TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**──────── \* ───────**







**LỜI NÓI ĐẦU**

Ngày nay việc học của học sinh, sinh viên ngày càng nặng, nhà trường tạo ra nhiều môn học khác nhau cho sinh viên.

Mỗi môn học thì lại có nhiều lớp học và những mốc thời gian khác nhau.

Nên để sắp xếp một thời gian biểu hợp lý, phù hợp với thời gian của bản thân là một việc khác mất thời gian. Ngoài ra, mỗi môn học thì có rất nhiều bài tập về nhà, nhiều sự kiện đáng lưu ý để ghi chú lại.

Do đó sinh viên cần một công cụ thích hợp để trợ giúp. Vì vậy, bọn em quyết định làm ra một trang web giúp sắp xếp thời khóa biểu cho học sinh, sinh viên tiết kiệm thời gian và có được một thời khóa biểu hợp lý nhất cho học kì của mình cùng với chức năng hỗ trợ ghi trú cho từng môn học

PHÂN CÔNG CÔNG VIỆC TRONG NHÓM

| **Họ và Tên** | **Công việc thực hiện** | **% công việc thực hiện** | **Đánh giá** |
| --- | --- | --- | --- |
| Nguyễn Đức Bình | * Leader: phân công công việc cho các thành viên. * Tìm kiếm dữ liệu * Thiết kế database | 25% | Hoàn thành đúng tiến độ |
| Hoàng Hải Long | * UC CRUD notes * Làm giao diện cho trang web | 25% | Hoàn thành đúng tiến độ |
| Phạm Thị Minh Thư | * UC CRUD class, subject * Thuật toán schedule | 25% | Hoàn thành đúng tiến độ |
| Đoàn Anh Tuấn | * UC đăng ký, đăng nhập, update thông tin cá nhân,   Thay đổi mật khẩu | 25% | Hoàn thành đúng tiến độ |

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

# 1.1 Mô tả yêu cầu bài toán

***Xác định các tác nhân của hệ thống:***

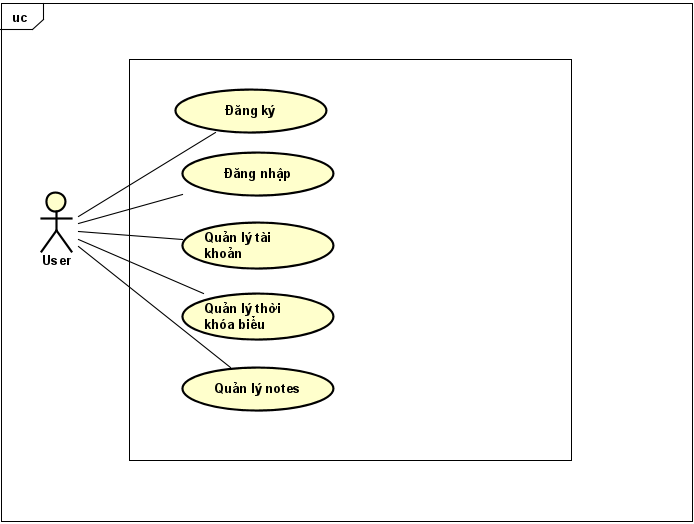
* Người sử dụng hệ thống: học sinh, sinh viên

| **STT** | **Tên tác nhân** | **Mô tả tác nhân** |
| --- | --- | --- |
| 1 | User | Là người trực tiếp sử dụng hệ thống |

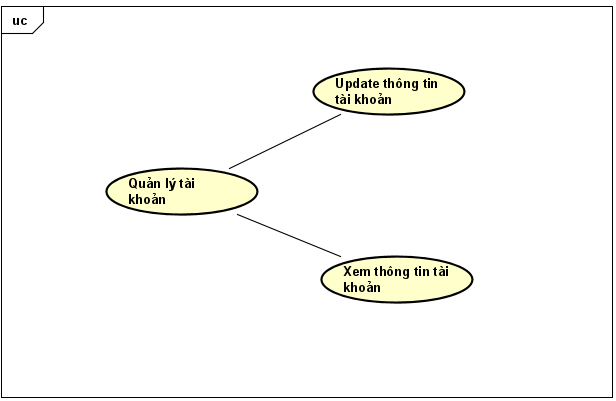
* **Các chức năng chính:**
* Đăng ký
* Đăng nhập
* Cập nhật thông tin tài khoản
* Thay đổi mật khẩu
* Đăng xuất
* Xem danh sách học phần
* Tạo thời khóa biểu tự động
* Thêm note
* Sửa note
* Xóa note
* Xem note

# 1.2 Biểu đồ Usecase

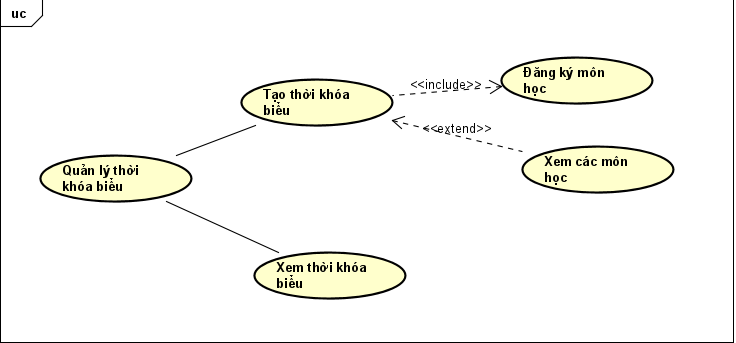
* Biểu đồ usecase tổng quan:



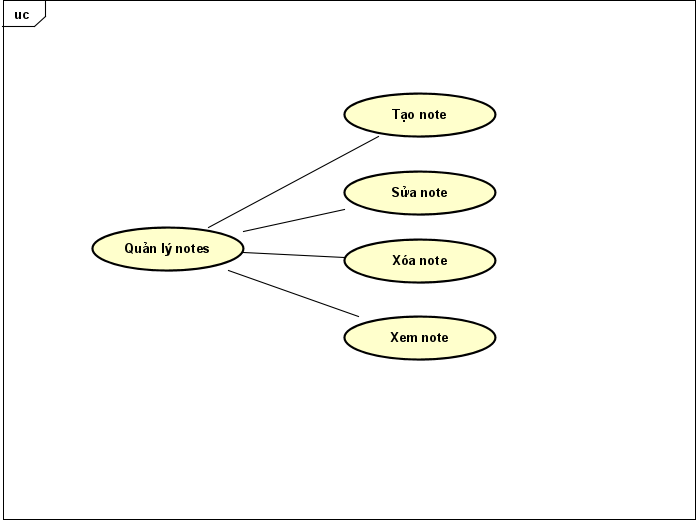
* Biểu đồ usecase phân rã mức 2
* Phân rã usecase quản lý tài khoản



* Phân rã usecase quản lý thời khóa biểu



* Phân rã usecase quản lý notes



# 1.3 Đặc tả usecase

***Đặc tả use case UC001 “Đăng nhập”***

| **Mã Use case** | UC001 | **Tên Use case** | Đăng nhập |
| --- | --- | --- | --- |
| **Tác nhân** | User | | |
| **Mô tả** | Cho phép một người dùng có tài khoản truy cập vào hệ thống | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính**  **(Thành công)** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 1. | User | chọn chức năng Đăng nhập | | 2. | Hệ thống | hiển thị giao diện đăng nhập | | 3. | User | nhập username và mật khẩu (mô tả phía dưới \*) | | 4. | User | yêu cầu đăng nhập | | 5. | Hệ thống | kiểm tra xem tác nhân đã nhập các trường bắt buộc nhập hay chưa | | 6. | Hệ thống | kiểm tra username và mật khẩu có hợp lệ do tác nhân nhập trong hệ thống hay không | | 7. | Hệ thống | tạo giao diện màn hình chính | | | |
| **Luồng sự kiện thay thế** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 6a. | Hệ thống | thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu người dùng nhập thiếu | | 7a. | Hệ thống | thông báo lỗi: username và/hoặc mật khẩu chưa đúng nếu không tìm thấy username và mật khẩu trong hệ thống | | | |
| **Hậu điều kiện** | Không | | |

