TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

🙠----🕮----🙢

Ảnh có chứa văn bản

Mô tả được tạo tự động

**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**TÊN ĐỀ TÀI:**

**XÂY DỰNG GIAO DIỆN WEBSITE**

**QUẢN LÝ HỌC SINH**

**Đơn vị thực tập: Công ty BYS**

**Vị trí thực tập: Frontend**

**Học sinh thực hiện:** Trương Công Minh

**Lớp:** 44K14

**Giảng viên hướng dẫn:** Cao Thị Nhâm

**Đà Nẵng, 7/2021**

**LỜI CẢM ƠN**

**LỜI CAM ĐOAN**

Em xin cam đoan đề tài: “Xây dựng website quản lý học sinh” là một dự án được thực hiện bởi nỗ lực của cá nhân em dưới sự hướng dẫn tận tình của giáo viên hướng dẫn và từ phía công ty mà em đang thực tập. Ngoài ra không có bất cứ sao chép nào của người khác.

Các số liệu và kết quả thực hành trong đề tài là trung thực và hoàn toàn không sao chép hay sử dụng kết quả kết quả nghiên cứu đề tài của người khác. Nếu có phát hiện điều gì không đúng, em xin chịu hoàn toàn trách nhiệm.

# MỤC LỤC

[MỤC LỤC 2](#_Toc80367824)

[LỜI MỞ ĐẦU 1](#_Toc80367825)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc80367826)

[1.1. Tổng quan về Angular framework 2](#_Toc80367827)

[1.1.1. Giới thiệu 2](#_Toc80367828)

[1.1.2. Tại sao nên dùng Angular 2](#_Toc80367829)

[1.2. Tổng quan về TypeScript 3](#_Toc80367830)

[1.3. Tổng quan về REST API và Json Server 4](#_Toc80367831)

[1.4. Cơ hội nghề nghiệp 5](#_Toc80367832)

[1.4.1. Mô tả vị trí việc làm 5](#_Toc80367833)

[1.4.2. Các kỹ năng cần có 5](#_Toc80367834)

[1.4.3. Mức lương 6](#_Toc80367835)

[CHƯƠNG 2. phân tích và thiết kế ứng dụng 7](#_Toc80367836)

[2.1. Mô tả bài toán: 7](#_Toc80367837)

[2.2. Thiết kế các chức năng 7](#_Toc80367838)

[2.3. Thiết kế cơ sở dữ liệu 9](#_Toc80367839)

[CHƯƠNG 3. PHÁT TRIỂN ỨNG DỤNG 10](#_Toc80367840)

[3.1. Sơ đồ chức năng 10](#_Toc80367841)

[3.2. Thông tin chi tiết các form làm việc 13](#_Toc80367842)

[3.2.1. Màn hình đăng nhập 13](#_Toc80367843)

[3.2.2. Màn hình trang chủ 14](#_Toc80367844)

[3.2.3. Form thay đổi mật khẩu 17](#_Toc80367845)

[CHƯƠNG 4. KẾT QUẢ 18](#_Toc80367846)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 19](#_Toc80367847)

[TÀI LIỆU THAM KHẢO 20](#_Toc80367848)

DANH MỤC HÌNH ẢNH

[*Hình 1‑1 Angular Framework* 2](#_Toc80368642)

[*Hình 1‑2 REST API* 4](#_Toc80368643)

[*Hình 1‑3 Postman* 5](#_Toc80368644)

[*Hình 2‑1 Use case* 8](#_Toc80368645)

[*Hình 2‑2 Thiết kế cơ sở dữ liệu* 9](#_Toc80368646)

[*Hình 3‑1 Sơ đồ chức năng* 10](#_Toc80368647)

[*Hình 3‑2 Lưu đồ đăng nhập* 10](#_Toc80368648)

[*Hình 3‑3 Lưu đồ thêm học sinh* 11](#_Toc80368649)

[*Hình 3‑4 Lưu đồ xóa học sinh* 11](#_Toc80368650)

[*Hình 3‑5 Lưu đồ sửa học sinh* 12](#_Toc80368651)

[*Hình 3‑6 Lưu đồ đổi mật khẩu* 12](#_Toc80368652)

[*Hình 3‑7 Màn hình đăng nhập* 13](#_Toc80368653)

[*Hình 3‑8: Màn hình trang chủ* 14](#_Toc80368654)

[*Hình 3‑9 Phần header* 14](#_Toc80368655)

[*Hình 3‑10 Form thêm học sinh* 15](#_Toc80368656)

[*Hình 3‑11 Tìm kiếm học sinh* 15](#_Toc80368657)

[*Hình 3‑12 Xóa học sinh* 16](#_Toc80368658)

[*Hình 3‑13 Sửa học sinh* 16](#_Toc80368659)

[*Hình 3‑14 Màn hình trang chủ* 17](#_Toc80368660)

[*Hình 3‑15 Màn hình thay đổi mật khẩu* 17](#_Toc80368661)

DANH MỤC BẢNG BIỂU

[*Bảng 2‑1 Mô tả chức năng* 7](#_Toc80192495)

[*Bảng 2‑2 Mô hình dữ liệu* 7](#_Toc80192496)

# LỜI MỞ ĐẦU

1. **Mục tiêu của đề tài**

* Nghiên cứu ngôn ngữ lập trình TypeScript
* Nghiên cứu framework xây dựng giao diện website Angular 12
* Nghiên cứu REST API và Json Server
* Xây dựng ứng dụng web Quản lý học sinh

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ngôn ngữ lập trình TypeScript
* Nghiên cứu framework xây dựng giao diện website Angular 12
* Nghiên cứu REST API và Json Server
* Phân tích bài toán quản lý học sinh
* Phát triển ứng dụng để quản lý học sinh

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm 4 chương nội dung và phần kết luận.

* **Chương 1**: Cơ sở lý thuyết
* **Chương 2**: Phân tích và thiết kế ứng dụng
* **Chương 3**: Phát triển ứng dụng
* **Chương 4**: Kết quả
* Kết luận và hướng phát triển

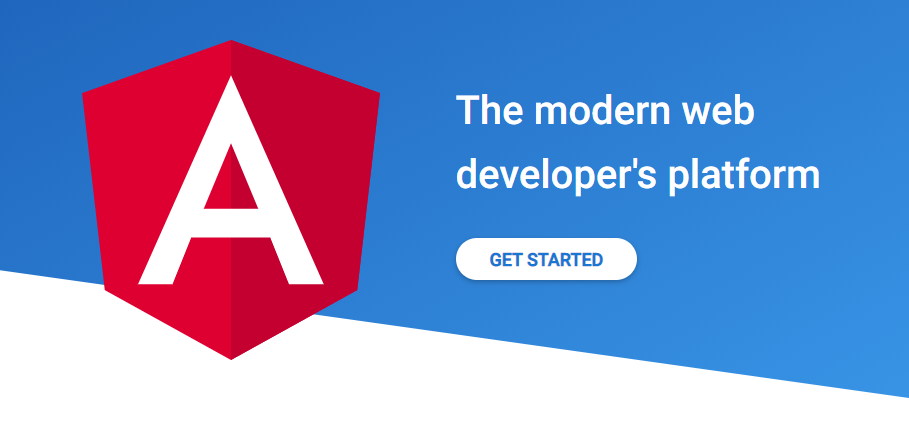
# CƠ SỞ LÝ THUYẾT

## Tổng quan về Angular framework

### Giới thiệu

Angular là một JavaScript framework dùng để viết giao diện web (Front-end), được tạo ra để xây dựng các ứng dụng web động (dynamic web app), nó thường được sử dụng để tạo ra các ứng dụng một trang (Single Page Application - SPA). Và hoạt động dựa trên việc mở rộng thêm các thuộc tính (attribute) cho các thẻ của HTML (Các thuộc tính theo quy tắc của Angular).

