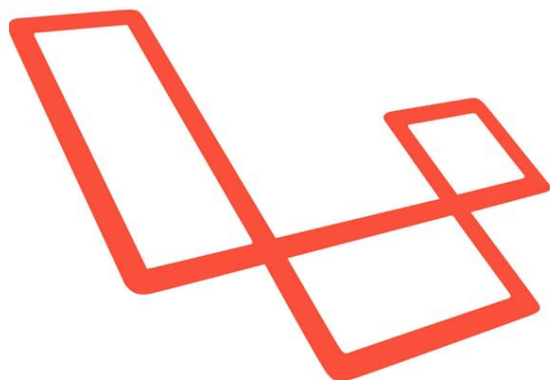


## MINI-PROJET :APPLICATION WEB Gestion LSI

### Rapport :Mini-Projet

Encadrée par : Prof. Lotfi EL AACHAK



Réalisé par

BOURYALN Sad  
ET-TOUIL Younes

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Conception</b>	<b>2</b>
2.1	Diagramme de cas d'utilis . . . . .	2
2.2	Diagramme de classe generale . . . . .	3
2.3	Base de Données . . . . .	4
<b>3</b>	<b>Back-End Laravel</b>	<b>5</b>
3.1	Authentification JSON Web Token (JWT) . . . . .	5
3.2	migration de base de donnes . . . . .	5
3.2.1	Creation des tableaux . . . . .	6
3.3	Controlleurs . . . . .	9
3.3.1	Exemple . . . . .	10
3.4	Models . . . . .	10
3.4.1	Exemple . . . . .	11
3.5	REST API . . . . .	11
3.5.1	APIs . . . . .	12
<b>4</b>	<b>Front-End VueJs</b>	<b>13</b>
4.1	Axios poure gérer les APIs . . . . .	13
4.2	Componentes : . . . . .	14
4.3	Les Vues . . . . .	15
4.3.1	HOME Page . . . . .	15
4.3.2	Login Page . . . . .	15
4.3.3	ADMIN Page . . . . .	16
4.3.4	CRUD Etudiant . . . . .	16
4.3.5	CRUD Professeurs . . . . .	17
4.3.6	CRUD Modules . . . . .	17

4.3.7	CRUD Emplois . . . . .	18
4.3.8	CRUD PFE . . . . .	18
4.3.9	Professeur Page . . . . .	19
4.3.10	CRUD Emplois . . . . .	19
4.3.11	Cours . . . . .	20
4.3.12	CRUD Notes . . . . .	20
4.3.13	Etudiants Page . . . . .	21
4.3.14	Emplois . . . . .	21
4.3.15	Cours . . . . .	22
4.3.16	notes . . . . .	22
<b>5</b>	<b>Conclusion</b>	<b>23</b>

# 1 Introduction

## **Objectif :**

l'objectif principal du projet est le développement d'une application orientée service pour garantir une bonne gestion du cycle d'ingénieur LSI de la FSTT.

## **Caractéristiques de l'application :**

l'application sera composée de deux parties, une partie Front end «VueJS», et la partie Back end « Laravel ».

L'utilisateur doit s'authentifier pour accéder à l'application. Gestion d'utilisateurs « étudiants / Professeurs ».

Gestion des Modules et affectation (Etudiants /Profs).

Gestion des PFE.

Gestion des notes (Etudiants).

Emploi du temps/réservation des salles.

## 2 Conception

La conception de logiciel met en œuvre un ensemble d'activités qui à partir d'une demande d'informatisation d'un processus permettent la conception, l'écriture et la mise au point d'un logiciel (et donc de programmes informatiques) .

