

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG
-----o0o-----



BÁO CÁO THỰC TẬP

ĐỀ TÀI: NHẬP MÔN LARAVEL

GVHD: Thạc sĩ Nguyễn Khánh Lợi

SVTH: Nguyễn Đăng Thức

MSSV: 2014691

TP. HỒ CHÍ MINH, THÁNG 8 NĂM 2023

LỜI CẢM ƠN

Trước tiên em xin được bày tỏ sự trân trọng và lòng biết ơn đối với thầy Thạc sĩ Nguyễn Khánh Lợi giảng viên bộ môn Viễn Thông thuộc khoa Điện – điện tử.

Trong suốt thời gian học và làm đề tài thực tập, thầy đã dành rất nhiều thời gian quý báu để tận tình chỉ bảo, hướng dẫn, định hướng cho em thực hiện đề án. Đặc biệt là chỉ ra những điểm yếu trong quá trình lên ý tưởng và thực hiện nó trên phần mềm.

Thực tập không chỉ là một nhiệm vụ, trải nghiệm cá nhân, mà còn là sự đóng góp của những người xung quanh. Do đó, em cũng muốn bày tỏ lòng biết ơn đến gia đình, bạn bè và những người thân yêu đã luôn đồng hành và ủng hộ tôi trong suốt thời gian học tập. Sự quan tâm và động viên của họ đã là nguồn động lực lớn để em vượt qua khó khăn và hoàn thành thực tập một cách thành công. Em xin được cảm ơn các thầy cô giáo đã giảng dạy trong quá trình học tập, thực hành, làm bài tập, giúp em hiểu thấu đáo hơn các nội dung học tập và những hạn chế cần khắc phục trong việc học tập, nghiên cứu và thực hiện đề tài thực tập này.

Em cũng muốn bày tỏ lòng biết ơn đến các thầy cô đã dành thời gian và công sức để đánh giá đề tài thực tập của em. Những góp ý và nhận xét chân thành từ thầy cô sẽ giúp em cải thiện và phát triển kỹ năng chuyên môn của mình trong tương lai.

Tp. Hồ Chí Minh, ngày 5 tháng 8 năm 2023 .

Sinh viên

Nguyễn Đăng Thúc

TÓM TẮT ĐỀ TÀI THỰC TẬP

1.1 Nhiệm vụ được giao thực tập

Đợt thực tập nhằm mục đích giúp sinh viên thực tập được đào tạo toàn diện về quá trình xây dựng Website trên nền tảng Laravel Framework. Nội dung thực tập bao gồm 15 nhiệm vụ (task). Từ task 0 đến task 14 được xây dựng để phù hợp cho sinh viên và có thể tóm gọn thành các nhiệm vụ sau :

Nội dung 1: Làm quen cách sử dụng các công cụ Git, Github, visual studio code,...

Nội dung 2: Tìm hiểu SQL, thực hành trên Xampp. Thực hiện các câu truy vấn.

Nội dung 3: Tìm hiểu PHP và Laravel Framework. Sử dụng Postman để thực hành.

Nội dung 4: Tìm hiểu Laravel, mô hình MVC khi xây dựng 1 Website.

Nội dung 5: Tìm hiểu Vuejs và cách nhúng Vuejs vào Laravel.

Nội dung 6: Áp dụng những lý thuyết đã tìm hiểu để thực hiện 1 ứng dụng hoàn chỉnh.

1.2 Thời gian và lịch trình thực tập

Thời gian thực tập: 2 tháng (từ ngày 18/06 đến 10/08)

Lịch trình thực tập:

Tuần 1 + 2: Thực hiện đọc tài liệu

Tuần 3 + 4: Thực hiện cài đặt dự án

Tuần 5 + 6: Thực hiện viết code các chức năng

Tuần 7 : Thực hiện sửa các lỗi khác nhau và báo cáo sau kì thực tập

MỤC LỤC

TASK 0. CÀI ĐẶT DỰ ÁN VÀO MÁY TÍNH	7
1. Các bước thực hiện.....	7
2. Kết quả đạt được sau khi thực hiện Task 0	8
TASK 1. LARAVEL MIX	9
1. Các bước thực hiện.....	9
2. Kết quả đạt được sau khi hoàn thành Task 1	11
TASK 2. TÌM HIỂU LÝ THUYẾT VÀ THỰC HÀNH.....	12
1. Kiến trúc MVC trong lập trình Web	12
2. Khái niệm về RESTFul API.....	13
3. Tìm hiểu về cơ sở dữ liệu quan hệ (MySql)	14
4. Thực hành và triển khai lý thuyết cho dự án.....	15
5. Sử dụng PostmanUI để test API.....	17
6. Kết quả:	17
TASK 3. VALIDATION TRONG LARAVEL	18
1. Các bước thực hiện.....	18
2. Kết quả:	19
TASK 4. TÌM HIỂU VỀ SEEDER	20
1. Lý thuyết chung.....	20
2. Các bước thực hiện.....	20
2. Kết quả:	22
TASK 5. THỰC HIỆN TRANG CẬP NHẬT THÔNG TIN SINH VIÊN.....	23
1. Các bước thực hiện.....	23
2. Kết quả:	26
TASK 6. TẠO TRANG THÔNG TIN CHO GIÁO VIÊN	27

1. Các bước thực hiện.....	27
2. Kết quả:	29
TASK 7, 8, 9. JOB VÀ IMPORT DANH SÁCH GIẢNG VIÊN	30
1. Các bước thực hiện.....	30
2. Kết quả:	37
TASK 10. TẠO DANH SÁCH GIẢNG VIÊN GIẢNG DẠY	38
TASK 11. TẠO BIỂU MẪU SINH VIÊN UPLOAD FILE	41
TASK 12. TẠO BIỂU MẪU GIÁO VIÊN CHẤM ĐIỂM.....	45
TASK 13. EXPORT BẢNG ĐIỂM.....	49
TASK 14. UPLOAD AVATAR NGƯỜI DÙNG	53
TÀI LIỆU THAM KHẢO	54

DANH SÁCH HÌNH MINH HỌA

Hình 1. Kết quả sau khi chạy dự án hiển thị trên terminal	8
Hình 2. Kết quả sau khi chạy dự án hiển thị trên Chrome	8
Hình 3. Kết quả sau giao diện đăng nhập hiển thị trên Chrome.....	11
Hình 4. Kết quả sau giao diện đăng kí hiển thị trên Chrome	11
Hình 6. Mô tả cách hoạt động của RESTful API	13
Hình 7. Kết quả sau khi tạo Seeder	22
Hình 8. Kết quả sau khi tạo create_sinhviens_table.....	24
Hình 9. Giao diện cập nhật thông tin sinh viên	26
Hình 10. Kết quả sau khi cập nhật thông tin sinh viên.....	26
Hình 11. Giao diện cập nhật thông tin giáo viên.....	29
Hình 12. Kết quả sau khi cập nhật thông tin giáo viên	29
Hình 13. Giao diện Import danh sách giảng viên.....	31
Hình 14. Kết quả sau khi Import danh sách giảng viên	37
Hình 15. Giao diện tạo mới giảng viên giảng dạy.....	38
Hình 16. Giao diện trả về khi đăng kí thành công.....	40
Hình 17. Giao diện sinh viên upload file.....	41
Hình 18. Kết quả được lưu sau khi nộp bài.....	44
Hình 19. Kết quả được lưu sau khi nộp bài ở phpMyadmin	44
Hình 20. File được lưu ở storage/app/public/reports.....	44
Hình 21. Giao diện chấm điểm giáo viên	45
Hình 22. Kết quả chấm điểm cho sinh viên	48
Hình 23. Giao diện export bảng điểm	49
Hình 24. Kết quả export bảng điểm thành file csv	52

Hình 25. Giao diện upload avatar người dùng	53
Hình 26. Trước khi cập nhật avatar người dùng	53
Hình 27. Sau khi cập nhật avatar người dùng	53

TASK 0. CÀI ĐẶT DỰ ÁN VÀO MÁY TÍNH

1. Các bước thực hiện

Truy cập vào link github để tải dự án về máy:

```
https://github.com/truongnguyen0128/bktel.git
```

Sau khi clone source code thực hiện chạy các lệnh sau để cài đặt các thư viện cần thiết

```
composer install
```

```
npm i
```

Sau đó tạo file .env mà thiết lập kết nối với MariaDB

```
DB_CONNECTION=mysql
```

```
DB_HOST=127.0.0.1
```

```
DB_PORT=3306
```

```
DB_DATABASE=bktel_project
```

```
DB_USERNAME=root
```

```
DB_PASSWORD=
```