***Đặc tả use case UC002 “Cập nhật thông tin tài khoản”***

| **Mã Use case** | UC002 | **Tên Use case** | Cập nhật thông tin tài khoản |
| --- | --- | --- | --- |
| **Tác nhân** | User | | |
| **Mô tả** | Cho phép người dùng thay đổi thông tin tài khoản của mình | | |
| **Tiền điều kiện** | Đã có tài khoản | | |
| **Luồng sự kiện chính**  **(Thành công)** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 1. | User | chọn chức năng update profile | | 2. | Hệ thống | hiển thị giao diện update profile | | 3. | User | Nhập những thông tin cần update như là Name, Date of birth, Gender, Phone number, Email, Avatar, Education, Url Facebook | | 4. | User | yêu cầu update profile | | 5. | Hệ thống | kiểm tra xem tác nhân đã nhập các trường bắt buộc nhập hay chưa | | 6. | Hệ thống | kiểm tra kiểu dữ liệu có hợp lệ do tác nhân nhập trong hệ thống hay không | | 7. | Hệ thống | Tạo giao diện My profile | | | |
| **Luồng sự kiện thay thế** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 6a. | Hệ thống | thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu người dùng nhập thiếu | | | |
| **Hậu điều kiện** | Các dữ liệu đã nhập được lưu vào cơ sở dữ liệu | | |

***Đặc tả use case UC003 “Đăng ký môn học”***

| **Mã Use case** | UC003 | **Tên Use case** | Đăng ký môn học |
| --- | --- | --- | --- |
| **Tác nhân** | User | | |
| **Mô tả** | Cho phép một người dùng đăng ký môn học cần để xếp thời khóa biểu | | |
| **Tiền điều kiện** | Biết được mã học phần của các môn học đó | | |
| **Luồng sự kiện chính**  **(Thành công)** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 1. | User | chọn chức năng đăng ký môn học | | 2. | Hệ thống | hiển thị giao diện đăng ký môn học | | 3. | User | nhập mã môn học cần đăng ký | | 4. | User | yêu cầu đăng ký | | 5. | Hệ thống | kiểm tra người dùng đã nhập đúng mã học phần môn học có trong hệ thống chưa | | 6. | Hệ thống | tạo giao diện những môn học đã đăng ký | | | |
| **Luồng sự kiện thay thế** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 5a. | Hệ thống | thông báo lỗi: Nếu nhập không đúng mã môn học có trong hệ thống | | | |
| **Hậu điều kiện** | Những môn học đã đăng ký được lưu vào trong cơ sở dữ liệu | | |

***Đặc tả use case UC004 “tạo thời khóa biểu”***

| **Mã Use case** | UC004 | **Tên Use case** | Tạo thời khóa biểu |
| --- | --- | --- | --- |
| **Tác nhân** | User | | |
| **Mô tả** | Cho phép người dùng tạo ra tất cả những thời khóa biểu từ những môn người dùng chọn | | |
| **Tiền điều kiện** | Đã chọn những môn học cần tạo thời khóa biểu | | |
| **Luồng sự kiện chính**  **(Thành công)** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 1. | User | chọn chức năng tạo thời khóa biểu | | 2. | Hệ thống | hiển thị giao diện tạo thời khóa biểu | | 3. | User | Chọn những môn học cần tạo thời khóa biểu | | 4. | User | yêu cầu tạo thời khóa biểu | | 5. | Hệ thống | Bắt đầu tạo thời khóa biểu | | 6. | Hệ thống | tạo giao diện hiển thị những thời khóa biểu đã được hệ thống tạo ra | | 7. | User | Chọn thời khóa biểu phù hợp | | 8. | User | Yêu cầu lưu lại | | | |
| **Luồng sự kiện thay thế** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 5a. | Hệ thống | thông báo lỗi: Nếu người dùng chưa chọn bất kì môn học nào | | | |
| **Hậu điều kiện** | Thời khóa biểu được chọn được lưu vào cơ sở dữ liệu | | |

***Đặc tả use case UC005 “tạo note”***

| **Mã Use case** | UC005 | **Tên Use case** | Tạo note |
| --- | --- | --- | --- |
| **Tác nhân** | User | | |
| **Mô tả** | Cho phép người dùng tạo ra những note ghi nhớ | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính**  **(Thành công)** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 1. | User | chọn chức năng tạo note | | 2. | Hệ thống | hiển thị giao diện tạo note | | 3. | User | Chọn ngày tạo note trên lịch | | 4. | User | Nhập tiêu đề, mô tả, thời gian bắt đầu, thời gian kết thúc của note | | 5. | User | Yêu cầu tạo | | 6. | Hệ thống | tạo giao diện hiển thị note đã được tạo ra | | | |
| **Luồng sự kiện thay thế** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 5a. | User | Yêu cầu hủy | | 6a.. | Hệ thống | Thông báo lỗi. | | | |
| **Hậu điều kiện** | Note tạo ra được lưu vào trong cơ sở dữ liệu | | |

***Đặc tả use case UC006 “sửa note”***

| **Mã Use case** | UC006 | **Tên Use case** | Sửa note |
| --- | --- | --- | --- |
| **Tác nhân** | User | | |
| **Mô tả** | Thay đổi nội dung của note | | |
| **Tiền điều kiện** | Note đã được tạo | | |
| **Luồng sự kiện chính**  **(Thành công)** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 1. | User | Click vào note muốn sửa | | 2. | Hệ thống | hiển thị giao diện note muốn sửa | | 4. | User | Nhập nội dung cần thay đổi như title, description, thời gian bắt đầu, thời gian kết thúc | | 5. | User | Yêu cầu sửa | | 6. | Hệ thống | tạo giao diện hiển thị note đã chỉnh sửa | | | |
| **Luồng sự kiện thay thế** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 5a. | User | Yêu cầu hủy | | 6a. | Hệ thống | Thông báo lỗi. | | | |
| **Hậu điều kiện** | Note đã sửa được cập nhật vào cơ sở dữ liệu | | |

***Đặc tả use case UC007 “xóa note”***

| **Mã Use case** | UC007 | **Tên Use case** | Xóa note |
| --- | --- | --- | --- |
| **Tác nhân** | User | | |
| **Mô tả** | Xóa note ra khỏi database | | |
| **Tiền điều kiện** | Note đã được tạo | | |
| **Luồng sự kiện chính**  **(Thành công)** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 1. | User | Click vào note muốn xóa | | 2. | Hệ thống | hiển thị giao diện note muốn xóa | | 3. | User | Yêu cầu xóa | | 4.. | Hệ thống | Xóa note đã tạo | | | |
| **Luồng sự kiện thay thế** | | **STT** | **Thực hiện bởi** | **Hành động** | | --- | --- | --- | | 3a.. | User | Yêu cầu hủy | | | |
| **Hậu điều kiện** | Note đã được xóa ra khỏi database | | |

CHƯƠNG 2: THIẾT KẾ GIAO DIỆN

## Thiết kế giao diện đảm bảo các tính chất:

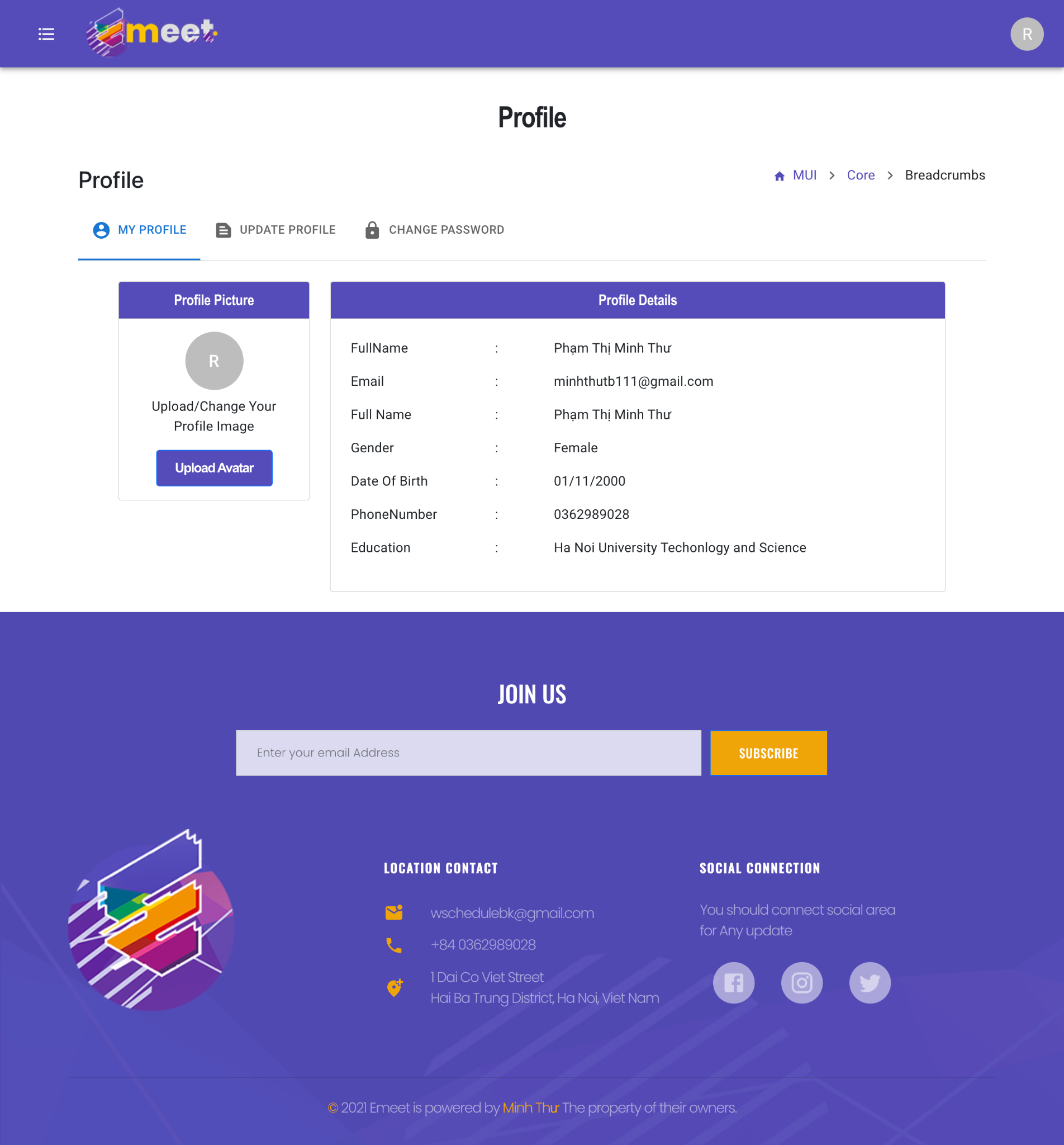
* Responsive: màn hình sẽ thay đổi phù hợp với kích thước màn hình. Nhóm đạt các tiêu chí trên thông qua sử dụng bootstrap
* Tái sử dụng các thành phần: nhóm thực hiện thông qua sử dụng ReactJS.