### 2.1 Diagramme de cas d'utilis

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. s

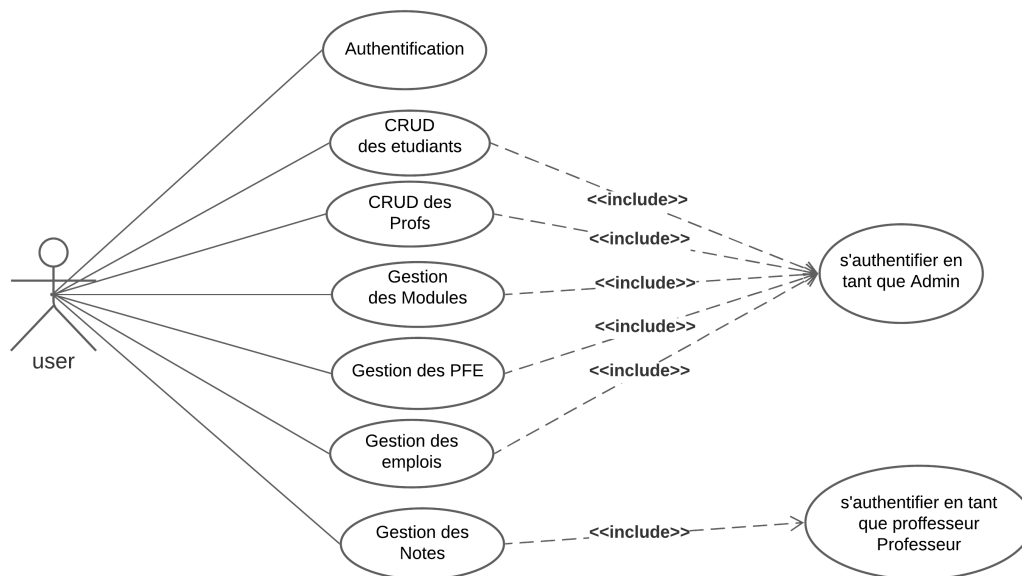


diagramme de cas d'utilisation

## 2.2 Diagramme de classe generale

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets

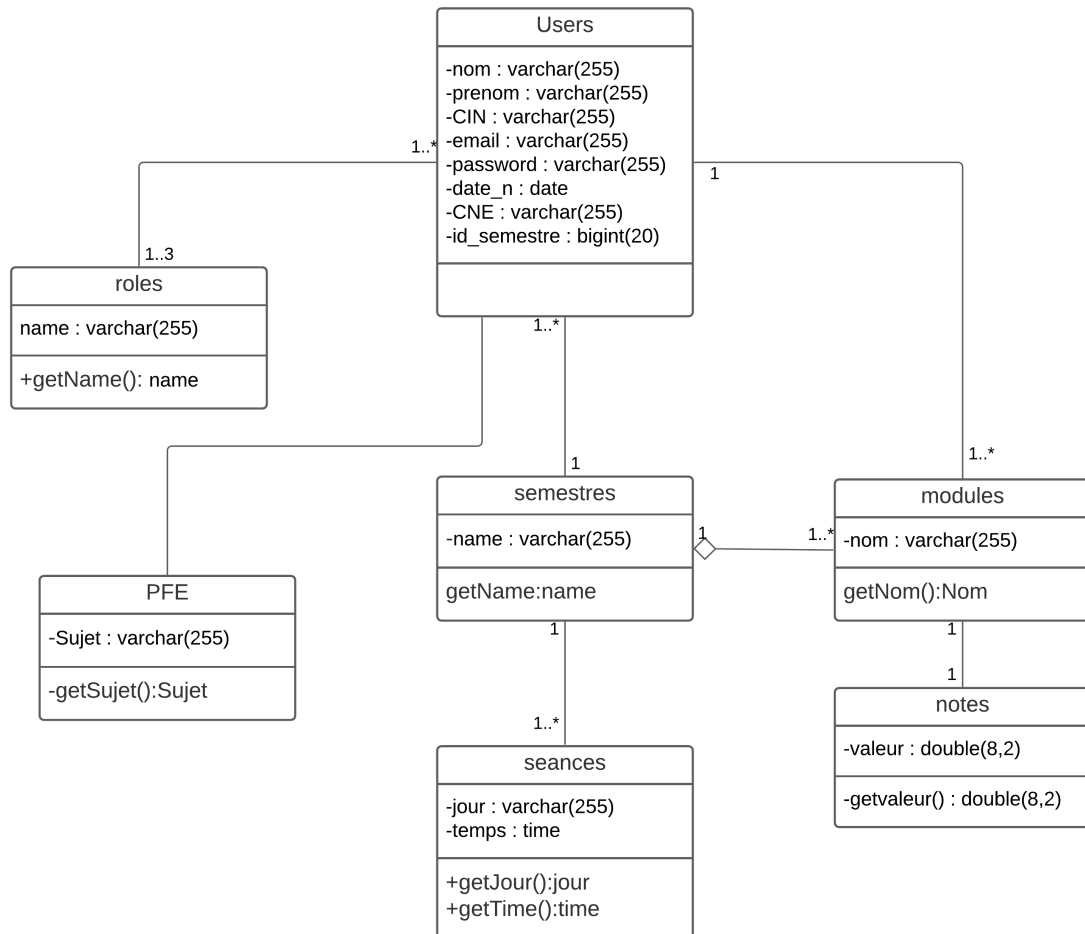
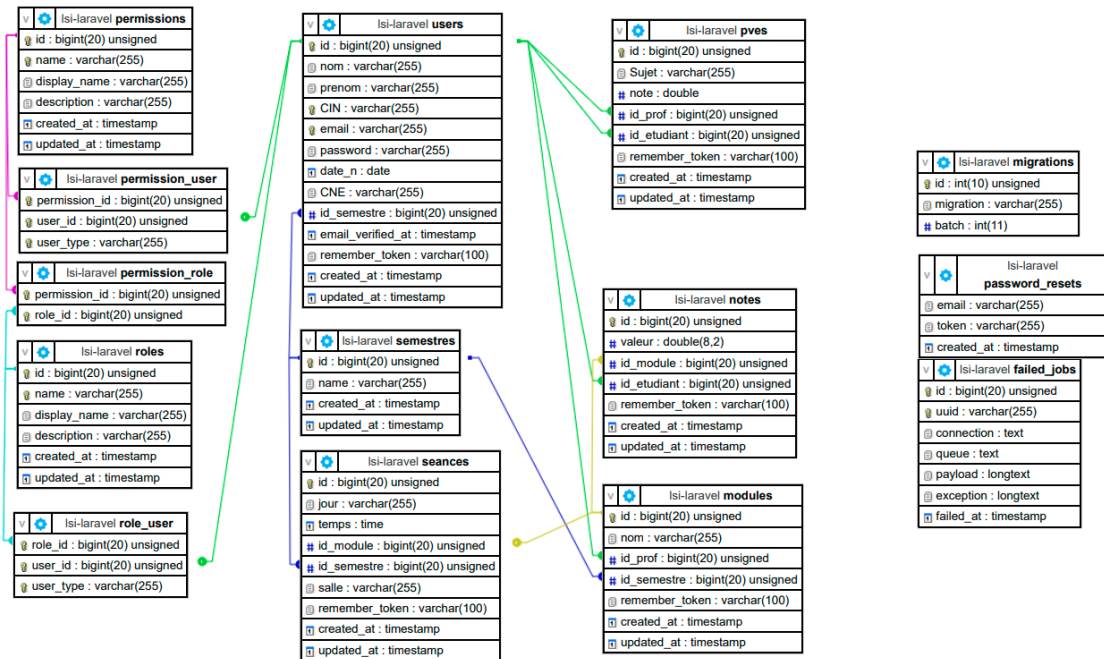


Diagramme de classe générale

## 2.3 Base de Données



Base de Données

## 3 Back-End Laravel

Laravel est un framework de développement d'applications Web open source écrit en PHP. Laravel est créé par Taylor Otwell et publié sous licence MIT. Laravel vous offre un développement rapide d'applications suivant le modèle architectural modèle-vue-contrôleur (MVC). Laravel est un framework qui vous permet de créer plus facilement des applications Web professionnelles mais puissantes en suivant une syntaxe et un modèle architectural très expressifs et élégants. Laravel est le framework de développement Web le plus populaire et le plus préféré par rapport aux autres frameworks MVC basés sur PHP en raison de sa simplicité, de ses performances, de ses fonctionnalités et de son évolutivité. Laravel rend le processus de développement assez simple et agréable sans sacrifier les fonctionnalités de l'application.

