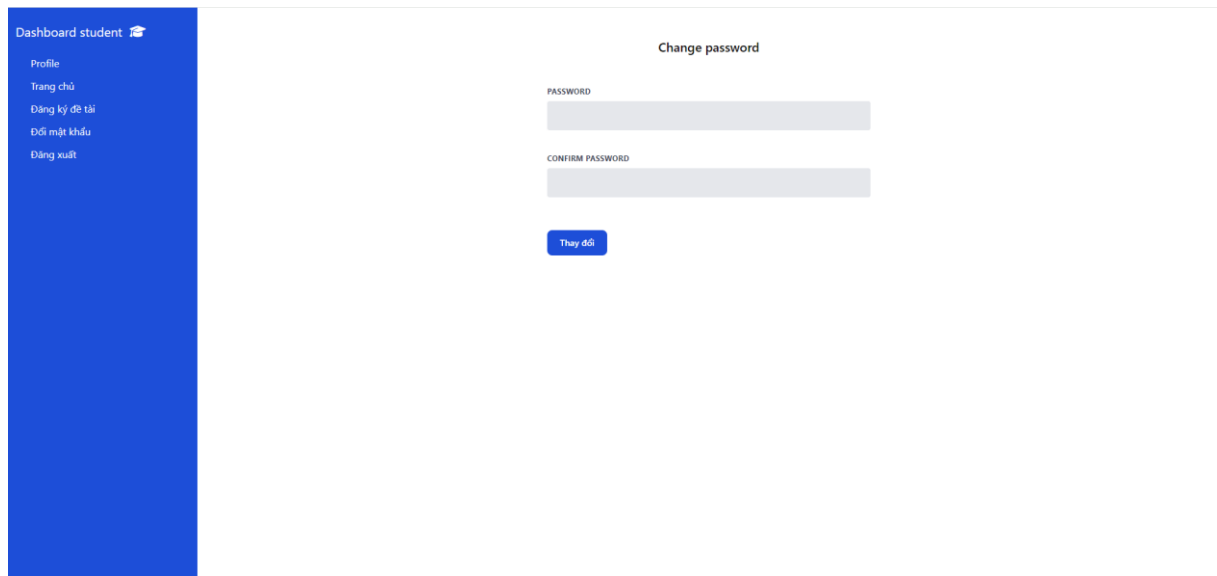
Đăng ký đề tài

The screenshot shows a student dashboard with a blue sidebar on the left containing the following menu items: Dashboard student, Profile, Trang chủ, Đăng ký đề tài, Đổi mật khẩu, and Đăng xuất. The main content area is titled 'Đăng ký đề tài' and contains a list of topics. The first two topics are already registered and have a status of 'Đã đăng ký'. The third topic is 'Website php' and has a status of 'Chưa đăng ký', with a green 'Đăng ký' button next to it. The fourth topic is 'Nodejs xây dựng ứng dụng quản lý đối tác' and has a status of 'Đã được giáo viên duyệt'. Each topic card displays the topic name, the supervising teacher's name, the deadline, and the subject.

Hình 32. Đăng ký đề tài

Trang này hiển thị các thông tin bao gồm : đề tài chưa được duyệt, đề tài đã đăng ký, đề tài chưa đăng ký và đề tài đã được giáo viên duyệt. Sinh viên có thể lựa chọn đăng ký đề tài yêu thích nhưng nếu sinh viên đã được giáo viên duyệt đề tài nào đó thì sinh viên sẽ không được đăng ký tài tài khác.

Đổi mật khẩu



The image shows a web interface for changing a password. On the left is a blue sidebar with the text 'Dashboard student' and a user icon, followed by a list of menu items: 'Profile', 'Trang chủ', 'Đăng ký đề tài', 'Đổi mật khẩu', and 'Đăng xuất'. The main content area is white and titled 'Change password'. It contains two input fields: the first is labeled 'PASSWORD' and the second is labeled 'CONFIRM PASSWORD'. Below these fields is a blue button with the text 'Thay đổi'.

Hình 33. Đổi mật khẩu

Sinh viên có thể đổi mật khẩu do giáo viên cấp, khi nhập mật khẩu mới và xác nhận mật khẩu mới và nhấn thay đổi hệ thống sẽ lưu thay đổi vào cơ sở dữ liệu.

CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.1.1 Kết quả đạt được

Ứng dụng đăng ký đồ án được xây dựng thành công với đầy đủ các chức năng cơ bản như:

Chức năng cho giảng viên: Thêm đề tài, sửa đề tài, xem danh sách đề tài, duyệt đề tài, tạo tài khoản cho sinh viên,....

Chức năng cho sinh viên: Đăng ký đề tài và thay đổi mật khẩu.

Ứng dụng chạy rất mượt mà và nhanh chóng, với giao diện thân thiện dễ dàng sử dụng.

5.1.2 Những đóng góp mới

Ứng dụng đăng ký của sinh viên bộ môn công nghệ thông tin là một ứng dụng hữu ích, góp phần nâng cao hiệu quả trong quá trình quản lý đề tài. Ứng dụng có những đóng góp mới sau:

Sử dụng các công nghệ hiện đại như Node.js, Express.js, Tailwind CSS và MongoDB giúp cho ứng dụng chạy tốt và đồng thời có thể mở rộng.

Thiết kế giao diện đơn giản, thân thiện và dễ sử dụng, giúp người dùng dễ dàng tìm thấy các chức năng cần thiết.

5.1.3 Hướng phát triển

Để nâng cao hiệu quả của ứng dụng, có thể thực hiện một số đề xuất sau:

Thêm chức năng tìm kiếm như: Tìm kiếm sinh viên, tìm kiếm đề tài,.... để giúp người dùng tiết kiệm thời gian hơn.

Cải thiện khả năng bảo mật của ứng dụng.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] [GitHub - nodejs/node: Node.js JavaScript runtime](#)
- [2] [express - npm \(npmjs.com\)](#)
- [3] [Setting up a Node development environment - Learn web development | MDN \(mozilla.org\)](#)
- [4] [How To Get Started with Node.js and Express | DigitalOcean](#)