Cuối cùng chạy lệnh php artisan

```
php artisan key:generate
```

Để khởi chạy dự án

```
npm run dev
```

```
npm run watch
```



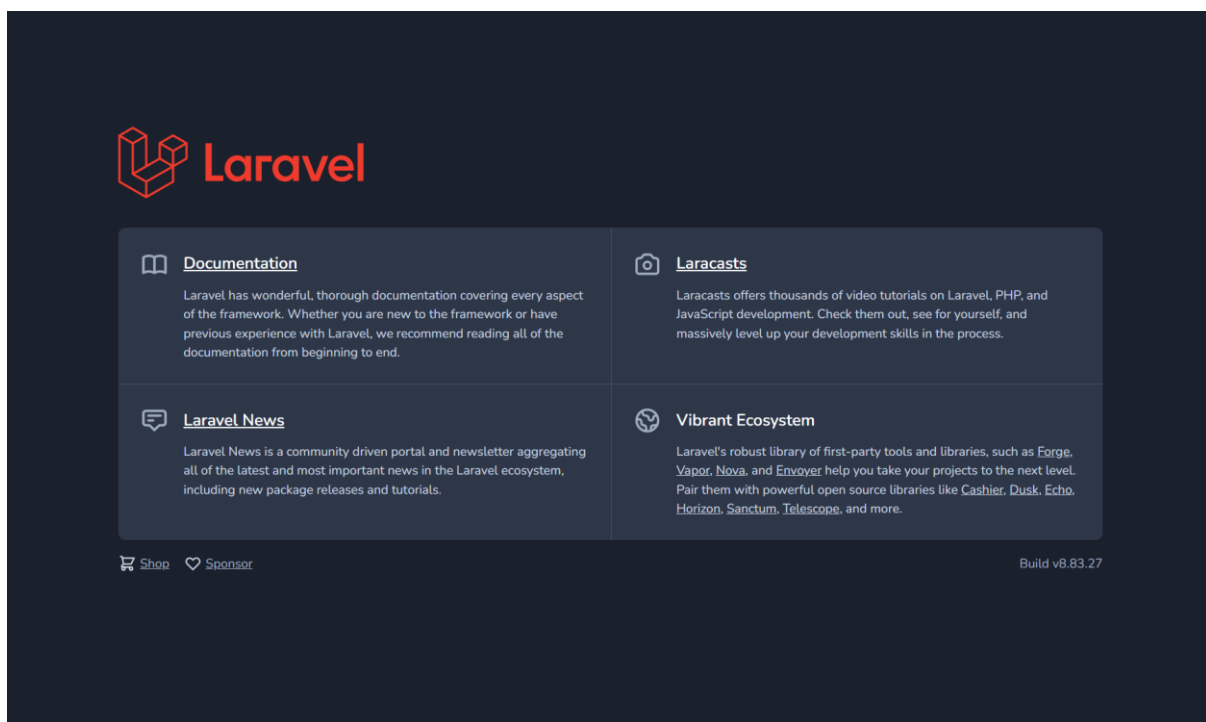
```
php artisan serve
```

Sau khi khởi chạy thành công, trong terminal của VSCode sẽ hiện lên các dòng lệnh sau:

```
nguye@THUCPC MINGW64 /d/bktel (main)
$ php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Fri Aug 4 15:24:52 2023] PHP 8.0.28 Development Server (http://127.0.0.1:8000) started
█
```

Hình 1. Kết quả sau khi chạy dự án hiển thị trên terminal

Sau khi truy cập <http://127.0.0.1:8000> trong Chrome ta được giao diện của Laravel:



Hình 2. Kết quả sau khi chạy dự án hiển thị trên Chrome

2. Kết quả đạt được sau khi thực hiện Task 0

Biết cách sử dụng công cụ quản lý source code Git, Github

Hiểu cách xây dựng dự án với Laravel, Vue

TASK 1. LARAVEL MIX

1. Các bước thực hiện

Tiến hành tạo lần lượt routes, views, controller,.. Nhưng ở những bản Laravel mới gần đây, ta có thể sử dụng scaffold để tạo ra tất cả những thứ trên chỉ bằng một câu lệnh:

```
php artisan make:auth
```

Sau khi thực hiện lệnh trong terminal thì chức năng đăng nhập, đăng kí tài khoản đã được cài đặt trong máy. Chúng ta tìm đến các file trong thư mục Resource/View/Auth. Ở đây chứa sẵn cho ta 3 file gồm login.blade.form, register.blade.form, verify.blade.form.

Chúng ta sẽ tiến hành lập trình giao diện đăng nhập, đăng kí tại các file này. Ở đây chúng ta sẽ tạo form đăng nhập có dạng như sau:

```
<!doctype html>

<head>

<meta name="csrf-token" content="{{ csrf_token() }}">

....

</head>

<form method="POST" action="{{ route('login') }}">

        @csrf

.....

</form>
```

Đối với form đăng kí cũng sẽ được viết tương tự chỉ khác:

```
<form method="POST" action="{ { route('register') } }">
```

Tiếp theo ta tạo Route cho chức năng. Nhưng do dùng dòng lệnh php artisan make:Auth nên ứng dụng sẽ tự động tạo Route sẵn cho chức năng. Ta dùng lệnh ở dưới để kiểm tra Route đã có hay chưa:

```
php artisan route:list
```

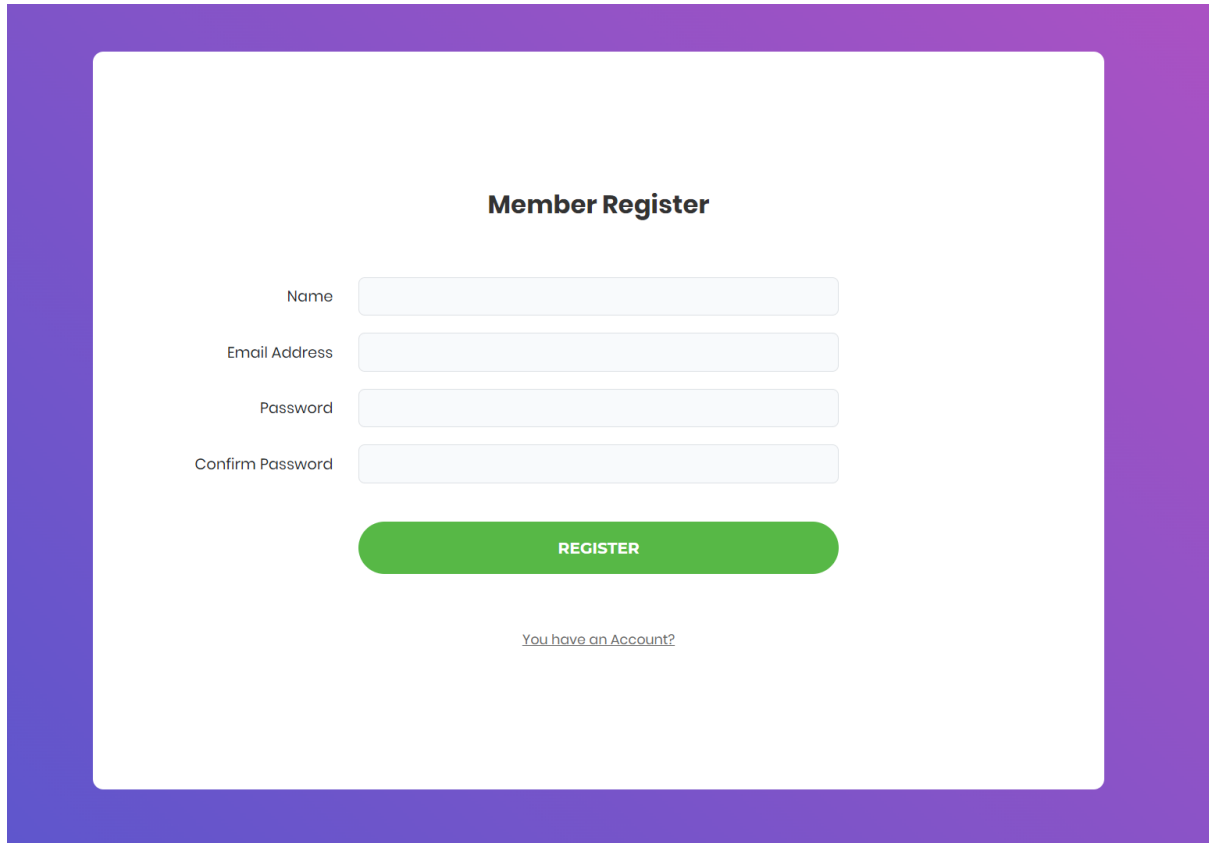
Kết quả trả về tại terminal:

POST	login		App\Http\Controllers\Auth\LoginController@login
POST	logout	logout	App\Http\Controllers\Auth\LoginController@logout
GET HEAD	password/confirm	password.confirm	App\Http\Controllers\Auth\ConfirmPasswordController@showConfirmForm
POST	password/confirm		App\Http\Controllers\Auth\ConfirmPasswordController@confirm
POST	password/email	password.email	App\Http\Controllers\Auth\ForgotPasswordController@sendResetLinkEmail
GET HEAD	password/reset	password.request	App\Http\Controllers\Auth\ForgotPasswordController@showLinkRequestForm
POST	password/reset	password.update	App\Http\Controllers\Auth\ResetPasswordController@reset
GET HEAD	password/reset/{token}	password.reset	App\Http\Controllers\Auth\ResetPasswordController@showResetForm
GET HEAD	register	register	App\Http\Controllers\Auth\RegisterController@showRegistrationForm
POST	register		App\Http\Controllers\Auth\RegisterController@register

Sau đó truy cập <http://127.0.0.1:8000/login> trong Chrome ta được giao diện đăng nhập:

Hình 3. Kết quả sau giao diện đăng nhập hiển thị trên Chrome

Tiếp tục truy cập <http://127.0.0.1:8000/register> trong Chrome ta được giao diện đăng nhập:

The image shows a web browser window with a purple border. Inside, there is a white rectangular area containing a registration form. The form is titled "Member Register" in bold black text. Below the title, there are four input fields with labels to their left: "Name", "Email Address", "Password", and "Confirm Password". Each input field is a light gray rectangle. Below these fields is a green rounded rectangular button with the word "REGISTER" in white capital letters. At the bottom of the form area, there is a link that says "You have an Account?" in a smaller, blue, underlined font.