## Template sử dụng:

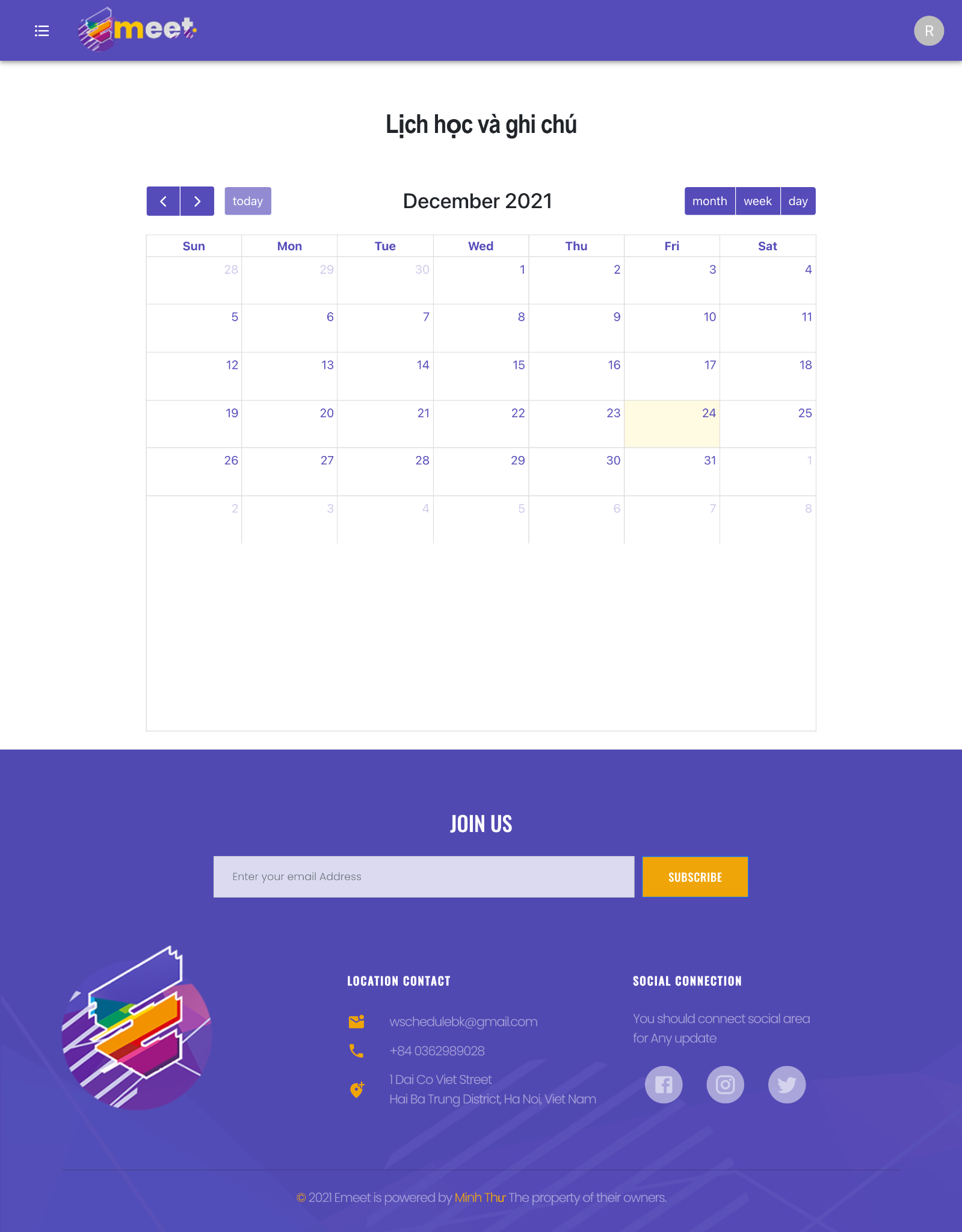
* Home: <http://themearth.com/demo/html/emeet/view/index.html>
* Calendar: <https://berrydashboard.io/app/calendar>
* Giao diện Home:

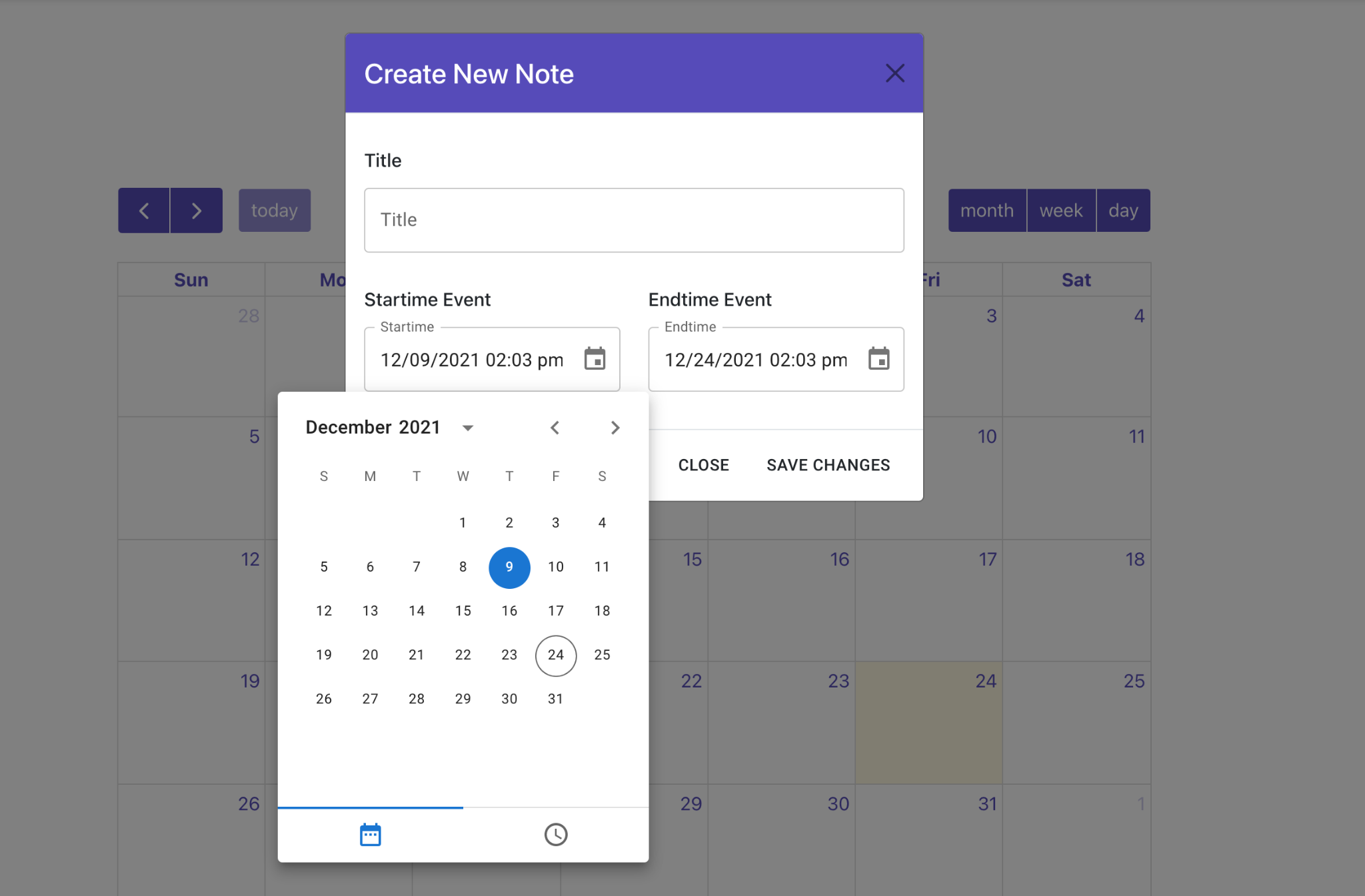


Giao diện profile

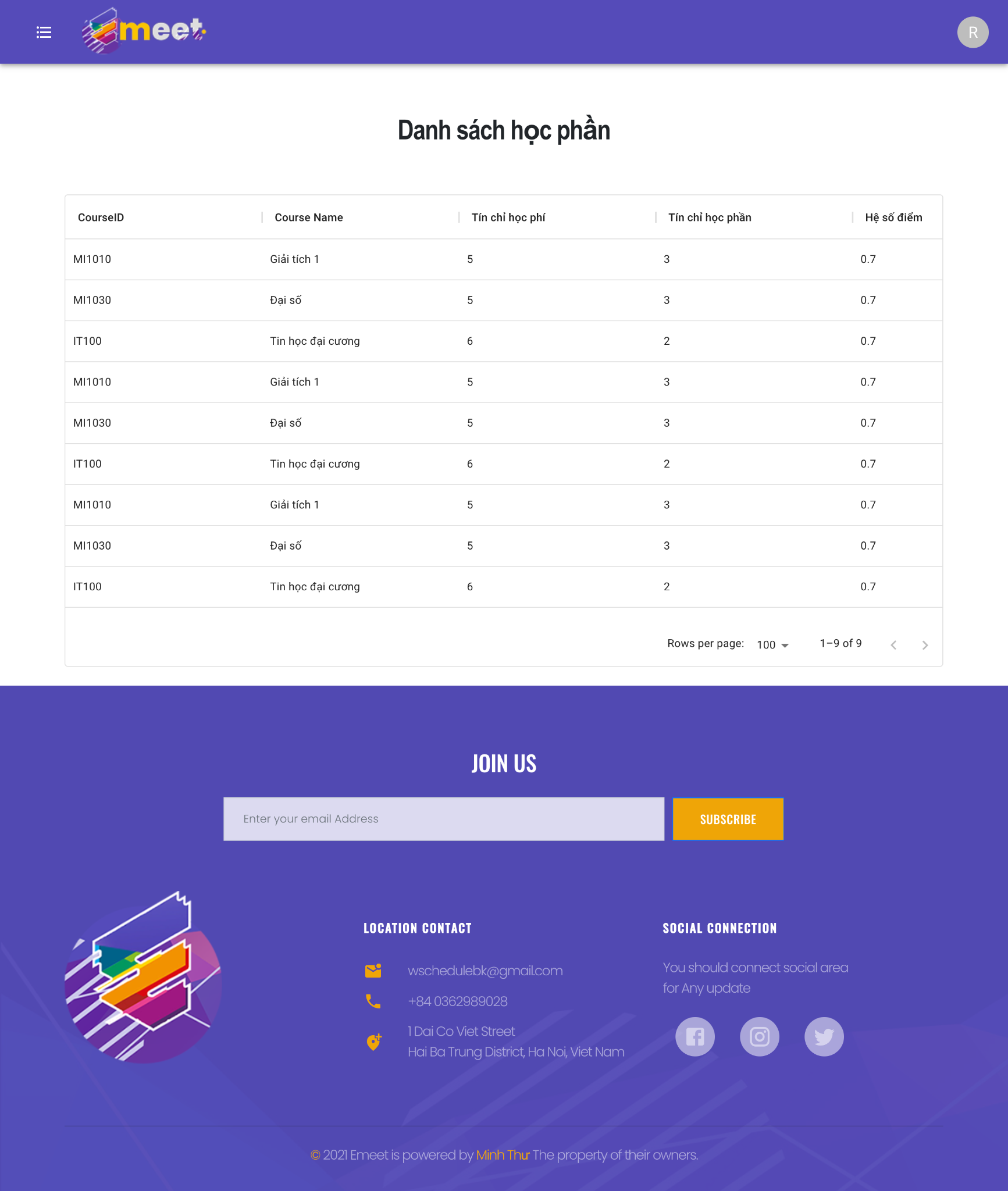


Giao diện note

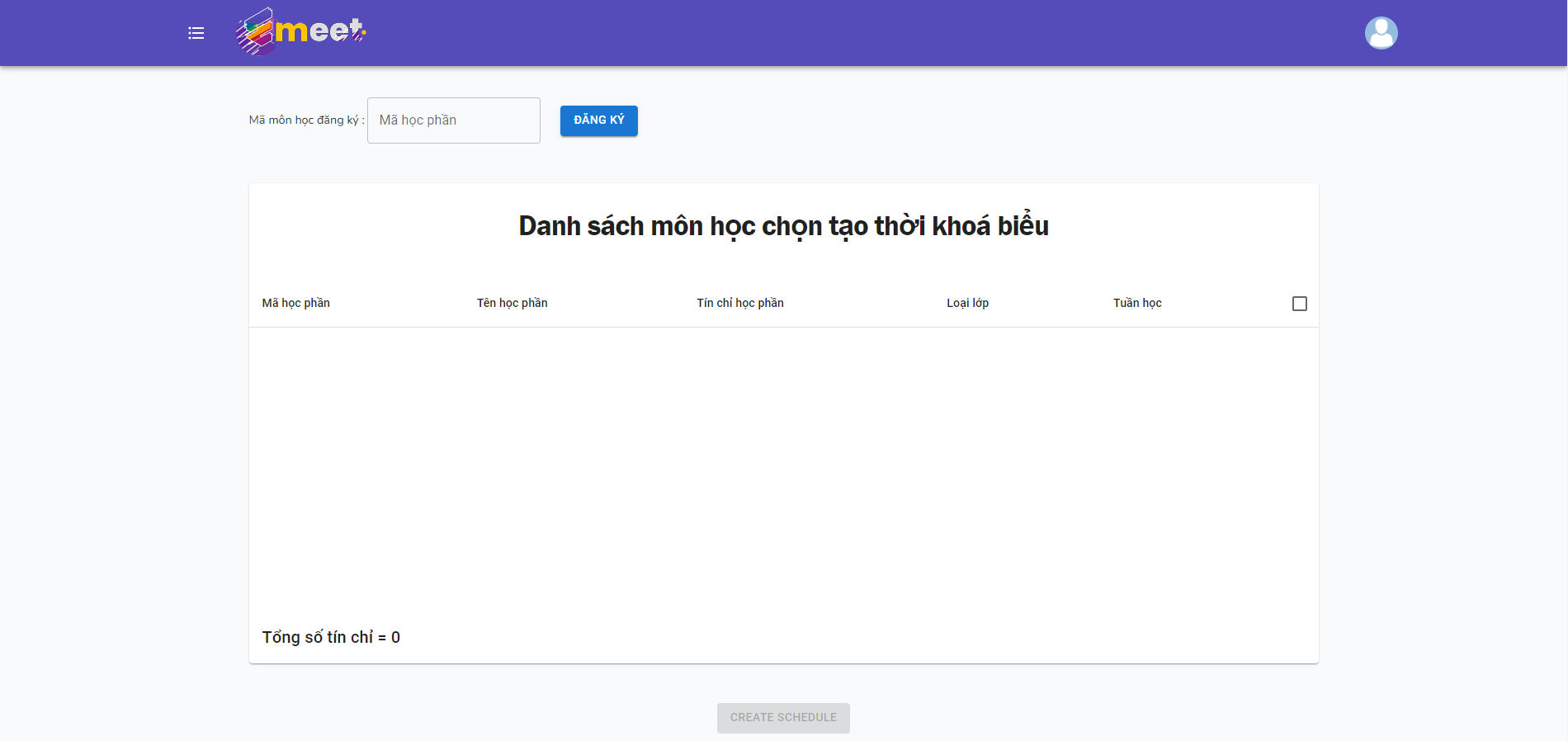




Giao diện xem danh sách subjects:

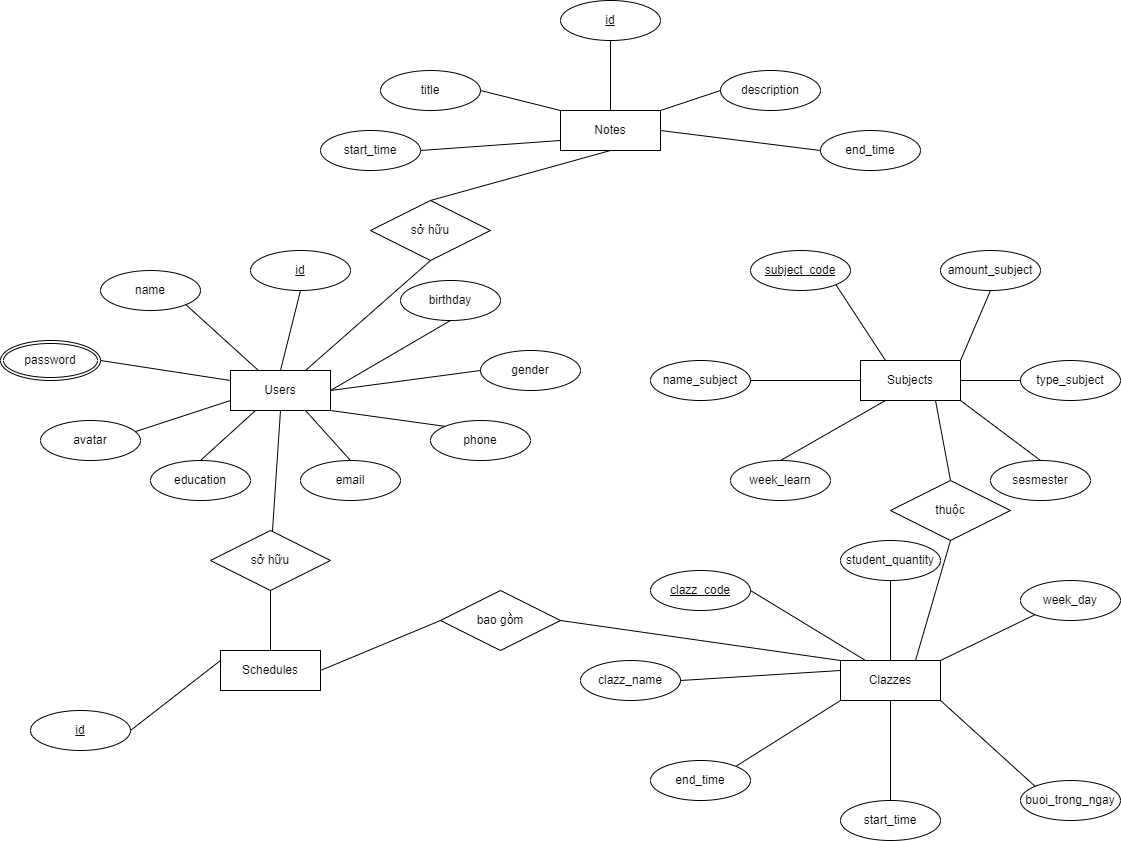


Giao diện tạo thời khóa biểu:

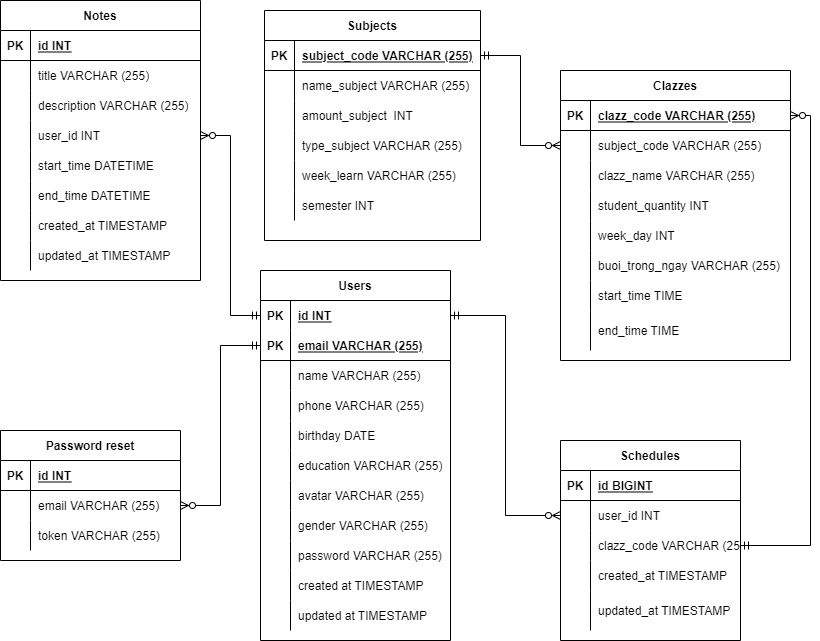


CHƯƠNG 3: THIẾT KẾ CƠ SỞ DỮ LIỆU

# 3.1 Sơ đồ thực thể liên kết:

****

# 3.2 Sơ đồ quan hệ:



# 3.3 Mô tả chi tiết:

* + Mỗi thực thể biểu diễn thành một bảng: users, clazzes, notes, subjects, schedules
  + Thuộc tính password là thuộc tính đa trị nên sang sơ đồ quan hệ được biểu diễn bằng 1 bảng
  + Quan hệ users-notes là quan hệ 1-n.
  + Quan hệ users-schedules là quan hệ 1-n.
  + Quan hệ giữa subjects-clazzes là quan hệ 1-n.
  + Quan hệ giữa schedules-clazzes là quan hệ n-m.

**A, Bảng users**

| **Attribute** | **Description** | **Data Type** |
| --- | --- | --- |
| id | Khóa chính dùng để phân biệt các người dùng với nhau. | INT |
| name | Tên người dùng | VARCHAR (255) |
| email | Email người dùng | VARCHAR (255) |
| phone | Số điện thoại người dùng | VARCHAR (255) |
| birthday | Ngày sinh người dùng | VARCHAR (255) |
| education | Trường người dùng đang học | VARCHAR (255) |
| gender | Giới tính | VARCHAR (255) |
| password | Mật khẩu | VARCHAR (255) |
| avatar | Ảnh người dùng | VARCHAR (255) |

**B, Bảng notes**

| **Attribute** | **Description** | **Data Type** |
| --- | --- | --- |
| id | Khóa chính dùng để phân biệt các note với nhau | INT |
| title | Tiêu đề của note | VARCHAR (255) |
| description | Miêu tả thêm về note | VARCHAR (255) |
| startTime | Thời điểm dự định bắt đầu | DATETIME |
| endTime | Thời điểm dự định kết thúc | DATETIME |

**C, Bảng schedules**

| **Attribute** | **Description** | **Data Type** |
| --- | --- | --- |
| scheduleID | Khóa chính dùng để phân biệt các schedule với nhau | BIGINT |

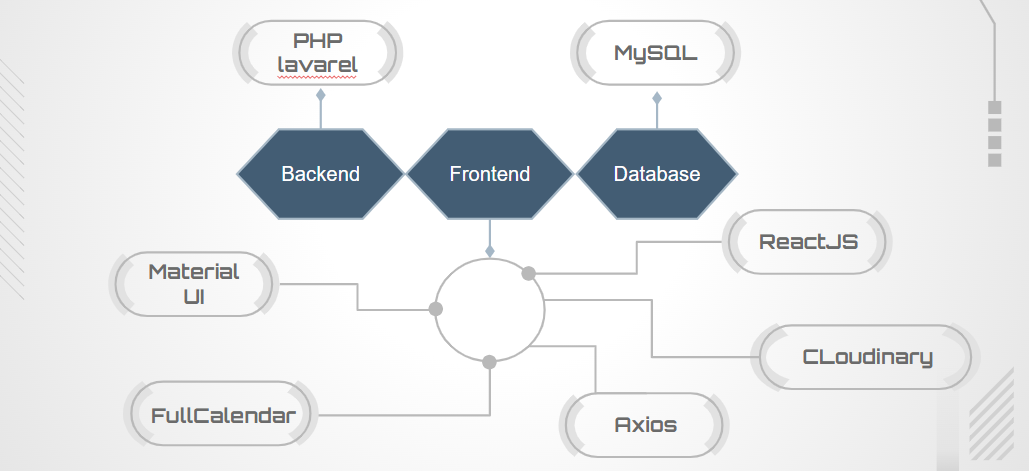
**D, Bảng subjects**

| **Attribute** | **Description** | **Data Type** |
| --- | --- | --- |
| subject\_code | Khóa chính dùng để phân biệt các subject với nhau | VARCHAR (255) |
| subject\_name | Tên của subject | VARCHAR (255) |
| amount\_subject | Số lượng subject trong kì | INT |
| type\_subject | Loại subject | VARCHAR (255) |
| week\_learn | Tuần học | VARCHAR (255) |
| semester | Kì học | INT |