### 3.1 Authentification JSON Web Token (JWT)

JSON Web Token (JWT) est une norme ouverte qui définit un moyen compact et autonome de transmettre en toute sécurité des informations entre les parties en tant qu'objet JSON. Ces informations peuvent être vérifiées et fiables car elles sont signées numériquement. Les JWT peuvent être signés en utilisant un secret ou un public.

Voici les scénarios où les jetons Web JSON sont utiles :

- **Autorisation** : il s'agit du scénario le plus courant pour l'utilisation de JWT. Une fois que l'utilisateur est connecté, chaque demande suivante inclura le JWT, permettant à l'utilisateur d'accéder aux routes, services et ressources autorisés avec ce jeton. L'authentification unique est une fonctionnalité qui utilise largement JWT de nos jours, en raison de sa faible surcharge et de sa capacité à être facilement utilisée dans différents domaines.

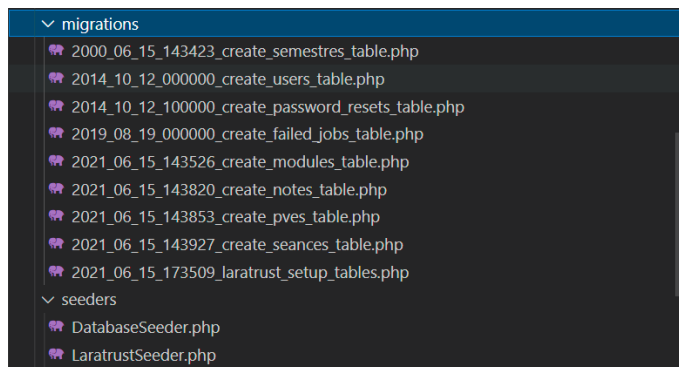
- **Échange d'informations** : les jetons Web JSON sont un bon moyen de transmettre des informations en toute sécurité entre les parties. Parce que les JWT peuvent être signés, par exemple, en utilisant

### 3.2 migration de base de données

Une migration permet de créer et de mettre à jour un schéma de base de données. Autrement dit, vous pouvez créer des tables, des colonnes dans ces tables, en sup-



primer, créer des index... Tout ce qui concerne la maintenance de vos tables peut être pris en charge par cet outil.



## Migration

### 3.2.1 Création des tableaux

- tableau des Utilisateurs

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('nom');
        $table->string('prenom');
        $table->string('CIN')->unique()->nullable();
        $table->string('email')->unique();
        $table->string('password');
        $table->date('date_n')->nullable();
        $table->string('CNE')->nullable();
        $table->unsignedBigInteger('id_semestre')->nullable();
        $table->foreign('id_semestre')->references('id')->on('semestres');
        $table->timestamp('email_verified_at')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
}
```

#### tableau Users

- tableau des Semestres

```
public function up()
{
    Schema::create('semestres', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->timestamps();
    });
}
```

tableau Semestres

- tableau des Modules

```
public function up()
{
    Schema::create('modules', function (Blueprint $table) {
        $table->id();
        $table->string('nom');
        $table->unsignedBigInteger('id_prof');
        $table->foreign('id_prof')->references('id')->on('users');
        $table->unsignedBigInteger('id_semestre');
        $table->foreign('id_semestre')->references('id')->on('semestres')
            ->onDelete('cascade')
            ->onUpdate('cascade');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

tableau Modules

- tableau des Seances

```
class Seance extends Model
{
    use HasFactory;
    protected $fillable = [
        'jour',
        'temps',
        'id_module',
        'id_semestre',
        'salle'
    ];

    protected $casts = [
        'temps' => 'date:hh:mm'
    ];
}
```