Angular là một nền tảng miễn phí và được nhiều lập trình viên trên thế giới ưa chuộng. Angular là một khung làm việc của JavaScript MVC phía máy khách (client) nhằm phát triển ứng dụng web động.



*Hình 1‑1 Angular Framework*

### Tại sao nên dùng Angular

Angular – ứng dụng là một tập hợp những component (thành phần) và một vài services (dịch vụ) cung cấp chức năng trên những component đó.

* Ứng dụng được giảm tối đa kích thước và tăng tối đa hiệu suất.
* HTML linh hoạt hơn.
* Code HTML mạnh mẽ hơn với những đặc trưng như IF , FOR, LOCAL VARIABLES, …
* Dễ dàng hiển thị các field từ data model của website và theo dõi những thay đổi, cập nhật lại từ người dùng nhờ binding data.
* Xây dựng và tái sử dụng nội dung nhờ vào những khối module độc lập.
* Nhanh chóng giải quyết các bài toán logic nhờ back- end service hỗ trợ giao tiếp.

## Tổng quan về TypeScript

TypeScript là một phiên bản cao hơn của JavaScript, được thiết kế để xây dựng các ứng dụng lớn và phức tạp. Nó kế thừa nhiều khái niệm từ Java và C#, TypeScript là ngôn ngữ tĩnh (Static typed) có nghĩa là nó nghiêm ngặt và có trật tự trái ngược với free-type. Nó còn được bổ sung thêm lớp hướng đối tượng mà điều này không có ở Javascript.

*Ứu điểm của TypeScript:*

* Dễ dàng hơn trong phát triển các dự án lớn, được hỗ trợ bởi các Javascript Framework lớn.
* Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override…v.v
* Cách tổ chức code rõ ràng hơn, hỗ trợ cơ chế giúp kiến trúc hệ thống code hướng module, hỗ trợ namespace, giúp xây dựng các hệ thống lớn nơi mà nhiều lập trình viên có thể làm việc cùng nhau một cách dễ dàng hơn.
* Hỗ trợ các tính năng mới nhất của Javascript. TypeScript luôn đảm bảo việc sử dụng đầy đủ các kỹ thuật mới nhất của Javascript, ví dụ như version hiện tại là ECMAScript 2015 (ES6).
* Một lợi thế của Typescript nữa là mã nguồn mở vì vậy nó miễn phí và có cộng đồng hỗ trợ rất lớn.
* Với static typing, code viết bằng TypeScript dễ dự đoán hơn, và dễ debug hơn.

## Tổng quan về REST API và Json Server

1. ***REST API***

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.



*Hình 1‑2 REST API*

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau. Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

1. ***Json Server***

Định nghĩa cơ bản JSON Server là gì như sau: JSON là sẽ sử dụng các cặp key-value để dữ liệu có thể sử dụng. Khi đó, nó sẽ hỗ trợ cho những cấu trúc dữ liệu dạng đối tượng và mảng. Các nhà phát triển thường xuyên sử dụng JSON để có thể làm việc với AJAX, cách định dạng này sẽ phối hợp hiệu quả với nhau để có thể nén dữ liệu tải không đồng bộ. Có nghĩa như sau: trang web có thể được cập nhật thông tin mà bạn không cần phải làm mới trang.

1. ***Sử dụng Postman để làm việc với API***

Postman là một App Extensions, cho phép làm việc với các API, nhất là REST, giúp ích rất nhiều cho việc testing. Hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, DELETE, OPTIONS, HEAD ...) Postman cho phép lưu lại các lần sử dụng. Sử dụng cho cá nhân hoặc team lớn.



*Hình 1‑3 Postman*

## Cơ hội nghề nghiệp

### Mô tả vị trí việc làm

Lập trình viên Front-end là người tập trung phát triển phía Client Side, nói một cách đơn giản dễ hiểu là tập trung vào mảng phát triển xây dựng giao diện và trải nghiệm cho người dùng, là người phụ trách phát triển hiển thị và trải nghiệm người dùng cho ứng dụng web. Front-end Developer chính là người quyết định cái nhìn đầu tiên của người dùng về trang web, đồng thời mang lại một trang web dễ dàng thao tác và sử dụng.

### Các kỹ năng cần có

* Thành thạo HTML, CSS, Boostrap, JQuery… và ngôn ngữ lập trình JavaScript
* Kiến thức về các framework của Javascript
* Kỹ năng về các Frontend frameworks
* Kỹ năng giải quyết và xử lý vấn đề
* Nắm rõ toàn bộ quá trình phát triển web (thiết kế, phát triển và thực thi)
* Có kiến thức về các quy tắc trong SEO
* Có kinh nghiệm sử dụng Photoshop (Hoặc Sketch)
* Có kiến thức cơ bản về UX/UI
* Có kiến thức về Responsive Design
* Khả năng làm việc tốt trong môi trường tốc độ cao
* Kinh nghiệm với CSS Preprocessors

### Mức lương

Như các công việc khác, mỗi mức trình độ kinh nghiệm của lập trình viên sẽ có mức thu nhập khác nhau. Tại nước ta, hiện nay vị trí frontend developer có mức lương trung bình dao động từ 13 - 20 triệu/ tháng. Còn mức lương hiện nay cho vị trí fresher frontend trung bình dao động từ 6 – 8 triệu/tháng.

# phân tích và thiết kế ứng dụng

## Mô tả bài toán:

**Khái quát**: Website giúp quản lý học sinh một cách đơn giản, chính xác và tiết kiệm thời gian cho giáo viên. Có thể truy cập bất cứ đâu và trên mọi thiết bị có kết nối internet.

**Mô tả bài toán**: Giáo viên được cung cấp tài khoản để truy cập và đăng nhập vào website để quản lý học sinh. Có thể tìm kiếm học sinh theo từ khóa và nếu có một học sinh mới thì giáo viên có thể thêm học sinh vào hệ thống. Khi một học sinh đó không còn học tập thì có thể xóa học sinh đó khỏi hệ thống. Giáo viên có thể sửa đổi thông tin học sinh nếu có thay đổi hoặc sai sót lúc tạo mới. Ngoài ra giáo viên có thể thay đổi mật khẩu đăng nhập và đăng xuất khỏi hệ thống sau khi sử dụng các chức năng của website.

## Thiết kế các chức năng

***Phạm vi:***

Xây dựng giao diện website quản lý học sinh bằng Angular Framework với các chức năng như xem thông tin, tìm kiếm, thêm, xóa, cập nhật học sinh, đăng nhập đăng xuất cũng như thay đổi mật khẩu đăng nhập.