Hình 4. Kết quả sau giao diện đăng kí hiển thị trên Chrome

2. Kết quả đạt được sau khi hoàn thành Task 1

Biết cách tạo chức năng đăng kí, đăng nhập hoàn chỉnh

TASK 2. TÌM HIỂU LÝ THUYẾT VÀ THỰC HÀNH

1. Kiến trúc MVC trong lập trình Web

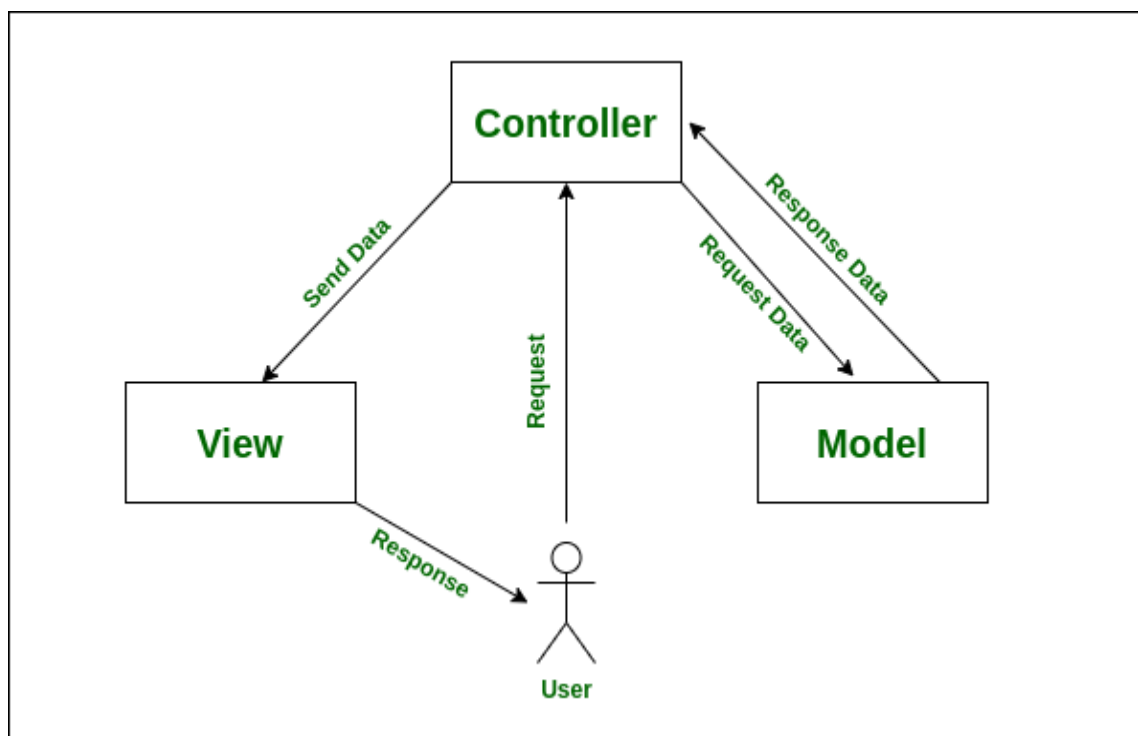
Mô hình Model-View-Controller (MVC) là một mẫu kiến trúc phân tách một ứng dụng thành ba thành phần logic chính Model, View và Controller. Do đó viết tắt MVC. Mỗi thành phần kiến trúc được xây dựng để xử lý khía cạnh phát triển cụ thể của một ứng dụng. MVC tách lớp logic nghiệp vụ và lớp hiển thị ra riêng biệt. Ngày nay, kiến trúc MVC đã trở nên phổ biến để thiết kế các ứng dụng web cũng như ứng dụng di động.

MVC quan trọng bao gồm:

Model: Nó bao gồm tất cả dữ liệu và logic liên quan của nó.

View: Trình bày dữ liệu cho người dùng hoặc xử lý tương tác của người dùng.

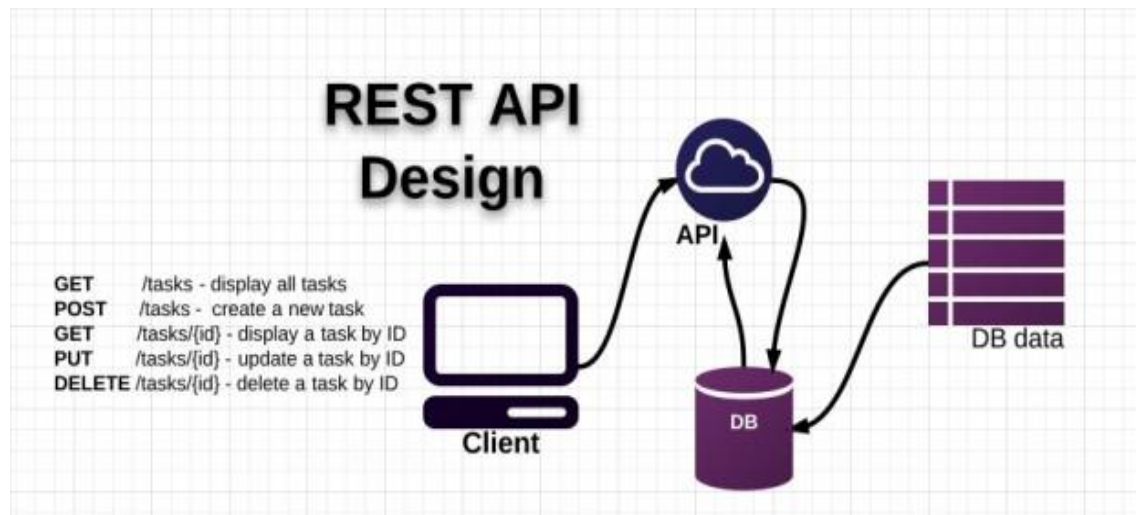
Controller: Là phần quan trọng nhất trong mô hình, nó liên kết phần Model và View.



Hình 5. Sơ đồ hoạt động của mô hình MVC

2. Khái niệm về RESTful API

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 6. Mô tả cách hoạt động của RESTful API

Diễn giải các thành phần

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.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

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.

3. Tìm hiểu về cơ sở dữ liệu quan hệ (MySQL)

a) Bảng (Table) là gì

Là đối tượng được Database sử dụng để tổ chức và lưu trữ dữ liệu.

Mỗi Table trong Database có thể liên kết với một hoặc nhiều Table khác, ở một hoặc nhiều thuộc tính.

Mỗi Table bao gồm:

Record: Là các dòng, hay còn gọi là các BẢN GHI. Biểu diễn cho một thực thể (ứng với một đối tượng)

Ví dụ: Bản ghi số 3 thể hiện mọi thông tin liên quan đối tượng Nguyễn Sơn Tùng như MaSV, TenSV, Khoa....

Column: Là các cột hay còn gọi là các TRƯỜNG THUỘC TÍNH. Biểu diễn cho một tính chất của thực thể.

Ví dụ: Trường TenSV lưu trữ tên của các sinh viên được lưu trữ trong Table SINHVIEN

a) Khóa chính là gì

Khóa chính (hay ràng buộc khóa chính) được sử dụng để định danh duy nhất mỗi record trong table của cơ sở dữ liệu.

Ngoài ra, nó còn dùng để thiết lập quan hệ 1-n (hay ràng buộc tham chiếu) giữa hai table trong cơ sở dữ liệu.

Dữ liệu (value) của field khóa chính phải có tính duy nhất. Và không chứa các giá trị Null.

Mỗi table nên chỉ có một khóa chính, khóa chính có thể tạo ra từ nhiều field của table.

b) Khóa ngoại là gì

Khóa ngoại của một table được xem như con trỏ trỏ tới khóa chính của table khác.

4. Thực hành và triển khai lý thuyết cho dự án

Tạo 1 table sinhvien với các fields như sau (kiểu dữ liệu cho từng field các bạn tự tham khảo và quyết định sao cho phù hợp nhất).