tableau Seances

- tableau des Notes

```
public function up()
{
    Schema::create('Notes', function (Blueprint $table) {
        $table->id();
        $table->float('valeur');
        $table->unsignedBigInteger('id_module');
        $table->foreign('id_module')->references('id')->on('modules')
            ->onDelete('cascade');
        $table->unsignedBigInteger('id_etudiant');
        $table->foreign('id_etudiant')->references('id')->on('users')
            ->onDelete('cascade');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

tableau Notes

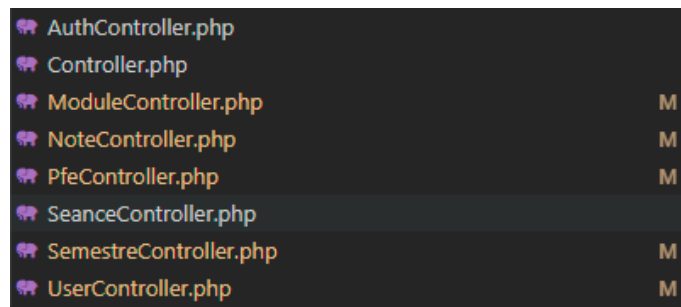
- tableau des PFE

```
public function up()
{
    Schema::create('pves', function (Blueprint $table) {
        $table->id();
        $table->string('Sujet');
        $table->double('note');
        $table->unsignedBigInteger('id_prof');
        $table->foreign('id_prof')->references('id')->on('users');
        $table->unsignedBigInteger('id_etudiant');
        $table->foreign('id_etudiant')->references('id')->on('users');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

tableau PFE

### 3.3 Contrôleurs

Au lieu de définir toute votre logique de traitement des requêtes en tant que fermetures dans vos fichiers de route, vous souhaitez peut-être organiser ce comportement à l'aide de classes "contrôleurs". Les contrôleurs peuvent regrouper la logique de traitement des requêtes associée dans une seule classe. Par exemple, une classe `UserController` peut gérer toutes les demandes entrantes liées aux utilisateurs, y compris l'affichage, la création, la mise à jour et la suppression d'utilisateurs. Par défaut, les contrôleurs sont stockés dans le répertoire `app/Http/Controllers`.



Controllers

### 3.3.1 Exemple

```
class UserController extends Controller
{
    public function GetStu($semestre)
    {
        $users = User::where('id_semestre', $semestre)->get();
        return $users;
    }
    public function getstudents()
    {
        $users = User::whereRoleIs('student')->get();
        return $users;
    }
    public function GetPro()
    {
        $users = User::whereRoleIs('Teacher')->get();
        return $users;
    }

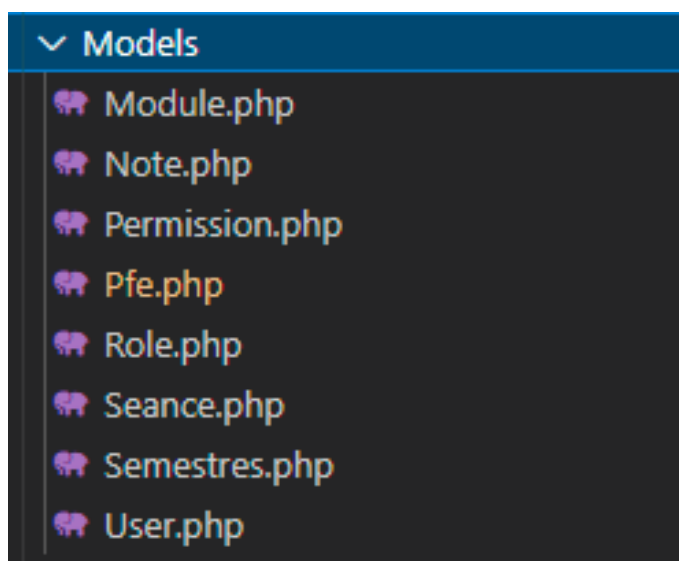
    public function DeleteUser($id)
    {
        $User = User::find($id);
        $User->delete();

        return response()->json('User deleted!');
    }
    public function update(Request $request, $id)
    {
        $Seance = User::find($id);
        $Seance->update($request->all());

        return response()->json('User updated!');
    }
}
```

UserController

## 3.4 Models



Models

### 3.4.1 Exemple

```
class User extends Authenticatable implements JWTSubject
{
    use LaratrustUserTrait;
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'nom',
        'prenom',
        'CNE',
        'CIN',
        'date_n',
        'email',
        'password',
        'id_semestre'
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];
}
```

UserModel

## 3.5 REST API

Les API REST sont couramment utilisées par divers sites Web, permettant aux utilisateurs de communiquer avec ces services cloud.