***Mô tả chức năng***

|  |  |  |
| --- | --- | --- |
| Actor | Chức năng | Mô tả |
| Admin | Quản lý học sinh | Có thể tra cứu, thêm mới, cập nhật và xóa học sinh |
| Admin | Thay đổi mật khẩu | Có thể thay đổi mật khẩu đăng nhập |
| Admin | Đăng nhập, đăng xuất | Có thể đăng nhập để thực hiện quản lý học sinh, thay đổi mật khẩu và đăng xuất |

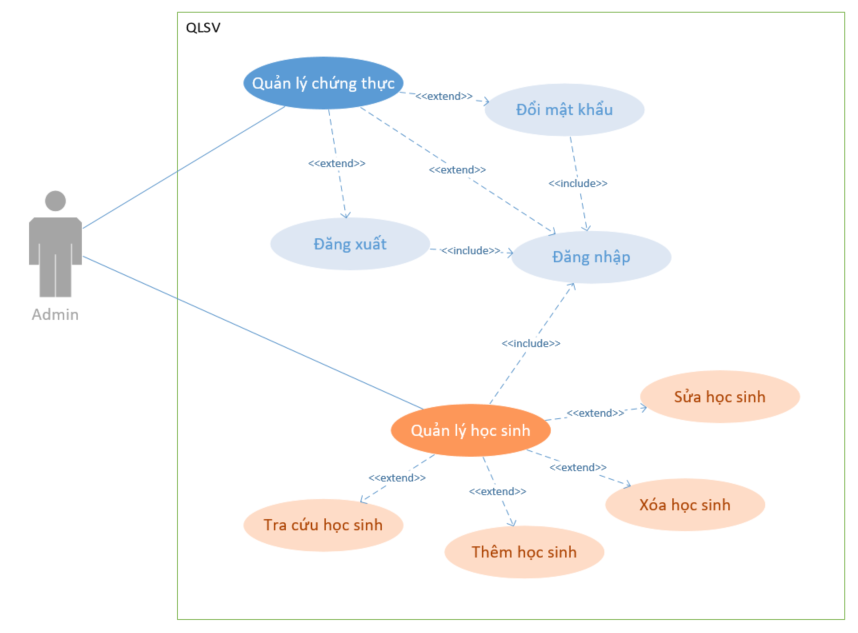
*Bảng 2‑1 Mô tả chức năng*

***Mô hình dữ liệu***

|  |  |
| --- | --- |
| **Thực thể** | **Thuộc tính** |
| Student | ***ID,*** Name, Address, PhoneNumber, Email, DateSumited |
| User | ***UserName***, password, name |