Tạo SinhvienController tương ứng CRUD như sau.

```
<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateSinhviensTable extends Migration

{

    * Run the migrations.

    */

    public function up():void

    {

        Schema::create('sinhvien', function (Blueprint $table) {
```



```
$table->id();

$table->string('Firstname');

$table->string('Lastname');

$table->string('Studentcode');

$table->string('Department');

$table->string('Faculty');

$table->string('Address');

$table->string('Phone');

$table->string('Note');

$table->timestamps();

});

}

* Reverse the migrations

*/

public function down():void

{

    Schema::dropIfExists('sinhviens');

}

}
```

Tạo Restful API trên laravel Tạo restful API CRUD (create, read, update, delete)
cho table sinhviens

```
Route::group(['prefix' => 'sinhvien'], function() {

    Route::get('/create', [App\Http\Controllers\SinhvienController::class, 'create'])->name('sinhvien.create');

    Route::put('/{id}', [App\Http\Controllers\SinhvienController::class, 'update'])->name('sinhvien.update');

    Route::get('/index', [App\Http\Controllers\SinhvienController::class, 'index'])->name('sinhvien.index');

    Route::get('/{id}', [App\Http\Controllers\SinhvienController::class, 'edit'])->name('sinhvien.edit');

    Route::post('/index', [App\Http\Controllers\SinhvienController::class, 'store'])->name('sinhvien.store');

    Route::delete('/{id}', [App\Http\Controllers\SinhvienController::class, 'destroy'])->name('sinhvien.destroy');

});
```

5. Sử dụng PostmanUI để test API

Postman là một App Extensions, nhằm thực hiện test API, nhất là REST, hỗ trợ tất cả các

phương thức HTTP (GET, POST, PUT, DELETE, OPTIONS, HEAD ...)

Download Postman tại <https://www.postman.com/downloads/> và chạy lệnh sau trên terminal

```
php artisan make:command DevPostman
```

6. Kết quả:

Hiểu cách tạo RESTful API trong Laravel

Hiểu cách thao tác CRUD

TASK 3. VALIDATION TRONG LARAVEL

1. Các bước thực hiện

Validate dữ liệu là kiểm tra dữ liệu đầu vào với các điều kiện trước khi thực hiện xử lý dữ liệu.

Validation là một công cụ do Laravel cung cấp dùng để validate (kiểm tra) dữ liệu request được gửi lên trước khi nó được xử lý Validation trong laravel.

Trong trường hợp dữ liệu phức tạp và nhiều trường cần được kiểm tra Laravel có hỗ trợ một lớp gọi là custom request classes that contain validation logic. Lớp này làm nhiệm vụ kiểm soát tính hợp lệ của dữ liệu đầu vào. Sau khi một request được gửi lên từ phía client thì request sẽ được validate dữ liệu bởi class form request trước khi được controller sử dụng

Cú pháp tạo một Form Request Validation

```
php artisan make:request NameRequest
```

Thực hiện validate dữ liệu với nhiệm vụ sau

Thực hiện validate email, password cho form register với điều kiện

- email: phải có dạng xyz@hcmut.edu.vn
- password: 8 ký tự, có cả chữ và số (hoa thường đều được)

Validate với form đăng kí: dùng hàm validate được cung cấp bởi class Request

```
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'ends_with:hcmut.edu.vn',
'unique:users'],
```

```
'password' => ['required', 'regex:/^(?=.*{3,})(?=.*[a-zA-Z])(?=.*[0-9])(?=.*[\d\x])(?=.*[!$#%]).*$/','min:8', 'confirmed'],

    ];

}
```

2. Kết quả:

Hiểu cách tạo Validation trong Laravel

Hiểu được cách kiểm tra dữ liệu đầu vào trong laravel

TASK 4. TÌM HIỂU VỀ SEEDER

1. Lý thuyết chung

Seeder là class cho phép các bạn xử lý dữ liệu trong database. Class này sẽ hỗ trợ các bạn tạo ra data test, thay đổi cập nhật dữ liệu khi cần thiết.

Lệnh tạo một seeder trong laravel

```
php artisan make:seeder NameSeeder
```

Lệnh thực thi seeder

```
php artisan db:seed --class=UserSeeder
```

2. Các bước thực hiện

Tạo thêm 1 table roles (id, name, note) và Table này tạo 1 mối quan hệ giữa roles và users là 1-to-many có nghĩa một roles có thể có nhiều users

Tạo seeder chứa các roles sau:

id: 1 - name: Administrator

id: 2 - name: Supervisor

id: 3 - name: Teacher

id: 4 - name: Student

Thêm 1 column role_id trong table user (set role_id là nullable, mặc định role student khi đăng ký user mới)

Thực hiện tạo seeder cho user admin với tài khoản mặc định

email: bmv@hcmut.edu.vn

name: BMVT

pass: Bmv@hcmut

role_id: 1

Cập nhật lại roles cho các student đã tạo trong db trước đó (nếu có) thông qua column role_id

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Facades\Hash;

class RoleExample extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        DB::table('roles') ->insert([

            ['name' =>'admin'],

            ['name' =>'supervisor'],

            ['name' =>'teacher'],

            ['name' =>'student'],
```

```
]);

DB::table('users')->insert([

    'email' =>'bmvmt@hcmut.edu.vn',

    'name' => 'BMVT',

    'password' => Hash::make('Bmvmt@hcmut'),

    'role_id' => 1 ]);

}

}
```

2. Kết quả:

Hiểu cách sử dụng Seeder trong Laravel

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: Filter rows:

Extra options

				id	name	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	admin	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	supervisor	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	teacher	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	student	NULL	NULL

Hình 7. Kết quả sau khi tạo Seeder

TASK 5. THỰC HIỆN TRANG CẬP NHẬT THÔNG TIN SINH VIÊN**1. Các bước thực hiện**

Tạo hệ sơ sở dữ liệu có tên create_sinhviens_table

```
<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateSinhviensTable extends Migration
{
    public function up():void
    {
        Schema::create('sinhviens', function (Blueprint $table) {

            $table->id();

            $table->string('Firstname');

            $table->string('Lastname');

            $table->string('Studentcode');

            $table->string('Department');

            $table->string('Faculty');

            $table->string('Address');

            $table->string('Phone');

            $table->string('Note');
```



```

        $table->timestamps();

    });

}

public function down():void

{

    Schema::dropIfExists('sinhviens');

}

}

```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id 🔑	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 Firstname	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 Lastname	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 Studentcode	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 Department	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	6 Faculty	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	7 Address	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	8 Phone	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	9 Note	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	10 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	11 updated_at	timestamp			Yes	NULL			Change Drop More

Hình 8. Kết quả sau khi tạo `create_sinhviens_table`

Tạo controller và giao diện cập nhật thông tin sinh viên. Giao diện cập nhật thông tin sinh viên có dạng Vuejs. Khi sử dụng file Vuejs làm giao diện cập nhật thông tin sinh viên thì phải khai báo trong file `app.js` ở `resource/js/app.js`.

Ví dụ cho khai báo trong `app.js` với file `ExampleComponent.vue`

```

Vue.component('example-component',
require('./components/ExampleComponent.vue').default);

```

Khai báo các chức năng khai báo thông tin sinh viên trong Route nằm ở route/web.php

```
<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\SinhvienController;

use App\Http\Controllers\TeacherController;

use App\Http\Controllers\HomeController;

use App\Http\Controllers\SubjectController;

use App\Http\Controllers\ImportController;

use App\Http\Controllers\TeacherToSubjectController;

use App\Http\Controllers\ReportController;

Route::group(['prefix' => 'sinhvien'], function() {

    Route::get('/create', [App\Http\Controllers\SinhvienController::class, 'create'])->name('sinhvien.create');

    Route::put('/{id}', [App\Http\Controllers\SinhvienController::class, 'update'])->name('sinhvien.update');

    Route::get('/index', [App\Http\Controllers\SinhvienController::class, 'index'])->name('sinhvien.index');

    Route::get('/{id}', [App\Http\Controllers\SinhvienController::class, 'edit'])->name('sinhvien.edit');

    Route::post('/index', [App\Http\Controllers\SinhvienController::class, 'store'])->name('sinhvien.store');