La forme complète de l'API REST est l'interface de programmation d'application de transfert d'état de représentation ou l'API REST. Cela signifie que lorsqu'une API RESTful est appelée, le serveur transfère a représentation des ressources demandées Etat au système client.

### 3.5.1 APIs

```
Route::group(
[
    'middleware' => 'api',
],
function ($router) {
    Route::post('/register', [AuthController::class, 'register']);
    Route::post('/login', [AuthController::class, 'login']);
    Route::post('/checkToken', [AuthController::class, 'checkToken']);
    Route::post('/logout', [AuthController::class, 'logout']);
    Route::post('/refresh', [AuthController::class, 'refresh']);
}
);

Route::get('/getstu/{semestre}', [UserController::class, 'GetStu']->middleware(['api', 'role:Admin|Teacher']));
Route::get('/getstu', [UserController::class, 'getstudents']->middleware(['api', 'role:Teacher']));
Route::put('/updateuser/{id}', [UserController::class, 'update']->middleware(['api', 'role:Admin']));

Route::get('/getpro', [UserController::class, 'GetPro']->middleware(['api', 'role:Admin']));
Route::delete('/deleteuser/{id}', [UserController::class, 'DeleteUser']->middleware(['api', 'role:Admin']));

Route::resource('Modules', ModuleController::class)->middleware(['api', 'role:Admin|Student|Teacher']);
Route::resource('Semestres', SemestreController::class)->middleware(['api', 'role:Admin']);
Route::resource('Seance', SeanceController::class)->middleware(['api', 'role:Admin|Student|Teacher']);
Route::resource('Note', NoteController::class)->middleware(['api', 'role:Teacher|Student']);
Route::resource('Pfe', PfeController::class)->middleware(['api', 'role:Admin|Teacher']);
```

APIs

## 4 Front-End VueJs

VueJS est un framework JavaScript progressif et un framework Front End utilisé pour développer des interfaces Web interactives. L'accent est davantage mis sur la partie vue, qui est l'extrémité avant. Il est très facile à intégrer avec d'autres projets et bibliothèques. D'un autre côté, Vue est également parfaitement capable d'alimenter des applications sophistiquées à page unique lorsqu'il est utilisé en combinaison avec des outils modernes et des bibliothèques de support.