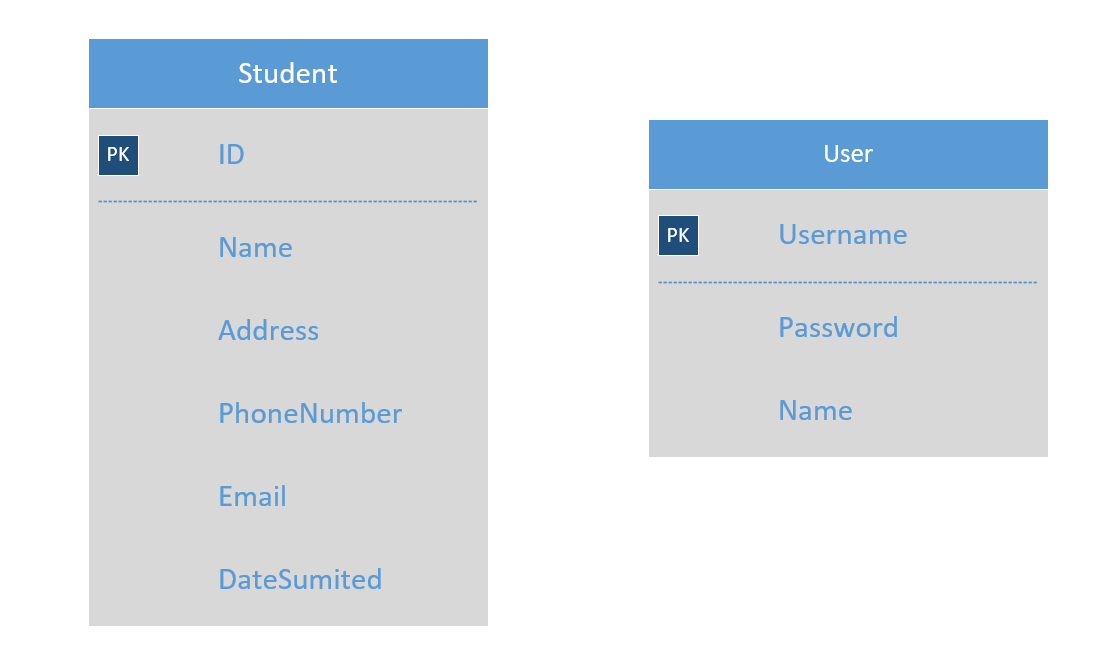
*Bảng 2‑2 Mô hình dữ liệu*

***Use Case***



*Hình 2‑1 Use case*

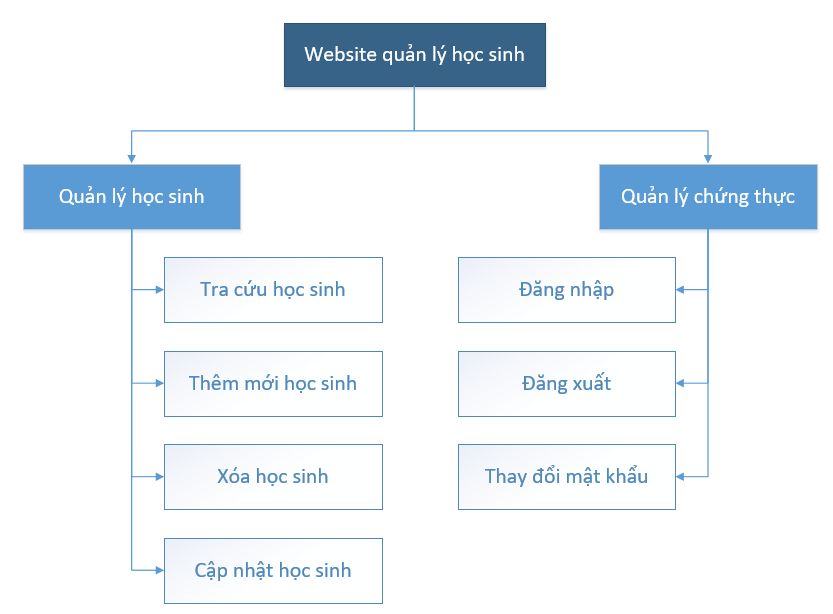
## Thiết kế cơ sở dữ liệu



*Hình 2‑2 Thiết kế cơ sở dữ liệu*

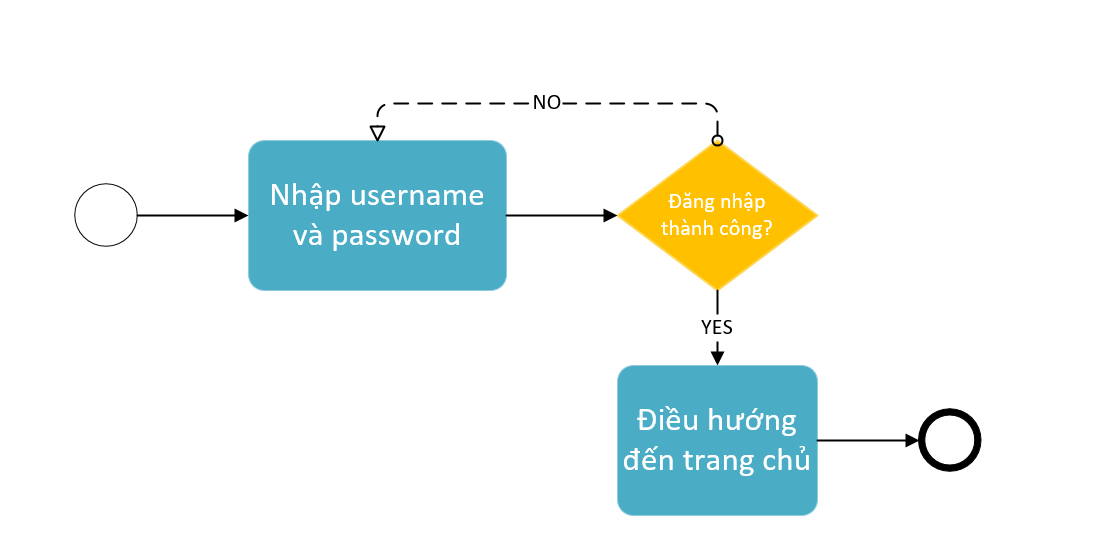
# PHÁT TRIỂN ỨNG DỤNG

## Sơ đồ chức năng



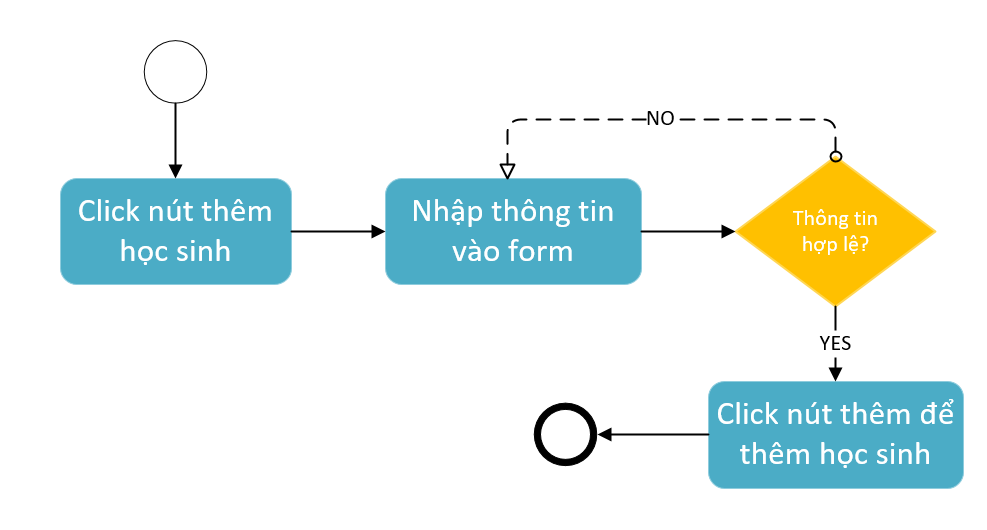
*Hình 3‑1 Sơ đồ chức năng*

Lưu đồ đăng nhập



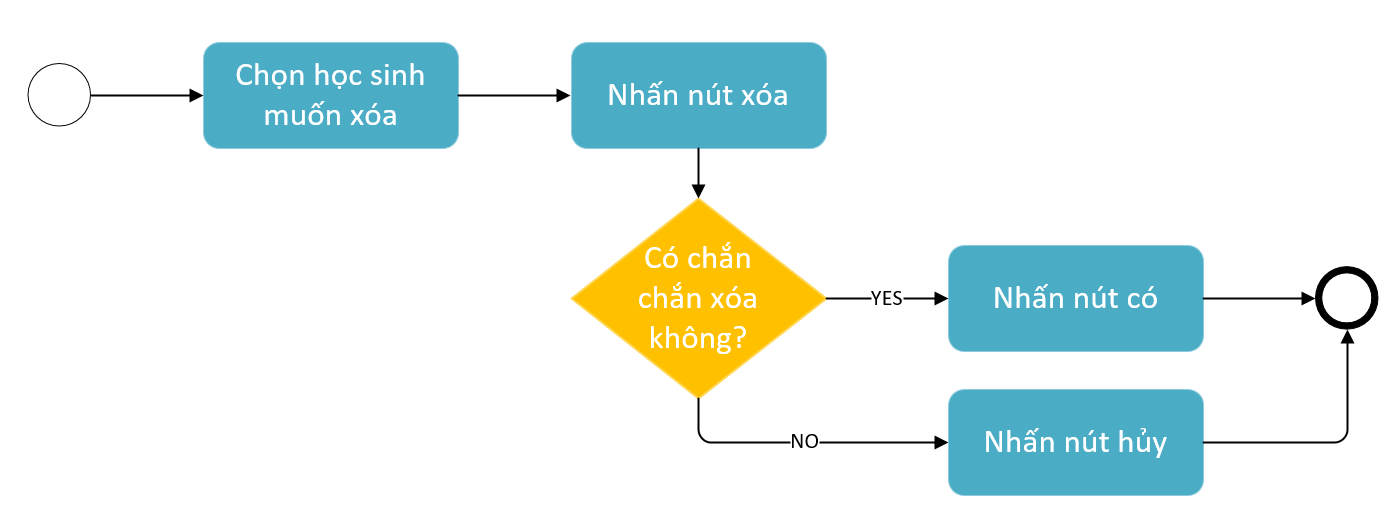
*Hình 3‑2 Lưu đồ đăng nhập*

Lưu đồ thêm học sinh



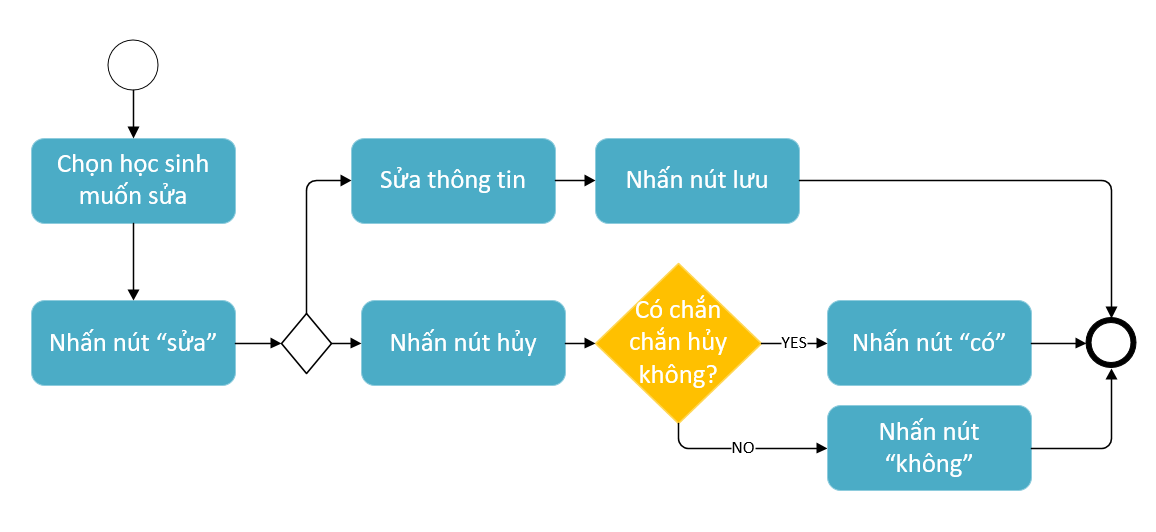
*Hình 3‑3 Lưu đồ thêm học sinh*

Lưu đồ xóa học sinh



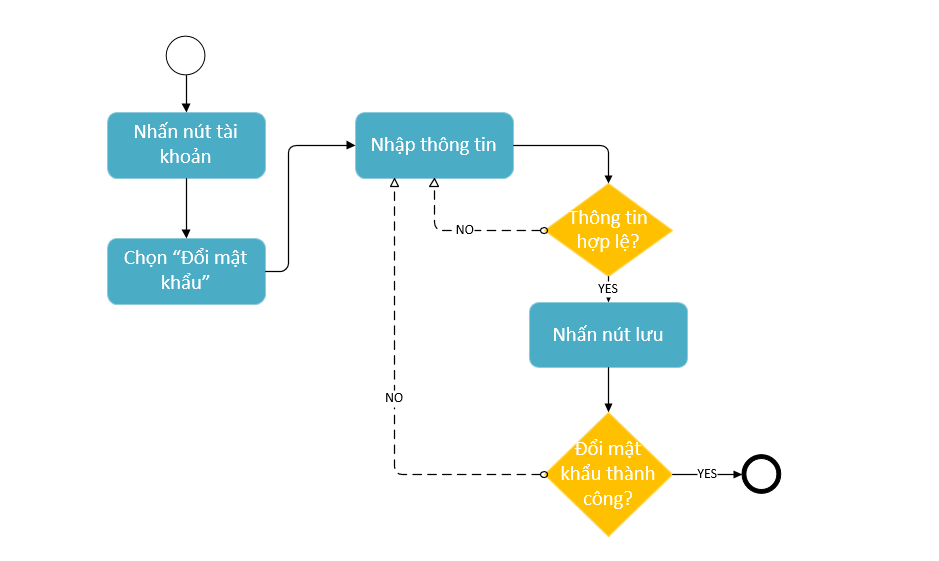
*Hình 3‑4 Lưu đồ xóa học sinh*