    Route::delete('/{id}', [App\Http\Controllers\SinhvienController::class, 'destroy'])->name('sinhvien.destroy');
```

$\}) ;$

Please Fill & Click Submit

First Name

First Name

Last Name

Last Name

Student Code

Student Code

Department

Department

Faculty

Faculty

Address

Address

Phone

Phone

Note

Note

Submit

Hình 9. Giao diện cập nhật thông tin sinh viên

Employee Information												
				▼ id	Firstname	Lastname	Studentcode	Department	Faculty	Address	Phone	Note
<input type="checkbox"/>				8	Daniel Davis	Joseph Robertson	1910000	GEOPET	UI-UX	Di An Binh Duong Viet Nam	550-677-6639	1
<input type="checkbox"/>				9	Peter Jackson	Darin Rose	1910001	SIM	Game development	Di An Binh Duong Viet Nam	396-338-9038	1
<input type="checkbox"/>				10	Chelsey Hogan	Daniel Fox	1910002	SIM	Game development	Di An Binh Duong Viet Nam	876-454-0137	1
<input type="checkbox"/>				11	Connie Villanueva	Steven Vasquez	1910003	CHE	Embedded c++	Di An Binh Duong Viet Nam	455-736-6604	1
<input type="checkbox"/>				12	Amanda Cole	Terri Cook	1910004	CSE	Embedded c++	Di An Binh Duong Viet Nam	954-865-6033	1
<input type="checkbox"/>				13	Jenna Thomas	Phillip Hill	1910005	SIM	System engineer	Di An Binh Duong Viet Nam	935-570-5382	1
<input type="checkbox"/>				14	Theresa King	Jordan Ellis	1910006	D-DT	Web development	Di An Binh Duong Viet Nam	548-214-1452	1
<input type="checkbox"/>				15	Sabrina Wilson	Robert Adams	1910007	FTE	Data engineer	Di An Binh Duong Viet Nam	728-834-2393	1
<input type="checkbox"/>				16	Jessica Brown	Alex Daniel	1910008	SIM	UI-UX	Di An Binh Duong Viet Nam	671-715-6339	1
<input type="checkbox"/>				17	Fred Gordon	Kristine Norris	1910009	GEOPET	UI-UX	Di An Binh Duong Viet Nam	109-465-1040	1
<input type="checkbox"/>				18	Brian Maldonado	Rebecca Thomas	1910010	D-DT	System engineer	Di An Binh Duong Viet Nam	432-040-2971	1
<input type="checkbox"/>				19	Jennifer Williams	Emily Martinez	1910011	CHE	Data engineer	Di An Binh Duong Viet Nam	149-870-5596	1
<input type="checkbox"/>				20	Christopher Maldonado	Gary Chambers	1910012	CHE	Web development	Di An Binh Duong Viet Nam	424-338-7585	1

Hình 10. Kết quả sau khi cập nhật thông tin sinh viên

2. Kết quả:

Thực hành lại cách dùng form, xử lý form trong Laravel và VueJs.

TASK 6. TẠO TRANG THÔNG TIN CHO GIÁO VIÊN

1. Các bước thực hiện

Tạo hệ cơ sở dữ liệu có tên create_teachers_table

```
<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateTeachersTable extends Migration
{
    public function up(): void
    {
        Schema::create('teachers', function (Blueprint $table) {

            $table->id();

            $table->string('last_name');

            $table->string('first_name');

            $table->string('teacher_code');

            $table->string('department');

            $table->string('faculty');

            $table->string('address');

            $table->string('phone');

            $table->string('note')->nullable();
```

```
        $table->timestamps();

    });

}

public function down(): void
{
    Schema::dropIfExists('teachers');
}
}
```

Tương tự với task 5 thì ta cũng sẽ khai báo tiếp tục ở route cho các chức năng cho trang thông tin giáo viên

```
Route::group(['prefix' => 'teacher'], function() {

    Route::get('/create', [App\Http\Controllers\TeacherController::class, 'create'])->name('teacher.create');

    Route::put('/{id}', [App\Http\Controllers\TeacherController::class, 'update'])->name('teacher.update');

    Route::get('/index', [App\Http\Controllers\TeacherController::class, 'index'])->name('teacher.index');

    Route::get('/{id}', [App\Http\Controllers\TeacherController::class, 'edit'])->name('teacher.edit');

    Route::post('/index', [App\Http\Controllers\TeacherController::class, 'store'])->name('teacher.store');

    Route::delete('/{id}', [App\Http\Controllers\TeacherController::class, 'destroy'])->name('teacher.destroy');

});
```

Please Fill & Click Submit

First Name

Last Name

Teacher Code

Department

Faculty

Address

Phone

Note

SUBMIT

Hình 11. Giao diện cập nhật thông tin giáo viên

				id	last_name	first_name	teacher_code	department	faculty	address	phc
<input type="checkbox"/>				4	Vo Ngoc	Hai	1940000	FME	Web development	Cau Giay Ha Noi Viet Nam	523
<input type="checkbox"/>				5	Le Huu	Tham	1940001	CSE	Data engineer	Cau Giay Ha Noi Viet Nam	163
<input type="checkbox"/>				6	Nguyen Duc	Tham	1940002	SIM	UI-UX	Cau Giay Ha Noi Viet Nam	744
<input type="checkbox"/>				7	Vo Que	Quy	1940003	D-DT	System engineer	Cau Giay Ha Noi Viet Nam	811
<input type="checkbox"/>				10	Bui Quang	Quy	1940006	GEOPET	Embedded c++	Cau Giay Ha Noi Viet Nam	450
<input type="checkbox"/>				11	Bui Quang	Hau	1940007	GEOPET	UI-UX	Cau Giay Ha Noi Viet Nam	386
<input type="checkbox"/>				12	Dao Van	Linh	1940008	FCE	Data engineer	Cau Giay Ha Noi Viet Nam	276
<input type="checkbox"/>				13	Trinh Van	Thien	1940009	FTE	Data engineer	Cau Giay Ha Noi Viet Nam	678
<input type="checkbox"/>				14	Bui Quang	Ngoc	1940010	SIM	UI-UX	Cau Giay Ha Noi Viet Nam	599
<input type="checkbox"/>				15	Vo Ngoc	Tham	1940011	FME	Embedded c++	Cau Giay Ha Noi Viet Nam	549
<input type="checkbox"/>				16	Le Huu	Thanh	1940012	CHE	UI-UX	Cau Giay Ha Noi Viet Nam	954
<input type="checkbox"/>				17	Tran Van	Hao	1940013	CSE	Web development	Cau Giay Ha Noi Viet Nam	720
<input type="checkbox"/>				18	Nguyen Duc	Son	1940014	GEOPET	Data engineer	Cau Giay Ha Noi Viet Nam	260

Hình 12. Kết quả sau khi cập nhật thông tin giáo viên

2. Kết quả:

Thực hành lại cách dùng form, xử lý form trong Laravel và VueJs.

TASK 7, 8, 9. JOB VÀ IMPORT DANH SÁCH GIẢNG VIÊN**1. Các bước thực hiện**

Một hàng đợi (queue) là một danh sách những việc cần làm (job) được quản lý theo thứ tự.

```
php artisan make:job NameJob
```

Laravel Excel: thư viện xử lý file csv, xlsx tương tự như PHPExcel, PhpSpreadsheet, hỗ trợ nhiều chức năng gồm:

- Import file excel thành collection.
- Export file excel.
- Import nhiều file cùng một lúc.
- Có hỗ trợ catching.
- Hỗ trợ chunk và queues importer.
- Đọc file excel trong chunk.

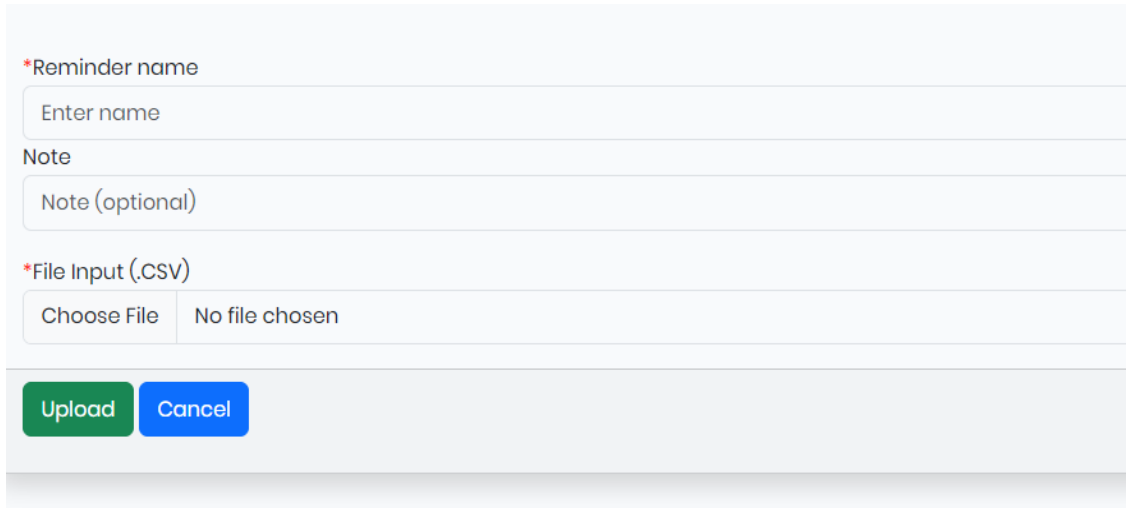
Thực hiện import danh sách giảng viên từ file teachers.csv

Tạo một class TeachersImport thực hiện import danh sách giảng viên

```
php artisan make:import TeachersImport
```

Tạo giao diện import bao gồm

- Input - name: người dùng điền tên gọi nhớ cho lần import này
- Input - selected file: chọn file csv
- Button - import: nút nhấn import



The screenshot shows a web form for importing data. It has three main sections: 1. A text input field labeled '*Reminder name' with a placeholder 'Enter name'. 2. A text input field labeled 'Note' with a placeholder 'Note (optional)'. 3. A file input section labeled '*File Input (.CSV)' containing a 'Choose File' button and the text 'No file chosen'. At the bottom, there are two buttons: a green 'Upload' button and a blue 'Cancel' button.

Hình 13. Giao diện Import danh sách giảng viên

Thực hiện import danh sách giảng viên từ file teachers.csv

Tạo một class TeachersImport thực hiện import danh sách giảng viên, thực hiện tương tự cho task 8 và task 9. Thay vì class là TeacherImport, ta đổi thành SinhviensImport, SubjectsImport.

