MỤC LỤC

[Chương 1. Tổng quan 2](#_Toc84787019)

[1.1 Công nghệ 2](#_Toc84787020)

[1.1.1 Front end 2](#_Toc84787021)

[1.1.2 Back end 2](#_Toc84787022)

[1.2 Vai trò 2](#_Toc84787023)

[Chương 2. Chức năng 3](#_Toc84787024)

[2.1 Người dùng 3](#_Toc84787025)

[2.1.1 Đăng nhập 3](#_Toc84787026)

[2.1.2 Quên mật khẩu 3](#_Toc84787027)

[2.1.3 Khôi phục mật khẩu 4](#_Toc84787028)

[2.1.4 Trang cập nhật thông tin cá nhân 5](#_Toc84787029)

[2.1.5 Trang Profile 6](#_Toc84787030)

[2.1.6 Lấy danh sách môn học 6](#_Toc84787031)

[2.1.7 Upload đề thi 7](#_Toc84787032)

[2.1.8 Xem danh sách/ Xóa đề thi 7](#_Toc84787033)

[2.1.9 Upload tài liệu 8](#_Toc84787034)

[2.1.10 Trang tìm kiếm tài liệu/ đề thi 8](#_Toc84787035)

[2.1.11 Trang chủ 9](#_Toc84787036)

[2.1.12 Gửi Email chúc mừng sinh nhật 9](#_Toc84787037)

[2.1.13 Gửi Email hàng loạt 10](#_Toc84787038)

[2.1.14 Xem một tài liệu 10](#_Toc84787039)

# Tổng quan

## Công nghệ

### Front end

* ReactJs
* Material UI

### Back end

* NodeJs
* MongoDb
* ExpressJs

## Vai trò

* Leader backend: Hoàng Duy Khánh
* Leader frontend: Lê Văn Nam

# Chức năng

## Người dùng

### Đăng nhập

#### Mô tả

Cho phép người dùng là thành viên/ quản trị viên đăng nhập.

#### Giao diện

Trang đăng nhập gồm:

* Input mã sinh viên
* Input mật khẩu
* Recaptcha v2
* Nút đăng nhập

#### Giao diện API

* Thành công:
* {
* "msg": "Đăng nhập thành công"
* }
* Lỗi:
* {
* "msg": "Đăng nhập thất bại"
* }

#### Luồng hoạt động

Khi người dùng ấn nút đăng nhập thì:

* Hệ thống kiểm tra xem captcha có chính xác không
* Kiểm tra thông tin tài khoản có tồn tại trong bảng **users** không.

### Quên mật khẩu

#### Mô tả

Cho phép người dùng tạo liên kết lấy lại mật khẩu tài khoản.

#### Giao diện

* Input nhập mã sinh viên
* Recaptcha v2
* Nút gửi

#### Giao diện API

* Thành công:
* {
* "msg": "Liên kết đặt lại mật khẩu đã được gửi đến email của bạn"
* }
* Thất bại:
* {
* "msg": "Mã sinh viên không tồn tại"
* }

#### Luồng hoạt động

* Hệ thống kiểm tra xem mã sinh viên có tồn tại trong cơ sở dữ liệu hay không.
* Tạo một bản ghi mới với user\_id và secret\_key trong bảng reset\_password.
* Gửi email chứa đường dẫn đến trang khôi phục mật khẩu: <https://hithaui.com/reset-password/secret_key>.

### Khôi phục mật khẩu

#### Mô tả

Cho phép người dùng đặt lại mật khẩu tài khoản

#### Giao diện

* Input nhập mật khẩu mới
* Nút gửi

#### Giao diện API

* Thành công:
* {
* "msg": "Khôi phục mật khẩu thành công"
* }
* Thất bại:
* {
* "msg": "Mã không hợp lệ"
* }

#### Luồng hoạt động

* Client gửi một request ban đầu lên server kiểm tra xem secret key có tồn tại hay không.
* Người dùng nhập mật khẩu mới và gửi lên.
* Hệ thống lấy ra người dùng có id ứng với secret key từ bảng reset\_password.
* Cập nhật mật khẩu người dùng và lưu lại.

### Trang cập nhật thông tin cá nhân

#### Mô tả

Cho phép người dùng cập nhật các thông tin cá nhân cơ bản, kỹ năng.

#### Giao diện

* Gồm 1 form cập nhật các thông tin cơ bản với textbox.
* Form thêm và xóa skill (là danh sách các chip).
* Form thay đổi mật khẩu

#### Giao diện API

* Lấy thông tin profile:

  {

    "full\_name": "Nguyễn Sỹ Tùng",

    "phone": "0967545300",

    ...