Lưu đồ sửa học sinh



*Hình 3‑5 Lưu đồ sửa học sinh*

Lưu đồ đổi mật khẩu

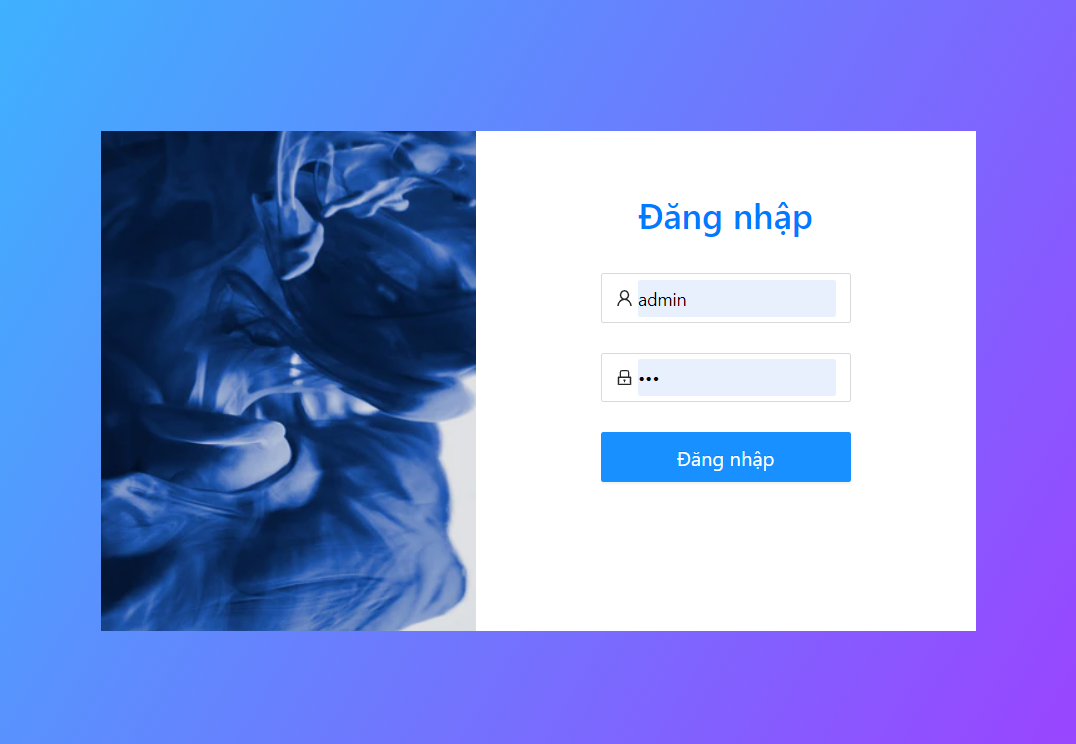


*Hình 3‑6 Lưu đồ đổi mật khẩu*

## Thông tin chi tiết các form làm việc

### Màn hình đăng nhập

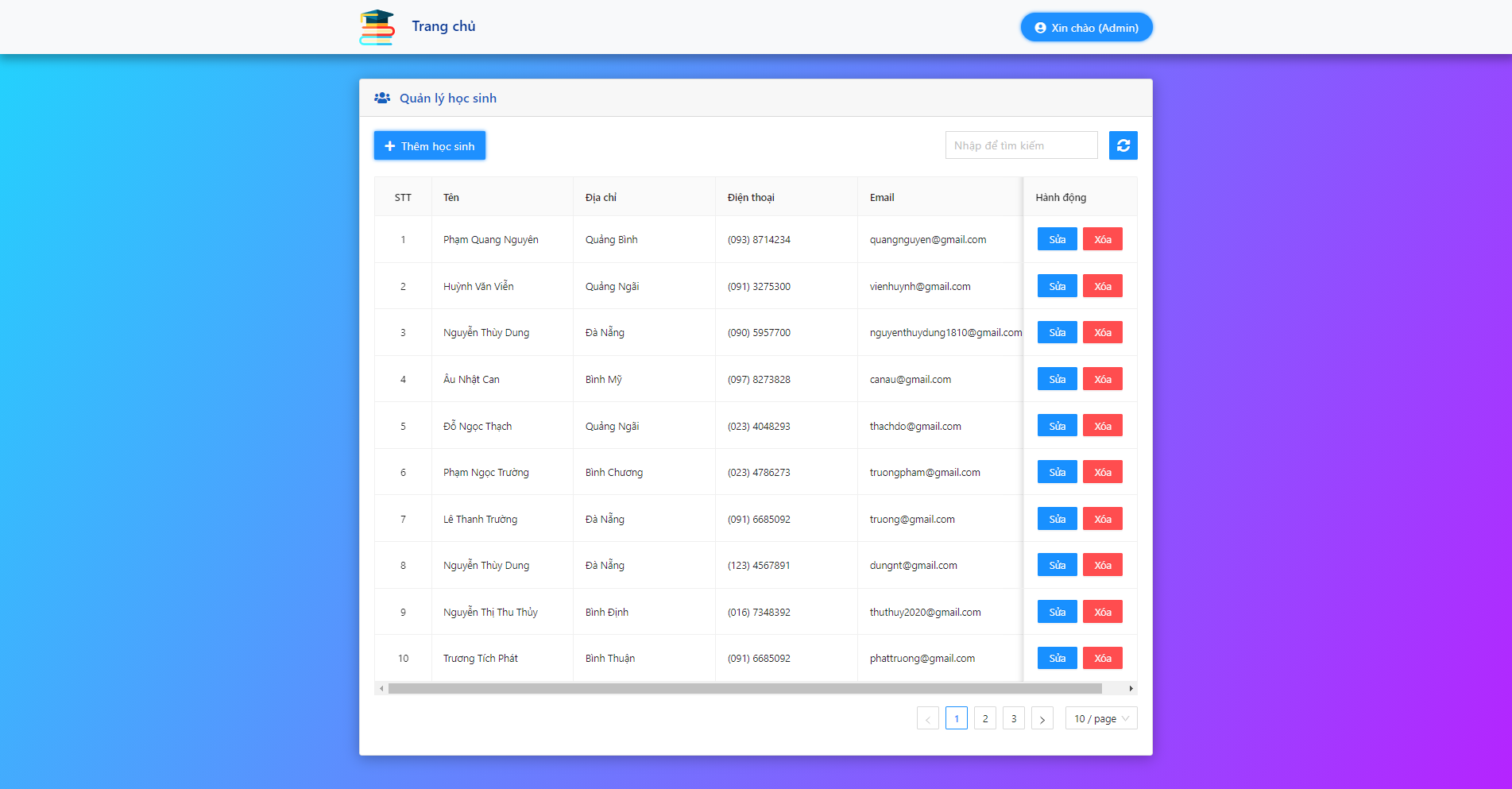
Theo yêu cầu của ứng dụng quản lý và đảm bảo tính bảo mật nên những người quản lý học sinh được cấp tài khoản mới vào được trang web. Nếu bạn không phải là người quản lý hay không được cấp tài khoản thì sẽ không đăng nhập được vào trang web. Khi nhập sai tên tài khoản hoặc mật khẩu thì sẽ thông báo cho người đăng nhập. Nếu đúng tên tài khoản và mật khẩu thì người quản lý được phép truy cập.