```
php artisan make:import TeachersImport
```

Nội dung file

```
<?php

namespace App\Imports;

use App\Models\Teacher;

use App\Models\User;

use Illuminate\Support\Collection;

use Illuminate\Support\Facades\Hash;

use Maatwebsite\Excel\Concerns\ToCollection;

use Maatwebsite\Excel\Concerns\WithStartRow;
```



```
class TeachersImport implements ToCollection, WithStartRow
{
    public function startRow(): int
    {
        return 2;
    }

    public function collection(Collection $rows)
    {
        foreach ($rows as $row ) {

            $teacher = new Teacher();

            $teacher->last_name = $row[0];

            $teacher->first_name = $row[1];

            $teacher->teacher_code = $row[2];

            $teacher->department = $row[3];

            $teacher->faculty = $row[4];

            $teacher->address = $row[5];

            $teacher->phone = $row[6];

            $teacher->note = $row[7];

            $teacher->save();

            $user = new User();

            $user->name = $row[0]." ".$row[1];
```

```
$user->email = $row[2];

$user->password = Hash::make("Bmvt@hcmut");

$user->role_id = 3;

$user->teacher_id = $teacher->id;

$user->save();

    }

}

}
```

Khai báo ở route cho các chức năng ở file web.php

```
Route::group(['prefix' => 'import'], function() {

    Route::get('/teacher', [App\Http\Controllers\ImportController::class,
'importTeacher'])->name('importteacher.create');

    Route::post('/teacher', [App\Http\Controllers\ImportController::class,
'storeTeacher'])->name('importteacher.store');

    Route::get('/student', [App\Http\Controllers\ImportController::class,
'importSinhvien'])->name('importsinhvien.create');

    Route::post('/student', [App\Http\Controllers\ImportController::class,
'storeSinhvien'])->name('importsinhvien.store');

    Route::get('/subject', [App\Http\Controllers\ImportController::class,
'importSubject'])->name('importsubject.create');

    Route::post('/subject', [App\Http\Controllers\ImportController::class,
'storeSubject'])->name('importsubject.store');

});
```

Nội dung file ImportController

```
<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

use App\Jobs\ImportTeachersCsv;

use App\Jobs\ImportSinhviensCsv;

use App\Jobs\ImportSubjectsCsv;

use Illuminate\Http\Request;

use App\Models\Import;

use Illuminate\Support\Facades\Auth;

class ImportController extends Controller
{

    public function importTeacher() {

        return view ('import/teacher');

    }

    public function storeTeacher(Request $request) {

        $file_name = date('Ymd_His_').$request->file->getClientOriginalName();

        $file_path = storage_path('app\\data\\'.$file_name);

        $import = new Import();

        $import->name = $file_name;

        $import->path = $file_path;
```

```
$import->status = 0;

$import->created_by = Auth::user()->name;

$import->save();

// save file

$request->file->move(storage_path('app\\data\\'), $file_name);

$teacherImport = Import::latest()->first();

$teacherImport = $import;

$path = $file_path;

ImportTeachersCsv::dispatch($path, $teacherImport)->delay(10);

return response()->json('Tải file thành công, đang chờ xử lý');

}

public function importSinhvien() {

    return view ('import/student');

}

public function storeSinhvien(Request $request) {

    $file_name = date('Ymd_His_').$request->file->getClientOriginalName();

    $file_path = storage_path('app\\data\\'.$file_name);

    $import = new Import();

    $import->name = $file_name;

    $import->path = $file_path;

    $import->status = 0;

    $import->created_by = Auth::user()->name;
```

```
$import->save();

// save file

$request->file->move(storage_path('app\\data\\'), $file_name);

$sinhvienImport = $import;

$path = $file_path;

ImportsinhviensCsv::dispatch($path, $sinhvienImport)->delay(10);

return response()->json('Tải file thành công, đang chờ xử lý');

}

public function importSubject() {

    return view ('import/subject');

}

public function storeSubject(Request $request) {

    $file_name = date('Ymd_His_').$request->file->getClientOriginalName();

    $file_path = storage_path('app\\data\\'.$file_name);

    $import = new Import();

    $import->name = $file_name;

    $import->path = $file_path;

    $import->status = 0;

    $import->created_by = Auth::user()->name;

    $import->save();

    $request->file->move(storage_path('app\\data\\'), $file_name);
```

```

$subjectImport = $import;

$path = $file_path;

ImportSubjectsCsv::dispatch($path, $subjectImport)->delay(10);































return response()->json('Tải file thành công, đang chờ xử lý');

}

}

```

Extra options

		id	name	path	status	created_by	note	created_at	updated_at
<input type="checkbox"/>	  	1	20230804_142539_teachers.csv	D:\bktel\storage\app\data\20230804_142539_teachers...	1	Nguyen dang Thuc	NULL	2023-08-04 14:25:39	2023-08-04 14:25:39
<input type="checkbox"/>	  	2	20230804_142716_teachers.csv	D:\bktel\storage\app\data\20230804_142716_teachers...	2	Nguyen dang Thuc	NULL	2023-08-04 14:27:16	2023-08-04 14:28:01
<input type="checkbox"/>	  	3	20230804_142802_teachers.csv	D:\bktel\storage\app\data\20230804_142802_teachers...	1	Nguyen dang Thuc	NULL	2023-08-04 14:28:02	2023-08-04 14:28:02
<input type="checkbox"/>	  	4	20230804_142802_teachers.csv	D:\bktel\storage\app\data\20230804_142802_teachers...	1	Nguyen dang Thuc	NULL	2023-08-04 14:28:02	2023-08-04 14:28:02
<input type="checkbox"/>	  	5	20230804_143005_subjects.csv	D:\bktel\storage\app\data\20230804_143005_subjects...	2	Nguyen dang Thuc	NULL	2023-08-04 14:30:05	2023-08-04 14:30:05
<input type="checkbox"/>	  	6	20230804_143128_students.csv	D:\bktel\storage\app\data\20230804_143128_students...	1	Nguyen dang Thuc	NULL	2023-08-04 14:31:28	2023-08-04 14:31:28
<input type="checkbox"/>	  	7	20230804_143218_students.csv	D:\bktel\storage\app\data\20230804_143218_students...	1	Nguyen dang Thuc	NULL	2023-08-04 14:32:18	2023-08-04 14:32:18
<input type="checkbox"/>	  	8	20230804_143235_test.xlsx	D:\bktel\storage\app\data\20230804_143235_test.xls...	1	Nguyen dang Thuc	NULL	2023-08-04 14:32:35	2023-08-04 14:32:35
<input type="checkbox"/>	  	9	20230804_143325_students.csv	D:\bktel\storage\app\data\20230804_143325_students...	1	Nguyen dang Thuc	NULL	2023-08-04 14:33:25	2023-08-04 14:33:25
<input type="checkbox"/>	  	10	20230804_143421_students.csv	D:\bktel\storage\app\data\20230804_143421_students...	1	Nouven dang Thuc	NULL	2023-08-04 14:34:21	2023-08-04 14:34:21


Hình 14. Kết quả sau khi Import danh sách giảng viên

2. Kết quả:

Thực hành lại thao tác với Laravel Excel và cách sử dụng job

TASK 10. TẠO DANH SÁCH GIẢNG VIÊN GIẢNG DẠY

Tạo form Vuejs CreateTeacherToSubject



Hình 15. Giao diện tạo mới giảng viên giảng dạy

Nội dung file Controller

```
class TeacherToSubjectController extends Controller
{
    public function index()
    {
        $teachertosubject = TeacherToSubject::paginate(5);

        return view('teachertosubject.index', compact('teachertosubject'))->with('i',(request()->input('page',1)-1)*5);
    }
}
```

```
public function create()

{

    $teachers = Teacher::all();

    $subjects = Subject::all();

    return view ('teachertosubject.create', [

        'teachers' => $teachers,

        'subjects' => $subjects

    ]);

}

public function store(Request $request)

{

    $teacherToSubject = new TeacherToSubject($request->all());

    $teacherToSubject->save();

    return response()->json($teacherToSubject);

}

public function fetchAllTeacher()

{

    return response()->json([

        'teachers' => Teacher::all()
```



```
    ]);  
  
}  
  
public function fetchAllSubject()  
  
{  
  
    return response()->json([  
  
        'subjects' => Subject::all()  
  
    ]);  
  
}
```

Kết quả trả về dưới dạng json lấy từ table teacher_to_subjects

```
{teacher_id: "1", subject_id: "2", semester: "1", year: "2002",...}  
  created_at: "2023-08-10T17:04:48.000000Z"  
  id: 3  
  semester: "1"  
  subject_id: "2"  
  teacher_id: "1"  
  updated_at: "2023-08-10T17:04:48.000000Z"  
  year: "2002"
```

Hình 16. Giao diện trả về khi đăng kí thành công

TASK 11. TẠO BIỂU MẪU SINH VIÊN UPLOAD FILE

1
2
1
2002

SEARCH

ID	Teacher ID	Subject ID	Semester	Year	
1	1	2	1	2002	✓
2	1	2	1	2002	✓
3	1	2	1	2002	✓

Choose File No file chosen

title

UPLOAD

Hình 17. Giao diện sinh viên upload file

Nội dung file Controller