**E, Bảng clazzes**

| **Attribute** | **Description** | **Data Type** |
| --- | --- | --- |
| class\_code | Khóa chính dùng để phân biệt các class với nhau | INT |
| className | Tên của class | VARCHAR (255) |
| student\_quantity | Số lượng sinh viên trong class | INT |
| week\_day | Những ngày diễn ra class trong tuần | VARCHAR (255) |
| buoi\_trong\_ngay | Buổi diễn ra lớp học | VARCHAR (255) |
| startTime | Thời gian bắt đầu trong ngày | TIME |
| endTime | Thời gian kết thúc trong ngày | TIME |

CHƯƠNG 4: CÔNG NGHỆ SỬ DỤNG



# 4.1 Cài đặt môi trường:

Nhóm cài đặt thông qua sử dụng composer và npm (Để quản lý các package như mix):

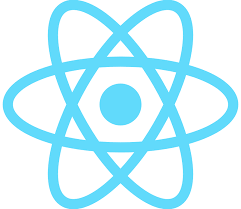
1. Cài đặt composer tại:

<https://getcomposer.org/download/>

1. Cài đặt npm tại: <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>
2. Cài đặt nodejs tại:

<https://hocwebchuan.com/tutorial/guide/nodejs/install_nodejs.php>

# 4.2 ReactJS



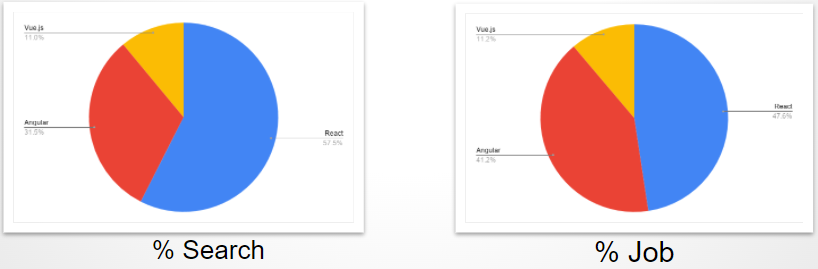
**Giới thiệu:**

* ReactJS là một thư viện JavaScript mã nguồn mở được Facebook thiết kế với mục đích tạo các ứng dụng web phong phú và hấp dẫn một cách nhanh chóng và hiệu quả với mục đích giảm thiểu việc code. Điều mà ReactJS mang lại chính là cung cấp hiệu suất hiển thị tốt nhất có thể. Sức mạnh của nó đến từ việc tập trung vào các components riêng lẻ- thay vì làm việc trên toàn bộ ứng dụng web, ReactJS cho phép nhà phát triển chia nhỏ giao diện người dùng phức tạp thành nhiều thành phần đơn giản hơn (components).

* Hiện tại, ReactJS đã trở nên RẤT PHỔ BIẾN vì tính đơn giản và linh hoạt của nó. Nhiều người thậm chí còn gọi nó là cả tương lai của phát triển của nền tảng web. Ước tính có hơn 1.300 nhà phát triển và hơn 94.000 trang web sử dụng ReactJS.

* Tỉ lệ tìm kiếm trên gg và tỉ lệ job cũng đứng hàng đồng khi so với các framework khác của js.

* Một phần của sự phổ biến khổng lồ này đến từ việc: Các tập đoàn hàng đầu như Facebook, PayPal, Uber, Instagram và Airbnb sử dụng ReactJS để giải quyết các vấn đề liên quan đến giao diện người dùng, Sự tín nhiệm này đã thu hút rất nhiều người đổ xô vào dùng ReactJS.



**Lý do nên dùng ReactJS:**

* Nó cho phép lập trình viên viết ứng dụng của họ trong JavaScript. JSX(một cú pháp duy nhất cho phép trích dẫn HTML cũng như ứng dụng cú pháp thẻ HTML để hiển thị các components cụ thể) là một trong những tính năng tuyệt vời nhất không chỉ làm cho ReactJS trở nên dễ dàng, dễ sử dụng mà còn thú vị. Các dev có thể dễ dàng tạo một tính năng giao diện người dùng mới và thấy nó xuất hiện một cách nhanh chống. Nó đưa HTML trực tiếp vào JS của trang web.
* Cho phép họ chia nhỏ các thành phần phức tạp và tái sử dụng chúng trong xuyên suốt dự án. Ý tưởng về các components là điều làm cho ReactJS trở nên độc đáo và hấp dẫn. Thay vì lo lắng về toàn bộ ứng dụng web, bạn có thể chia nhỏ quá trình phát triển UI / UX phức tạp thành các components con đơn giản hơn. Điều này rất quan trọng trong việc làm cho cả trang web trở nên trực quan hơn.
* ReactJS chính xác là đang cung cấp giải pháp mà các dev đang tìm kiếm. Nó sử dụng JSX Điều này rất hữu ích trong việc thúc đẩy việc xây dựng các mã máy có thể đọc được và đồng thời kết hợp các thành phần thành một tệp có thể xác minh một lần.
* Props giúp bạn có thể dễ dang sử dụng dữ liệu một cách tùy chỉnh hơn. Ưu điểm trong ReactJS giúp bạn có thể chuyển dữ liệu tùy chỉnh đến một thành phần giao diện người dùng cụ thể. Đây là một cách tuyệt vời để lấp đầy khoảng trống khi hiển thị các thành phần JSX
* State giúp bạn có thể lưu trữ tất cả các thay đổi trong một class/component cụ thể. Khi sử dụng ReactJS, các nhà phát triển lấy tất cả nội dung có thể thay đổi trong quá trình ứng dụng và đặt nó vào một vị trí duy nhất (State). Với trạng thái, ứng dụng web đơn lẻ trở nên rất đơn giản vì tất cả các chi tiết đều có nguồn gốc từ một phần.

Nó sử dụng Virtual DOM giúp ứng dụng nhanh hơn. Khi ReactJS được tạo ra, các nhà thiết kế đã dự đoán đúng rằng việc cập nhật HTML cũ sẽ trở nên cực kỳ quan trọng và tốn kém tài nguyên. Do đó, ý tưởng về Virtual DOM chính là từ khóa giúp ReactJS biết chính xác khi nào cần kết xuất lại hoặc khi nào bỏ qua một số phần cụ thể của DOM vì nó có thể phát hiện khi nào dữ liệu đã thay đổi. Giao diện người dùng phản ứng kịp thời là rất quan trọng trong việc nâng cao trải nghiệm người dùng.

**Hướng dẫn cài đặt:**

Link: <https://hocwebchuan.com/tutorial/reactjs/install_reactjs.php>

# 4.3 Material UI

* UI là một framework giao diện người dùng, mã nguồn mở tập hợp các components của React với hơn 74k stars trên github. MaterialUI dựa trên Material Design của Google để cung cấp trải nghiệm kỹ thuật số, chất lượng cao trong khi phát triển đồ họa front end.
* Với rất nhiều components được xây dựng sẵn, cùng với khả năng tùy biến cao, Material-UI hứa hẹn đáp ứng cho tất cả dự án từ nhỏ đến lớn.

**Ưu điểm:**

* Một số framework frontend không được ghi tài liệu đầy đủ, dẫn đến việc khó có thể phát triển với chúng. Tuy nhiên, material ui có tài liệu đầy đủ, giúp ta dễ dàng tìm hiểu về framework này.

* Material UI luôn được cập nhật thường xuyên. Tại thời điểm này, bản cập nhật gần đây nhất là 5.3.0

* Các components xuyên suốt, nhất quán về thiết kế, tông màu, cho phép ứng dụng, trang web được phát triển hấp dẫn về mặt thẩm mỹ.

# 4.4 Fullcalendar

* Hiện nay phần lớn các bạn khi làm việc sẽ đều làm một điều đó là lên lịch làm việc hàng ngày, hàng tuần hay thậm chí là hàng tháng. Với các ứng dụng, dịch vụ trực tuyến như Google Calendar các bạn sẽ dễ dàng lên lịch làm việc của mình. Đây là nhu cầu của mọi người còn đối với các bạn lập trình viên thì việc tạo ra một ứng dụng lịch để lên lịch làm việc, các sự kiện cung cấp đến cho người dùng thì như thế nào? Rất đơn giản, các bạn có thể sử dụng thư viện FullCalendar để thiết kế các ứng dụng lịch một cách nhanh chóng.
* Thư viện FullCalendar có thể tạo ra nhiều loại lịch khác nhau như lịch theo tuần, theo tháng, theo năm với nhiều trường thông tin bổ sung. Ngoài ra, các bạn còn có thể tạo các sự kiện ngay trên lịch. Hiện nay, thư viện FullCalendar được nhiều ứng dụng web sử dụng bởi tính linh động, chuyên nghiệp mà thư viện mang lại cho các bạn lập trình viên.