*Hình 3‑7 Màn hình đăng nhập*

Sau khi đăng nhập thành công thì sẽ được điều hướng vào trang chủ đồng thời cũng là trang Dashboard quản lý học sinh.

### Màn hình trang chủ

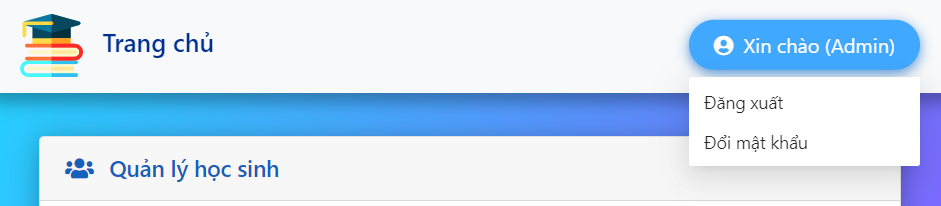


*Hình 3‑8: Màn hình trang chủ*

Đây là giao diện chính đồng thời cũng là giao diện quản lý học sinh. Giúp người quản lý thực hiện các chức năng chính của trang web.

*Bao gồm:*

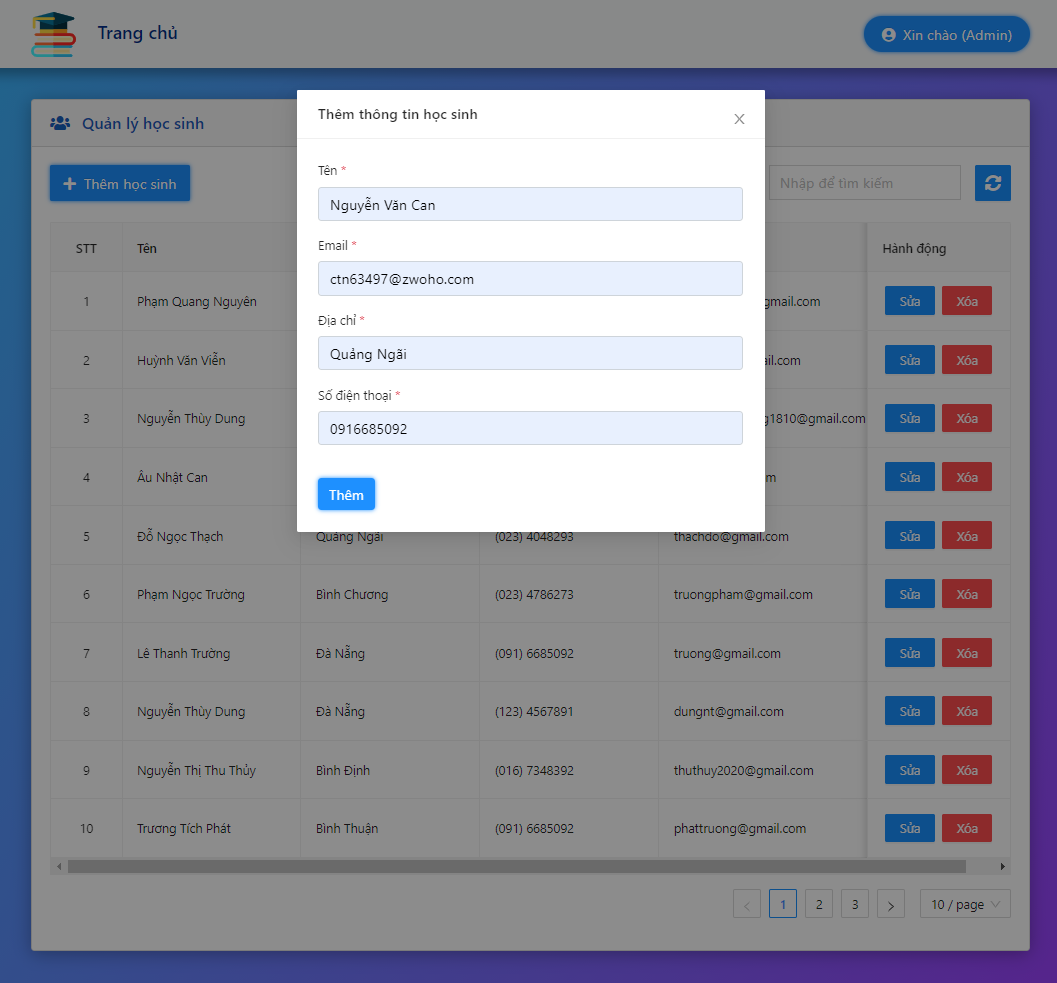
* Phần header bao gồm logo và nút quản lý chứng thực