```
class SetmarkController extends Controller
{
    public function ShowformUploadMark()
    {
        return view ('/setmark/index');
    }

    public function SearchReport(Request $request)
    {
        $subject_id=$request -> subject_id;
```

```
$sinhvien_id=$request -> sinhvien_id;

$teacher_id= auth()->user()->teacher_id;


// $subject_id = DB::table('subjects')->where('code', $subject_code)->value('id') ;

$semester = $request -> semester;

$data = DB::table('reports')->join('teacher_to_subjects',
'teacher_to_subjects.id','=', 'reports.teacher_to_subjects_id')

->select('*', 'reports.id as report_id')

->where('sinhvien_id', $sinhvien_id)

// ->where('subject_id', $subject_id)

->where('teacher_id', $teacher_id)

->get();


return response()->json($data);
}

public function SetMarkReport(Request $request)
{

    $report_id= $request -> report_id;

    $report = Report::find($report_id);

    $report->mark= $request -> mark;
```

```
$report->save();

$data = $report;

return response()->json($data);

}

public function DownloadfileReport(Request $request)
{

    $report_id= $request -> report_id;

    $report = Report::find($report_id);

    $path = storage_path($report->path);

    // return response()->json($pathToFile);

    $name = $report -> title;

    $headers = array(

        'Content-Type: application/pdf',

    );

    return response()->download($path);

}

}
```

1
2
1
2002

SEARCH

ID	Teacher ID	Subject ID	Semester	Year
1	1	2	1	2002
2	1	2	1	2002
3	1	2	1	2002

Choose File TTNT_223 (2).pdf
nop bai
UPLOAD

Hình 18. Kết quả được lưu sau khi nộp bài

Extra options

	id	sinhvien_id	teacher_to_subjects_id	title	path	mark	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	5	1	nop bai	app/reports/1691689226.TTNT_223 (2).pdf	NULL	2023-08-10 17:40:26	2023-08-10 17:40:26

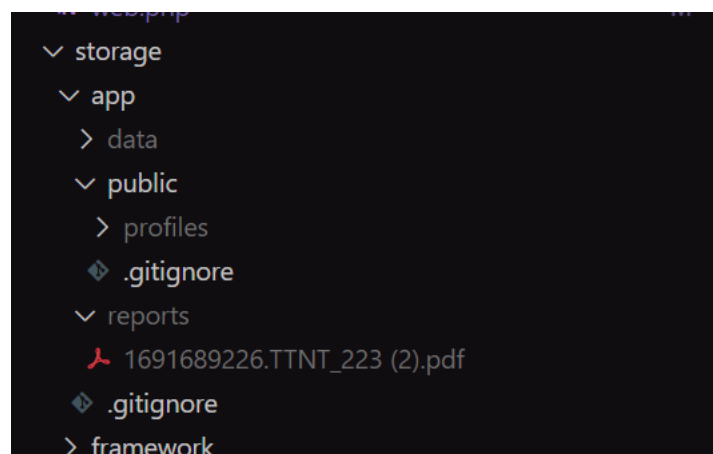
↑ ☐ Check all With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all Number of rows: 25 Filter rows: Search this table

Query results operations

☐ Print ☐ Copy to clipboard ☐ Export ☐ Display chart ☐ Create view

Hình 19. Kết quả được lưu sau khi nộp bài ở phpMyadmin



Hình 20. File được lưu ở storage/app/public/reports

TASK 12. TẠO BIỂU MẪU GIÁO VIÊN CHẤM ĐIỂM

The interface consists of three main components:

- Upload File Mark Form:** Contains four input fields with values 2, 2002, 1, and 5, and a pink 'SEARCH' button.
- Table:** A table with 8 columns: STT, Sinhvien_id, Subject_id, Teacher_id, Title, Note, Mark, and Download. It contains one row of data.
- Form to Set Mark:** Contains two input fields labeled 'STT' and 'Mark', and a pink 'SET MARK' button.

STT	Sinhvien_id	Subject_id	Teacher_id	Title	Note	Mark	Download
1	5	2	1	nop bai			Download

Hình 21. Giao diện chấm điểm giáo viên

Nội dung file Controller

```
class ExportMarkController extends Controller
{
    public function FormExportFileMark()
    {
        return view ('/exportmark/index');
    }

    public function SearchAllReport(Request $request)
    {
        $subject_id=$request -> subject_id;
```

```
$teacher_id= auth()->user()->teacher_id;

// $subject_id = DB::table('subjects')->where('code', $subject_code)->value('id') ;

$semester = $request -> semester;

$data = DB::table('reports')->join('teacher_to_subjects',
'teacher_to_subjects.id','=', 'reports.teacher_to_subjects_id')

->join('teachers',
'teachers.id','=', 'teacher_to_subjects.teacher_id')

->join('subjects',
'subjects.id','=', 'teacher_to_subjects.subject_id')

->join('sinhvien', 'sinhvien.id','=', 'reports.sinhvien_id')

->select('*', 'reports.id as report_id','teachers.first_name as
teacher_first_name')

->where('subject_id', $subject_id)

->where('teacher_id', $teacher_id)

->get();

return response()->json($data);

}

public function ExportFileMarkCsv(Request $request)

{

    $report_id = $request -> report_id;
```

```
$subject_id=$request -> subject_id;

$teacher_id= auth()->user()->teacher_id;

$semester = $request -> semester;

$data = DB::table('reports')->join('teacher_to_subjects',
'teacher_to_subjects.id','=', 'reports.teacher_to_subject_id')

->join('teachers',
'teachers.id','=', 'teacher_to_subjects.teacher_id')

->join('subjects',
'subjects.id','=', 'teacher_to_subjects.subject_id')

->join('sinhviens', 'sinhviens.id','=', 'reports.sinhvien_id')

->select('*', 'reports.id as report_id','teachers.first_name as
teacher_first_name',)

-
>select('teacher_to_subjects.semester','teacher_to_subjects.year',

'teachers.id as teacher_id','teachers.first_name as
teacher_name',

'subjects.id as subject_id','subjects.name',

'sinhviens.id as sinhvien_id','sinhviens.first_name as
sinhvien_name',

'reports.mark'

)

//->where('subject_id', $subject_id)

->where('teacher_id', $teacher_id)
```



```
// ->where('report_id', $report_id)

->get();

return response()->json($data);

}

}
```

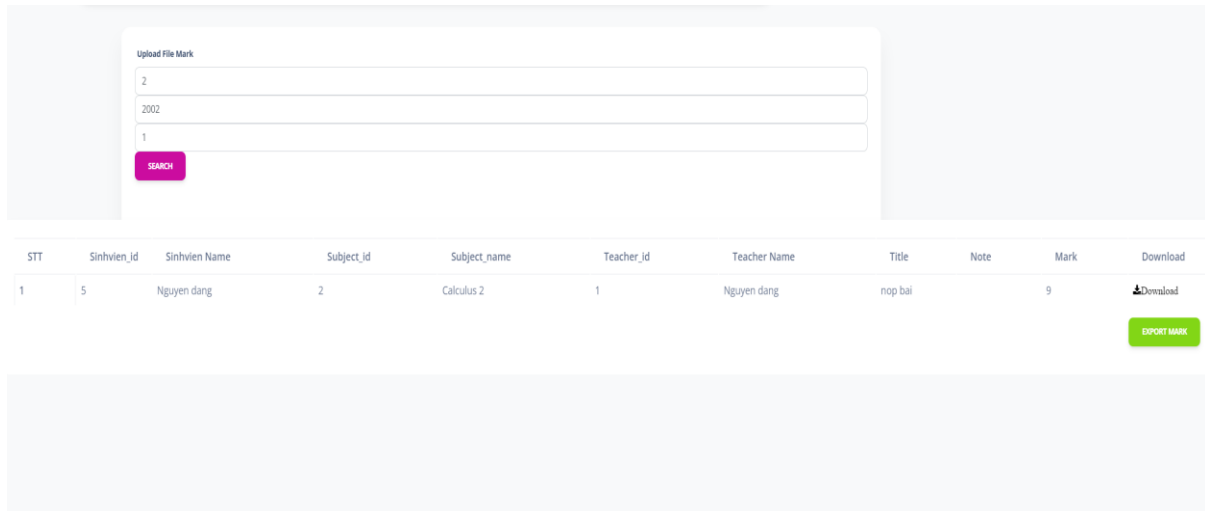
Extra options

	id	sinhvien_id	teacher_to_subjects_id	title	path	mark	created_at	updated_at
<input type="checkbox"/>	1	5	1	nop bai	app/reports/1691689226.TTNT_223 (2).pdf	9.00	2023-08-10 17:40:26	2023-08-10 17:54:08

☐ Check all With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all Number of rows: 25 Filter rows:

Hình 22. Kết quả chấm điểm cho sinh viên

TASK 13. EXPORT BẢNG ĐIỂM


The screenshot shows a web interface for uploading and viewing marks. At the top, there is a form titled 'Upload File Mark' with three input fields containing the values '2', '2002', and '1'. Below these fields is a pink 'SEARCH' button. Below the form is a table with the following data:

STT	Sinhvien_id	Sinhvien Name	Subject_id	Subject_name	Teacher_id	Teacher Name	Title	Note	Mark	Download
1	5	Nguyen dang	2	Calculus 2	1	Nguyen dang	nop bai		9	Download

Below the table, there is a green 'EXPORT MARK' button.

Hình 23. Giao diện export bảng điểm

Nội dung file controller