  }

* Thêm skill:
* {
* "msg": "Thêm thành công"
* }
* Xóa skill:
* {
* "msg": "Xóa thành công"
* }
* Đổi mật khẩu:
* {
* "msg": "Đổi mật khẩu thành công"
* }
* Cập nhật profile:
* {
* "full\_name": "Nguyễn Sỹ Tùng",
* "phone": "0967545300",
* ...
* }
* Lỗi:
* {
* "msg": "Lỗi ..."
* }

#### Luồng hoạt động

* Thêm/ xóa skill: thêm và xóa skill trong trường skills của user
* Đổi mật khẩu: So sánh mật khẩu cũ với mật khẩu hiện tại, nếu khớp thì cập nhật mật khẩu mới.
* Cập nhật profile: Cập nhật các trường trong profile với các trường gửi lên.

### Trang Profile

#### Mô tả

Cho phép hiển thị các thông tin cá nhân như avatar, họ tên, địa chỉ, khóa, khoa, lớp, chức vụ, kỹ năng, giải thưởng …

#### Giao diện

### Lấy danh sách môn học

#### Mô tả

Cho phép người dùng tìm kiếm danh sách môn học.

#### Giao diện API

  [

    {

      "\_id": "12121212121",

      "name": "Lập trình căn bản"

    },

    {

      "\_id": "12121212121",

      "name": "Lập trình hướng đối tượng"

    }

  ]

#### Luồng hoạt động

* Hệ thống tìm trong bảng subjects những môn học có tên giống với trường name và trả về danh sách.

### Upload đề thi

#### Mô tả

Cho phép người dùng upload file đề thi lên hệ thống.

#### Giao diện

* Gồm 1 ô search và select môn
* Input tên
* Input chọn file
* Nút upload

#### Giao diện API

  {

    "msg": "Upload thành công"

  }

#### Luồng hoạt động

* Hệ thống thêm một bản ghi mới vào bảng exams với tên tài liệu, mã môn học, mã người dùng
* Đổi tên file upload trùng với mã bản ghi và lưu vào thư mục uploads/exams

### Xem danh sách/ Xóa đề thi

#### Mô tả

Cho phép người dùng xóa đề thi đã upload

#### Giao diện

* Gồm danh sách đề thi đã upload (Tên, môn, ngày)
* Nút xóa cạnh mỗi đề thi

#### Giao diện API

  [

    {

      "\_id": "12121212121",

      "name": "Đề thi kết thúc học phần",

      "subject\_name": "Quản trị mạng"

    },

    {

      "\_id": "12121212121",

      "name": "Đề kiểm tra lần 1",

      "subject\_name": "Nhập môn công nghệ phần mềm"

    }

  ]

  {

    "msg": "Xóa thành công"

  }

#### Luồng hoạt động

* Hệ thống kiểm tra người dùng có upload đề thi này hay không
* Hệ thống xóa bản ghi trong bảng exams
* Hệ thống xóa file tương ứng trong thư mục uploads/exams

### Upload tài liệu

#### Mô tả

Cho phép người dùng upload các tài liệu/ bài tập lớn/ báo cáo các môn.

#### Giao diện

* Gồm 1 ô seach và select tên môn
* Input tên tài liệu
* Input chọn file
* Nút upload

#### Giao diện API

  {

    "msg": "Upload thành công"

  }

#### Luồng hoạt động

* Hệ thống thêm một bản ghi mới vào bảng documents với tên tài liệu, mã môn học, mã người dùng
* Đổi tên file upload trùng với mã bản ghi và lưu vào thư mục uploads/documents

### Trang tìm kiếm tài liệu/ đề thi

#### Mô tả

Cho phép người dùng tìm kiếm tài liệu/ đề thi

#### Giao diện

* Ô search và select tên môn
* Input nhập tên
* Select chọn đề thi, bài tập lớn/ đồ án/ báo cáo
* Nút tìm kiếm
* Danh sách kết quả

#### Giao diện API

  [

    {

      "\_id": "12121212121",

      "name":"Đề thi kết thúc học phần",

      "subject\_name": "Nhập môn công nghệ phần mềm",

      "uploaded\_by": "Ngô Bá Khá"

    },

    {

      "\_id": "12121212121",

      "name":"Đề thi kết thúc học phần",

      "subject\_name": "Quản trị mạng",

      "uploaded\_by": "Ngô Bá Khá"

    }

    {

      "\_id": "12121212121",

      "name":"Phần mềm quản lý nhà hàng",

      "subject\_name": "Lập trình Windows",

      "uploaded\_by": "Bùi Xuân Huấn"

    }

  ]

#### Luồng hoạt động

* Tùy theo lựa chọn của người dùng mà chọn api để tìm kiếm (đề thi hoặc tài liệu)
* Hệ thống kiếm tra trong bảng exams/ documents tài liệu có tên giống với tên người dùng tìm kiếm (regex) và mã môn tương ứng.