*Hình 3‑9 Phần header*

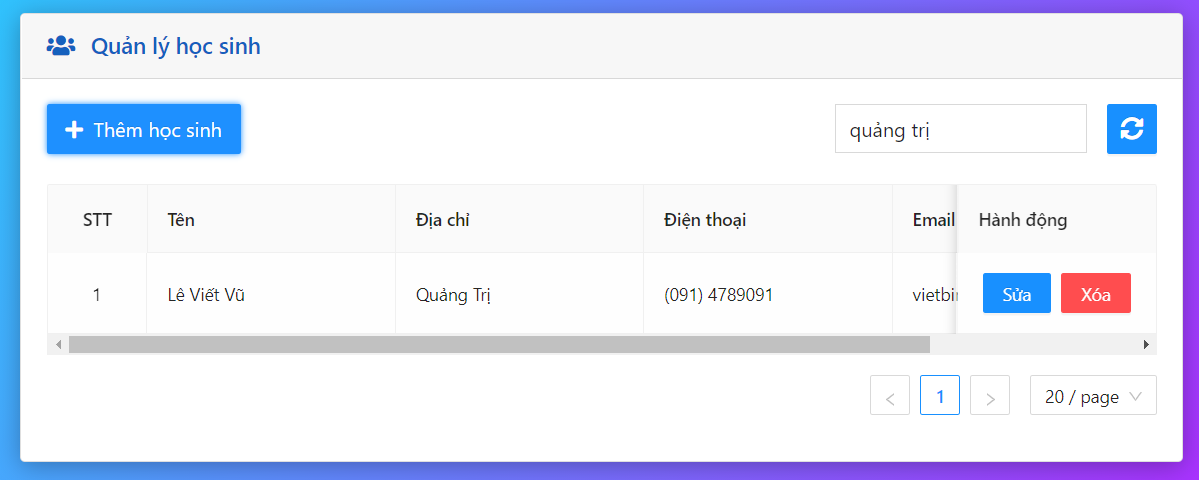
* Phần body là một bảng dữ liệu các học sinh bao gồm các trường như tên, email, địa chỉ, số điện thoại, quê quán. Ở phần thân của trang web có chức năng như tạo mới, tìm kiếm, chỉnh sửa, xóa, reload dữ liệu và phân trang.

Tạo mới: Form tạo mới cho phép người dùng thêm thông tin của một học sinh. Khi muốn thêm học sinh, người quản lý nhấn nút thêm học sinh, đồng thời form thêm thông tin học sinh sẽ hiện lên. Người dùng điền đầy đủ thông tin và nhấn nút thêm để thêm thông tin học sinh vào cơ sở dữ liệu. Sau đó dữ liệu vừa được thêm sẽ hiển thị ở vị trí đầu tiên trong bảng dữ liệu.



*Hình 3‑10 Form thêm học sinh*

Tìm kiếm: chức năng này cho phép người dùng nhập các từ khóa tìm kiếm vào ô tìm kiếm và khi dừng gõ từ khóa thì sau 0.5s các dữ liệu tìm kiếm theo từ khóa vừa nhập được hiển thị hết phía ngay bên dưới bảng dữ liệu



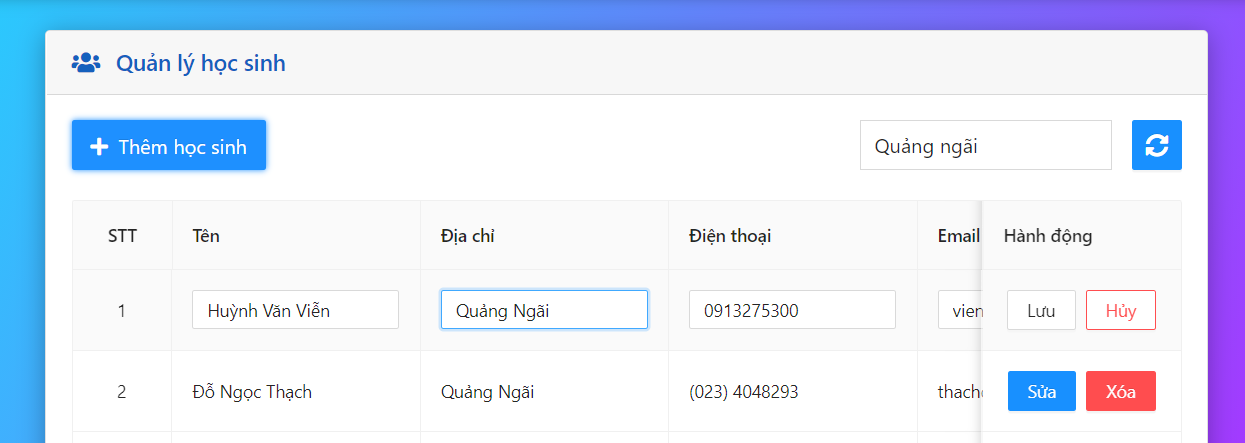
*Hình 3‑11 Tìm kiếm học sinh*

Xóa: Khi người quản lý muốn xóa thông tin của một học sinh thì tìm đến hàng chứa thông tin học sinh muốn xóa và nhấn nút xóa. Ngay lúc này một popup xác nhận xóa hiện lên để xác nhận có chắc chắn xóa hay là không. Khi nhấn hủy thì sẽ hoàn tác lại như ban đầu và nhấn xóa để xóa dữ liệu ngay lập tức.



*Hình 3‑12 Xóa học sinh*

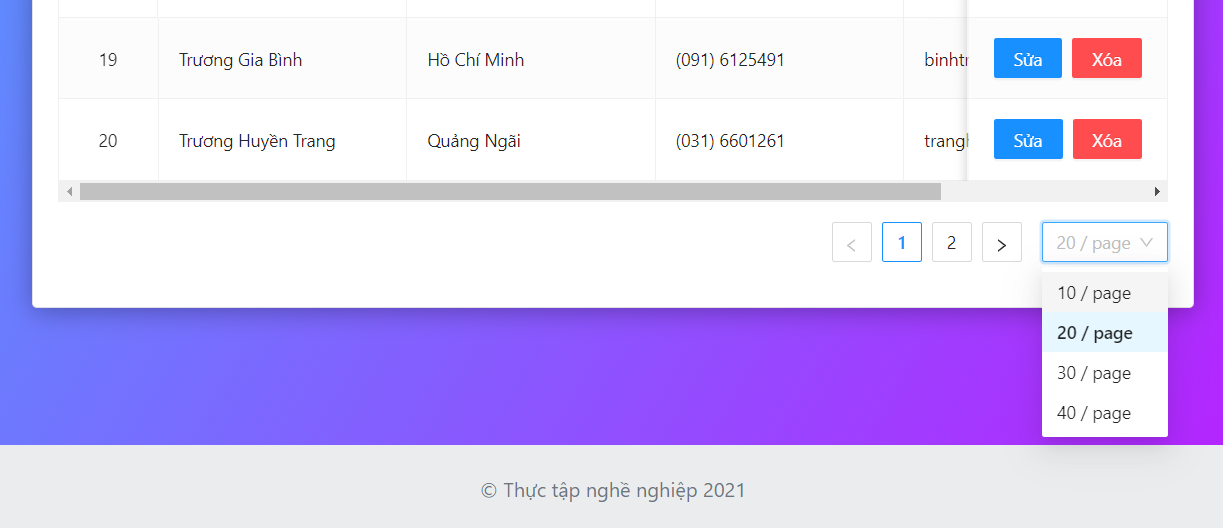
Sửa: Khi có nhu cầu thay đổi thông tin học sinh, người quản lý thực hiện chức năng sửa đổi thông tin bằng cách tìm đến hàng thông tin học sinh muốn sửa, sau đó nhấn nút sửa và tiến hành thay đổi thông tin ngay trong các ô thông tin. Sau khi thay đổi thông tin người quản có thể nhấn nút lưu để lưu dữ liệu cập nhật hoặc nhấn hủy đề hoàn tác dữ liệu, trở lại ban đầu.



*Hình 3‑13 Sửa học sinh*

Reload: chức năng này giúp người dùng load lại toàn bộ dữ liệu trong bảng.