# 4.5 Cloudinary

* Cloudinary là một cloud-based service, nó cung cấp một giải pháp quản lý hình ảnh bao gồm upload, lưu trữ, thao tác, tối ưu hóa và delivery.
* Với cloudinary bạn có thể dễ dàng upload ảnh lên cloud, tự động thực thi các thao tác với ảnh một cách thông minh mà không cần phải cài đặt bất kì một phần mềm phức tạp nào khác. Cloudinary cung cấp các APIs toàn diện và màn hình quản lý giúp chúng ta dễ dàng tích hợp vào các trang web và ứng dụng di động

# 4.6 Axios

* ​​Axios là một HTTP client được viết dựa trên Promises được dùng để hỗ trợ cho việc xây dựng các ứng dụng API từ đơn giản đến phức tạp và có thể được sử dụng cả ở trình duyệt hay Node.js
* Việc tạo ra một HTTP request dùng để fetch hay lưu dữ liệu là một trong những nhiệm vụ thường thấy mà một ứng dụng Javascript phía client cần phải làm khi muốn giao tiếp với phía server. Các thư viện bên thứ 3, đặc biệt là jQuery từ xưa đến nay vẫn là một trong những cách phổ biến để giúp cho các browser API tương tác tốt hơn, rõ ràng mạch lạc hơn và xóa đi những điểm khác biệt giữa các browser với nhau
* Có lẽ bạn cũng đã để ý rằng, các browser hiện đại bây giờ thường đi kèm và xây dựng sẵn các tính năng [Fetch API](https://www.sitepoint.com/introduction-to-the-fetch-api/) mới hơn, vậy tạo sao ta lại không dùng nó luôn cho xong? Rất nhiều những sự khác biệt giữa Axios và Fetch khiến rất nhiều người vẫn lựa chọn Axios thay vì Fetch.
* Axios sử dụng thuộc tính data, nhưng fetch thì dùng body để chứa data.
* Data của axios đã tự động convert sang json, còn data của fetch là ở dạng string
* Url trong Axios là đối tượng config, còn trong fetch là 1 tham số URL và options

# 4.7 Laravel ( Framework)

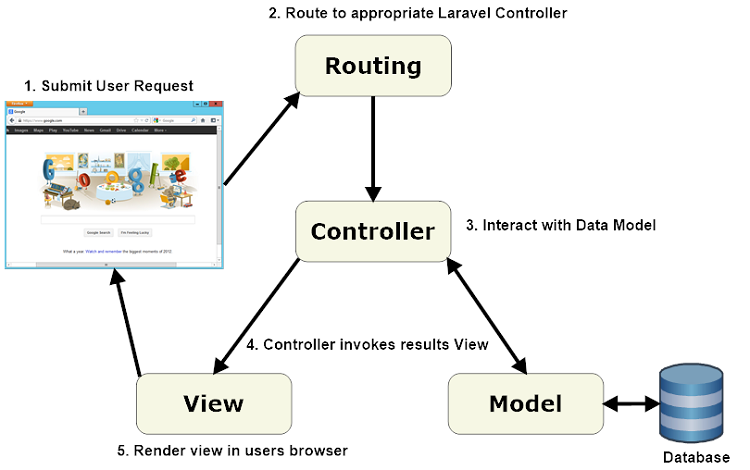
****

**Giới thiệu:**

* Laravel là một PHP framework, có mã nguồn mở và miễn phí, được xây dựng nhằm hỗ trợ phát triển các phần mềm, ứng dụng, theo kiến trúc MVC. Hiện nay, Laravel đang là PHP framework phổ biến nhất và tốt nhất.
* Theo TIOBE, tháng đầu 2020 PHP xếp hạng 8 trong các ngôn ngữ lập trình phổ biến nhất



* MVC (Model-View-Controller) là mẫu kiến trúc phần mềm trên máy tính nhằm mục đích tạo lập giao diện cho người dùng. Theo đó, hệ thống MVC được chia thành ba phần có khả năng tương tác với nhau và tách biệt các nguyên tắc nghiệp vụ với giao diện người dùng.



* Ba thành phần ấy bao gồm:

* Controller: Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng.
* Model: Là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý…
* View: Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images…

Bằng cách này, thông tin nội hàm được xử lý tách biệt với phần thông tin xuất hiện trong giao diện người dùng. Bảo đảm các nguyên tắc nghề nghiệp của lập trình viên.

**Ưu điểm:**

* Nguồn tài nguyên vô cùng lớn và sẵn có

Nguồn tài nguyên của Laravel rất thân thiện với các lập trình viên với đa dạng tài liệu khác nhau để tham khảo. Các phiên bản được phát hành đều có nguồn tài liệu phù hợp với ứng dụng của mình.

* Tích hợp với dịch vụ mail

Lavarel là framework được trang bị API sạch trên thư viện SwiftMailer, do đó, bạn có thể gửi thư qua các dịch vụ dựa trên nền tảng đám mây hoặc local.

* Tốc độ xử lý nhanh

Laravel hỗ trợ hiệu quả cho việc tạo lập website hay các dự án lớn trong thời gian ngắn. Vì vậy, nó được các công ty công nghệ và lập trình viên sử dụng rộng rãi để phát triển các sản phẩm của họ.

* Dễ sử dụng

Laravel được đón nhận và trở nên sử dụng phổ biến vì nó rất dễ sử dụng. Thường chỉ mất vài giờ, bạn có thể bắt đầu thực hiện một dự án nhỏ với vốn kiến thức cơ bản nhất về lập trình với PHP.

* Tính bảo mật cao

Laravel cung cấp sẵn cho người dùng các tính năng bảo mật mạnh mẽ để người dùng hoàn toàn tập trung vào việc phát triển sản phẩm của mình:

Sử dụng PDO để chống lại tấn công SQL Injection.

Sử dụng một field token ẩn để chống lại tấn công kiểu CSRF.

**Áp dụng:**

* Nên áp dụng Laravel ở những dự án vừa và nhỏ

**Cấu trúc của Laravel:**

lar-53/

|--- app/

│ |--- Console

│ |--- Exceptions

│ |--- Http

│ │ |--- Controllers/

│ |--Auth

│ |--Controller.php

│ │ |--- Middleware

│ │ └── Kernel.php

│ |--- Providers

| └--- User.php

|--- bootstrap/

|--- config/

|--- database/

|--- public/

| |--- ...

| |--- index.php

|--- resources/

| |--- assets

| |--- lang

| |--- views

|--- routes/

| |--- api.php

| |--- console.php

| |--- web.php

|--- storage/

|--- tests/

|--- vendor/

|--- .env

|--- .env.example

|--- .gitattributes

|--- .gitignore

|--- artisan

|--- composer.json

|--- composer.lock

|--- gulpfile.js

|--- package.json

|--- phpunit.xml

|--- readme.md

|--- server.php

**Chú thích:**

* **app**: Chứa các thư mục, các tập tin php, thư viện, models.
  + **Console**: Chứa các tập tin định nghĩa các câu lệnh trên artisan.
  + **Exception**: Chứa các tập tin quản lý, điều hướng lỗi.
  + **Http**
    - **Controllers** : Chứa các controllers của project.
    - **Middleware**: Chứa các tập tin lọc và ngăn chặn các requests.
    - **Kernel**.**php**: Cấu hình, định nghĩa Middleware.
  + **Providers**: Chứ các providers thực hiện việc binding vào service container(ở phần nâng cao mình sẽ nói sau).
  + **User**.**php**: Là model User mà Laravel tự địn sẵn cho chúng ta.
* **bootstrap**:Chứa tập tin điều hướng hệ thống.
* **config**: Chứa mọi tập tin cấu hình của Laravel.
* **database**: Chứa các thư mục tập tin vềdatabase.
  + **migrations**: Chứa các tập tin định nghĩa,khởi tạo và sửa bảng.
  + **seeds**: Chứa các tập tin định nghĩa dữ liệu insert(thêm) vào trong database.
  + **factories**: Chứa các tập tin định nghĩa các cột bảng dữ liệu để tạo ra các dữ liệu ảo.
* **public**: Chứa các tập tin css, js, image.
  + **index**.**php**: Đây là tệp tin root của Laravel.
* **resources**: Chứa các views, ngôn ngữ(language) của project.
* **routes**: Chứa các tập tin định nghĩa các router, xử lý điều hướng router bao gồm : web, api và console
* **storage**: Chứa các tập tin hệ thống cache, session, ...
* **tests**: ...
* **vendor**: Chứa các package mà laravel yêu cầu.
* .**env**: Là tập tin cấu hình chính của laravel như key app,database.
* .**env.example**: Tệp tin cấu hình mẫu của laraver.
* **composer.json**: tập tin của composer.
* **composer.lock**: tập tin của composer.
* **package.js**: Tập tin cấu hình của [nodejs](https://toidicode.com/nodejs-la-gi-26.html) (chứa các package cần dùng cho projects).
* **gulpfile.js**: Là tập tin gulp builder.
* **phpunit.xml**: Là tập tin xml của phpunit dùng để testing project.
* **server.php:** Là tập tin để artisan trỏ đến tạo server khi gõ lệnh php artisan serve .
* **artisan**: Tập tin thực thi lệnh của Laravel.