### Trang chủ

* Convert lại thành Material UI

### Gửi Email chúc mừng sinh nhật

#### Mô tả

Gửi email chúc mừng sinh nhật theo template có sẵn đến thành viên.

#### Luồng hoạt động

* Cronjob chạy vào lúc 08:00 mỗi ngày tìm thành viên có ngày sinh nhật là hôm nay rồi gửi email theo template có sẵn

### Gửi Email hàng loạt

#### Mô tả

Gửi email hàng loạt theo template upload, dữ liệu được lấy từ file excel.

##### Ví dụ file excel

|  |  |  |  |
| --- | --- | --- | --- |
| username | name | email | Password |
| tung1 | Tung | [tungaqhd@gmail.com](mailto:tungaqhd@gmail.com) | 123456 |
| tung2 | Tung | [tungaqhd@gmail.com](mailto:tungaqhd@gmail.com) | 123456 |
| tung3 | Tung | [tungaqhd@gmail.com](mailto:tungaqhd@gmail.com) | 123456 |

#### <p>Xin chào #username, mật khẩu của bạn là #password</p>

#### Luồng hoạt động

* Dòng đầu tiên sẽ là placeholder tương ứng trong template
* Mỗi dòng sẽ tương ứng với 1 email.
* Khớp cột username với #username, name với #name, …. Trong template

### Xem một tài liệu

#### Mô tả

Cho phép người dùng xem một tài liệu.

#### Giao diện

Hiển thị các thông tin:

* Tên
* Loại
* Lượt xem
* Tên môn học
* Người upload
* Ngày tải lên
* Nút tải xuống

#### Giao diện API

{

  "name": "Phần mềm quản lý nhà hàng",

  "type": "exam",

  "exam\_code": "String",

  "year": 2020,

  "view": 1,

  "status": "active",

  "user": {

    "full\_name": "Nguyễn Sỹ Tùng",

    "club\_year": 13

  },

  "subject": {

    "name": "Lập trình Windows"

  }

}

#### Luồng hoạt động

* Người dùng gửi lên id của tài liệu
* Hệ thống trả về thông tin tài liệu như giao diện API

### Tìm kiếm thành viên

#### Mô tả

Cho phép người dùng, quản trị viên tìm kiếm người dùng.

#### Giao diện

* Input nhập tên
* Input nhập khóa
* Nút tìm kiếm

### Trang chủ

#### Mô tả

Xào lại nội dung từ trang hithaui.com tùy người làm (theo template đã gửi)

### Xem danh sách người dùng người dùng

#### Mô tả

Cho phép xem bảng người dùng, có thể thêm ở ngay trang tìm kiếm thành viên đã có

#### Giao diện

* Bảng hiển thị các thông tin cơ bản như:
  + Mã sinh viên
  + Họ và tên
  + Khóa
  + Ngày tháng năm sinh
  + Phân trang

#### Giao diện API

{

    "total": 200,

    "users": [

        {

            "\_id": "12345",

            "fullName": "Nguyễn Sỹ Tùng",

            ...

        }

    ]

}

### Sửa người dùng

#### Mô tả

Cho phép sửa người dùng bằng cách nhấn vào nút sửa người dùng trong danh sách người dùng, một modal sẽ hiện ra để người quản trị nhập thông tin

#### Giao diện

* Modal hiển thị form nhập thông tin
* Nút cập nhật
* Nút đóng

### Xóa người dùng

#### Mô tả

Cho phép QTV xóa người dùng. Phải có confirm box khi nhấn xóa (“Bạn có chắc muốn xóa người dùng này không”)

### Xem danh sách tài liệu

#### Mô tả

Cho phép quản trị viên xem danh sách tài liệu, đồng thời có thể chọn duyệt tài liệu hoặc xóa tài liệu.

#### Giao diện

* Table hiển thị danh sách các tài liệu theo thời gian
* Nút duyệt tài liệu
* Nút xóa tài liệu