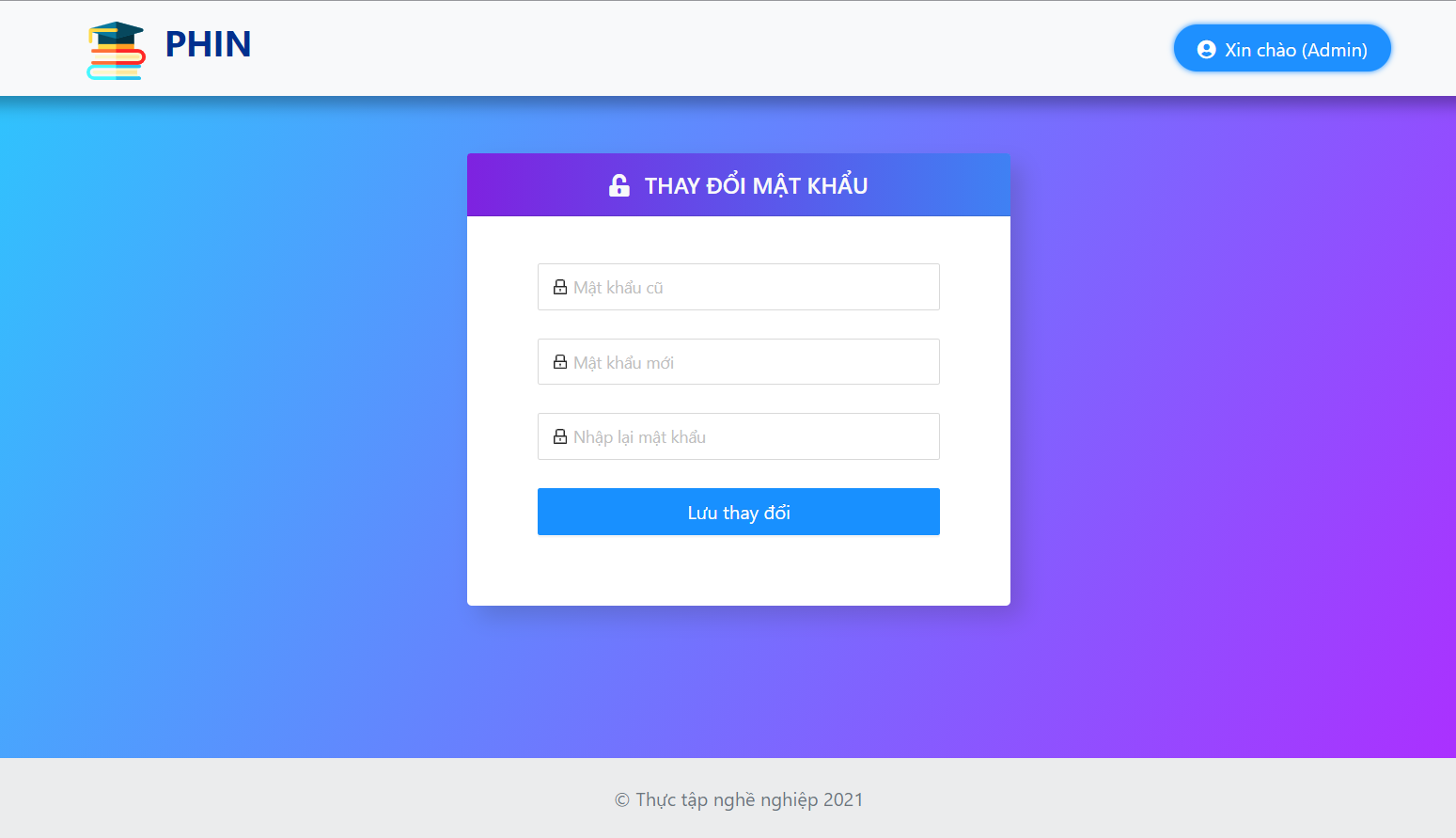
Phân trang: người dùng có thể chọn trang hiển thị dữ liệu. Đồng thời có thể thay đổi số dòng dữ liệu hiển thị trong một trang của bảng.



*Hình 3‑14 Màn hình trang chủ*

### Form thay đổi mật khẩu

Để thay đổi mật khẩu người dùng phải nhập vào mật khẩu hiện tại và mật khẩu mới. Nếu mật khẩu cũ hoặc mật khẩu mới và mật khẩu xác nhận không khớp thì sẽ thông báo lỗi và việc thực hiện thay đổi không thành công và ngược lại



*Hình 3‑15 Màn hình thay đổi mật khẩu*

# KẾT QUẢ

Đề tài xây dựng giao diện website quản lý học sinh này được thực hiện trong kì thực tập nghề nghiệp dưới sự hướng dẫn tận tình của giáo viên hướng dẫn và phía doanh nghiệp. Mặc dù trong thời gian ngắn nhưng em cũng đã học được cách làm việc cơ bản với Angular Framework và tạo ra được một trang web quản lý học sinh nhỏ để vận dụng các kiến thức được học trong quá trình thực tập tại công ty BYS. Trang web được xây dựng với các chức năng cơ bản như đăng nhập, đăng xuất, đổi mật khẩu, thêm – xóa – sửa – tìm kiếm học sinh và được thiết kế đẹp mắt đồng thời người dùng có thể sử dụng trang web một cách dễ dàng.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sau gần 3 tháng thực tập và rèn luyện dưới sự hướng dẫn tận tình của cô giáo viên hướng dẫn và các nhân viên lập trình lâu năm tại công ty BYS. Với đề tài xây dựng giao diện website quản lý học sinh dựa trên kiến thức về Angular Framework, bản thân em đã học được thêm kiến thức, kỹ năng và rút ra nhiều kinh nghiệm. Ứng dụng lượng kiến thức được học tập và tích lũy tại trường vào quá trình thực tập, nhận ra được nhiều thiếu sót của bản thân cả về kiến thức lẫn kỹ năng, từ đó không ngững nỗ lực, cố gắng và trau dồi thêm.

Vì tình hình dịch bệnh diễn biến phức tạp nên thời gian thực tế tại doanh nghiệp là không nhiều và phần lớn thời gian buộc phải thực tập tại nhà, nhưng bản thân em ý thức được khoảng thời gian được học hỏi tại doanh nghiệp là vô cùng quý giá, giúp em cải thiện kiến thức về lập trình, cải thiện kỹ năng mềm của bản thân, và trau dồi thêm kinh nghiệm, trải nghiệm, có thêm những mối quan hệ mới. Khoảng thời gian thực tập tại Công ty Cổ phần BYS, bản thân em đã có cơ hội tiếp xúc trực tiếp với văn hóa công ty, biết thêm nhiều điều mới để có sự chuẩn bị cho thời gian thực tập tốt nghiệp và xa hơn là sau khi ra trường.

Hiện tại trang web chỉ mới dừng lại ở mức cơ bản, để mang đến thêm tiện ích và tối ưu hiệu quả, trong thời gian sắp tới em dự định sẽ học thêm về backend, để phát triển thêm nhiều chức năng mới của web, để có thể tiến bộ hơn nữa trong lĩnh vực mà mình theo đuổi.

Do những hạn chế về mặt thời gian, kinh nghiệm và kỹ năng nên bài báo cáo còn tồn đọng một số hạn chế nhất định. Em rất mong nhận được sự góp ý của thầy cô để có thể hoàn thiện và phát triển bản thân trên con đường trở thành một lập trình viên fullstack. Em xin chân thành cảm ơn.

# TÀI LIỆU THAM KHẢO

1. Series về các chủ đề liên quan tới Angular trong vòng 100 ngày:

<https://github.com/angular-vietnam/100-days-of-angular>

1. Angular Docs: <https://angular.io/docs>