```

class ExportMarkController extends Controller
{
    public function FormExportFileMark()
    {
        return view ('/exportmark/index');
    }

    public function SearchAllReport(Request $request)
    {
        $subject_id=$request -> subject_id;

        $teacher_id= auth()->user()->teacher_id;

        // $subject_id  =  DB::table('subjects')->where('code',  $subject_code)-
        >value('id') ;
    }
}

```

```
$semester = $request -> semester;

$data = DB::table('reports')->join('teacher_to_subjects',
'teacher_to_subjects.id','=', 'reports.teacher_to_subjects_id')

->join('teachers',
'teachers.id','=', 'teacher_to_subjects.teacher_id')

->join('subjects',
'subjects.id','=', 'teacher_to_subjects.subject_id')

->join('sinhviens', 'sinhviens.id','=', 'reports.sinhvien_id')

->select('*', 'reports.id as report_id','teachers.first_name as
teacher_first_name')

->where('subject_id', $subject_id)

->where('teacher_id', $teacher_id)

->get();

return response()->json($data);
}

public function ExportFileMarkCsv(Request $request)
{
    $report_id = $request -> report_id;

    $subject_id=$request -> subject_id;

    $teacher_id= auth()->user()->teacher_id;

    $semester = $request -> semester;

    $data = DB::table('reports')->join('teacher_to_subjects',
'teacher_to_subjects.id','=', 'reports.teacher_to_subject_id')
```

```
->join('teachers',
'teachers.id','=', 'teacher_to_subjects.teacher_id')

->join('subjects',
'subjects.id','=', 'teacher_to_subjects.subject_id')

->join('sinhviens', 'sinhviens.id','=', 'reports.sinhvien_id')

->select('*', 'reports.id as report_id','teachers.first_name as
teacher_first_name',)

-
>select('teacher_to_subjects.semester','teacher_to_subjects.year',

'teachers.id as teacher_id','teachers.first_name as
teacher_name',

'subjects.id as subject_id','subjects.name',

'sinhviens.id as sinhvien_id','sinhviens.first_name as
sinhvien_name',

'reports.mark'

)

//->where('subject_id', $subject_id)

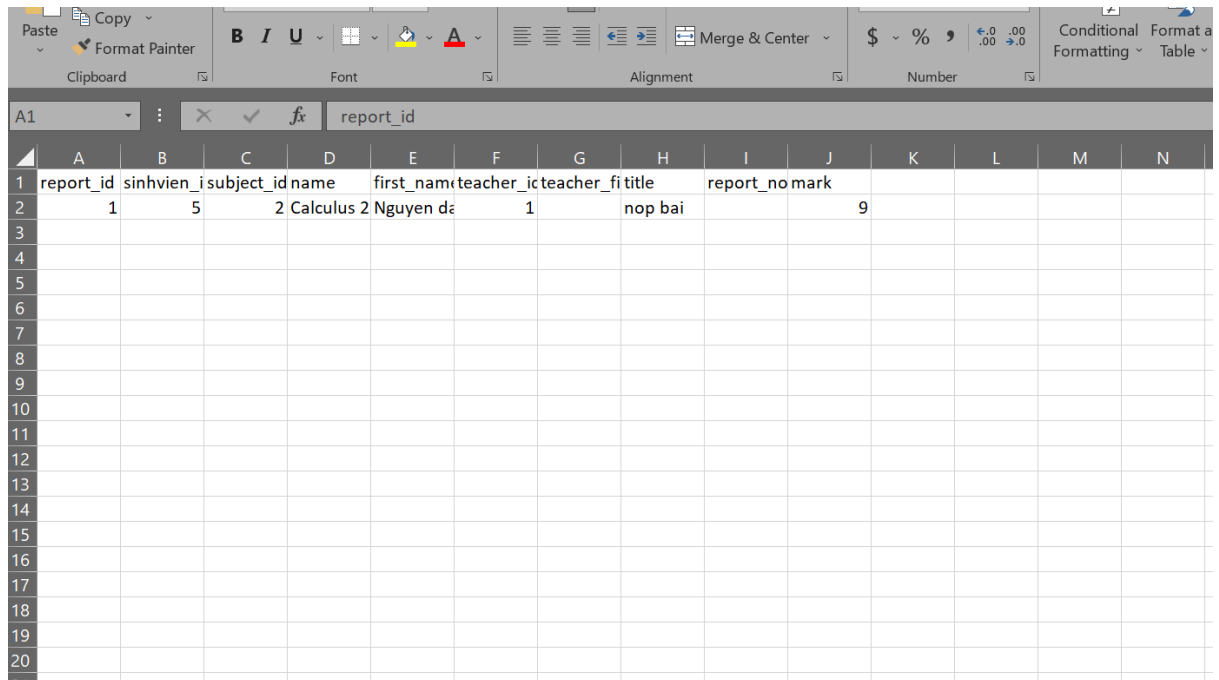
->where('teacher_id', $teacher_id)

// ->where('report_id', $report_id)

->get();

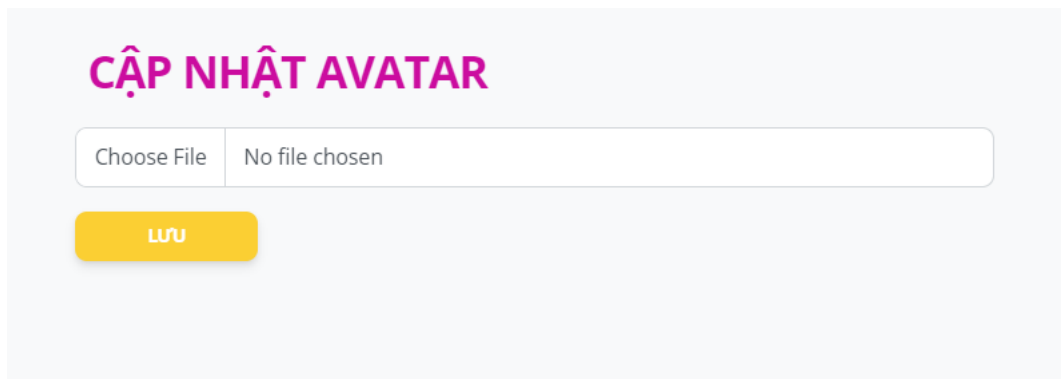
return response()->json($data);

}
```



report_id	sinhvien_id	subject_id	name	first_name	teacher_id	teacher_firstname	report_no	mark
1	5	2	Calculus 2	Nguyen da	1	nop bai		9

Hình 24. Kết quả export bảng điểm thành file csv

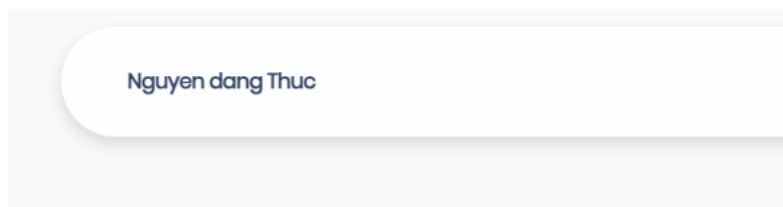
TASK 14. UPLOAD AVATAR NGƯỜI DÙNG

CẬP NHẬT AVATAR

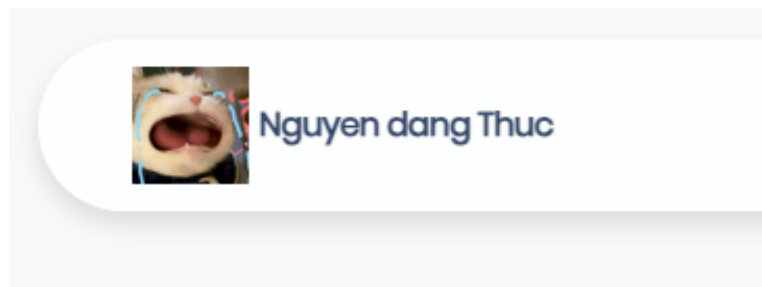
Choose File No file chosen

LƯU

Hình 25. Giao diện upload avatar người dùng



Hình 26. Trước khi cập nhật avatar người dùng



Hình 27. Sau khi cập nhật avatar người dùng

TÀI LIỆU THAM KHẢO

- [1] Google, “VuejsDocument”, <https://vuejs.org/guide/introduction.html>,
- [2] Google, “Laravel Document 8.x”, <https://laravel.com/docs/8.x/installation>
- [3] Google, “Học VueJs từ căn bản đến nâng cao”, <https://viblo.asia/s/hoc-vuejs-va-laravel-tu-can-ban-den-nang-cao-z45bxjq6ZxY>