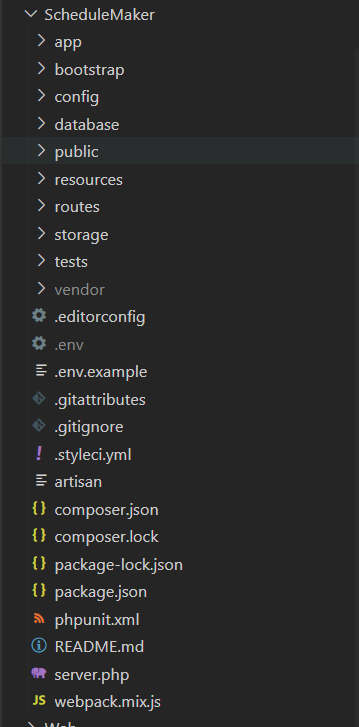
**Hướng dẫn cài đặt Laravel:**

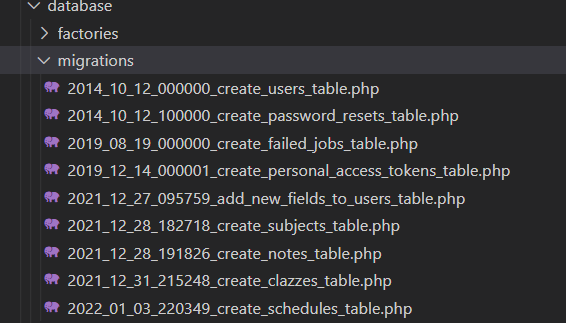
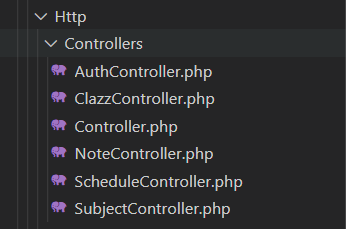
### Yêu cầu về hệ thống để cài đặt Laravel

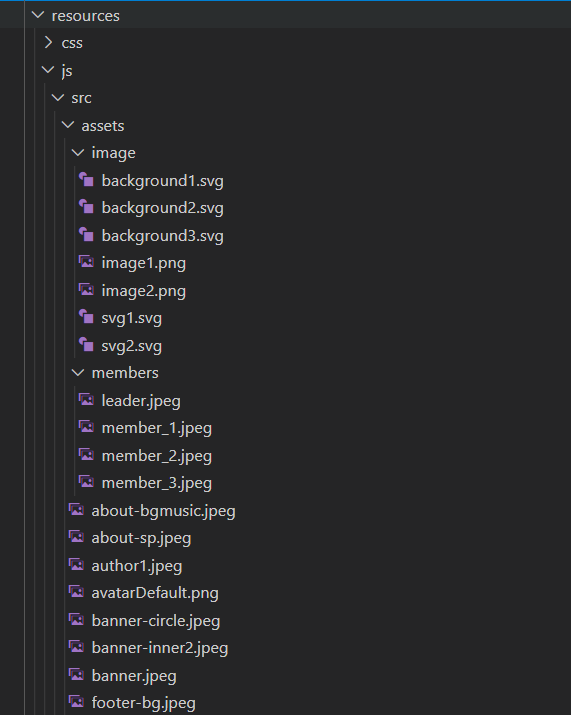
* Để có thể cài đặt Laravel, bạn cần đáp ứng các yêu cầu bắt buộc như:
* PHP >= 5.5.9
* OpenSSL PHP Extension
* PDO PHP Extension
* Mbstring PHP Extension
* Tokenizer PHP Extension
* Đối với Windows cần sử dụng phần mềm tạo Webserver trên windows như Wamp, Xampp, Openserver,...
* Link: <https://allaravel.com/blog/cai-dat-laravel-nhanh-cho-he-dieu-hanh-windows>

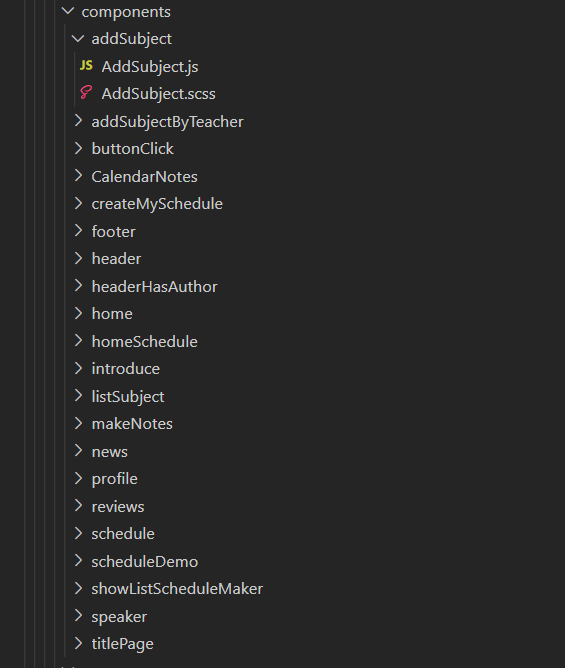
CHƯƠNG 5: MÔ TẢ MÃ NGUỒN

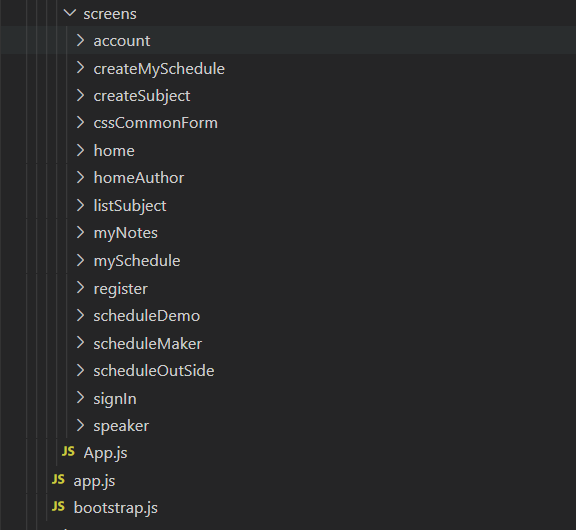
* Mã nguồn được tổ chức như một project của laravel trong đó có:
* routes: quản lý các đường dẫn, api sử dụng cho web.
* resources: quản lý các file view để render cho người dùng.
* app: chứa các file controller quản lý các trang của web.
* database: chứa các file migration để quản lý cơ sở dữ liệu.
* Cụ thể:











CHƯƠNG 6: TỔNG KẾT

## **Công việc đạt được:**

* Hoàn thiện những chức năng đề ra từ trước. Tạo ra một công cụ giúp cho sinh viên có thể tạo ra thời khóa biểu dễ dàng, nhanh chóng, phù hợp với thời gian, yêu cầu của mỗi người.

## **Định hướng tương lai cho web:**

* Nâng cấp về giao diện
* Thông báo về ghi chú của mỗi người ( Hệ thống tự động gửi mail nhắc nhở người dùng)
* Thêm chức năng liên quan đến quản lý bài tập: Thầy cô có thể gửi bài tập cho sinh viên => cập nhật bài tập hoặc thông báo thông tin gì đó cho sinh viên => sinh viên có thể quản lý bài tập về nhà hay thông báo của giáo viên về môn học của mình ngay trên web.
* Thêm chức năng mới cho giáo viên: Giáo viên có thể tạo môn học, quản lý môn học ngay trên web
* Thêm tùy chọn khi tạo thời khóa biểu: vd như là chọn khung giờ không muốn học, để ứng dụng sẽ không đề xuất những thời khóa biểu trùng vào giờ đó, ...