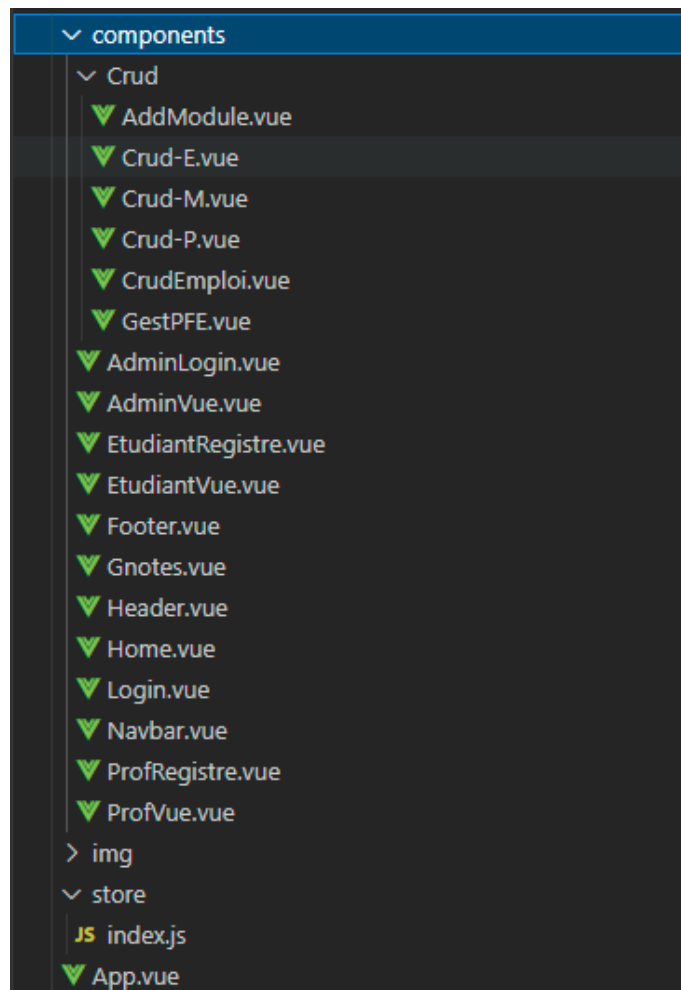
### 4.1 Axios pour gérer les APIs

Lors de la création d'une application pour le Web, vous souhaiterez peut-être souvent utiliser et afficher des données à partir d'une API. Il existe plusieurs façons de le faire, mais une approche très populaire consiste à utiliser axios, un client HTTP basé sur des promesses.

Dans cet exercice, nous utiliserons l'API CoinDesk pour afficher les prix Bitcoin, mis à jour toutes les minutes. Tout d'abord, nous installerions axios avec npm/yarn ou via un lien CDN.



## 4.2 Composantes :

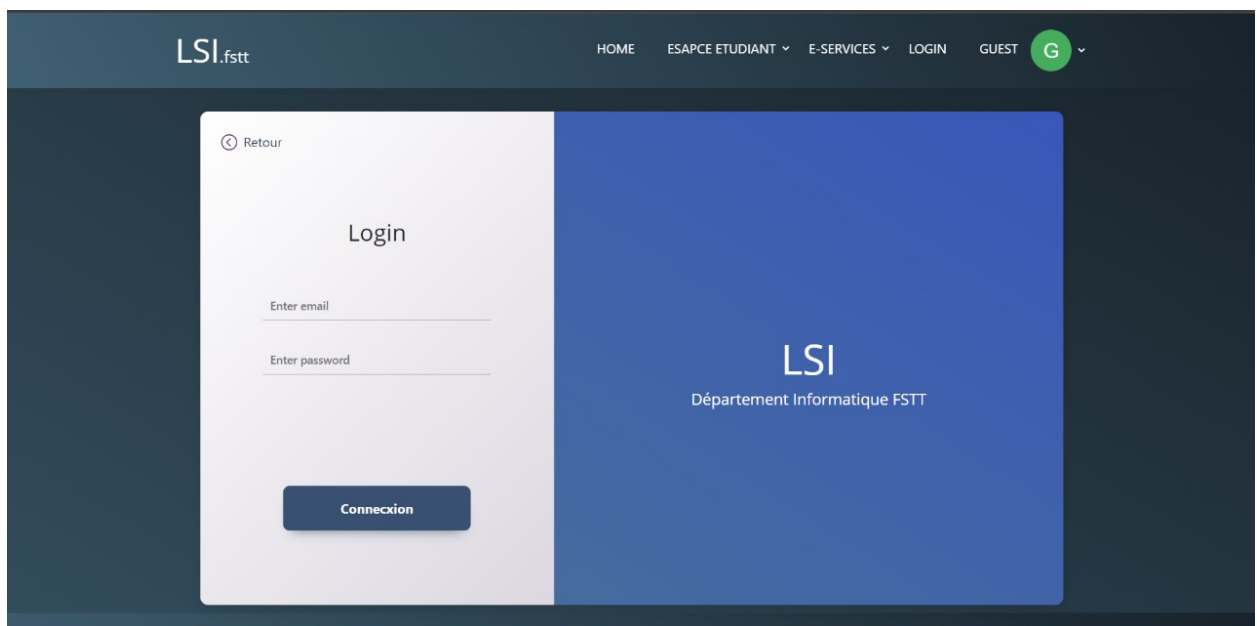


## 4.3 Les Vues

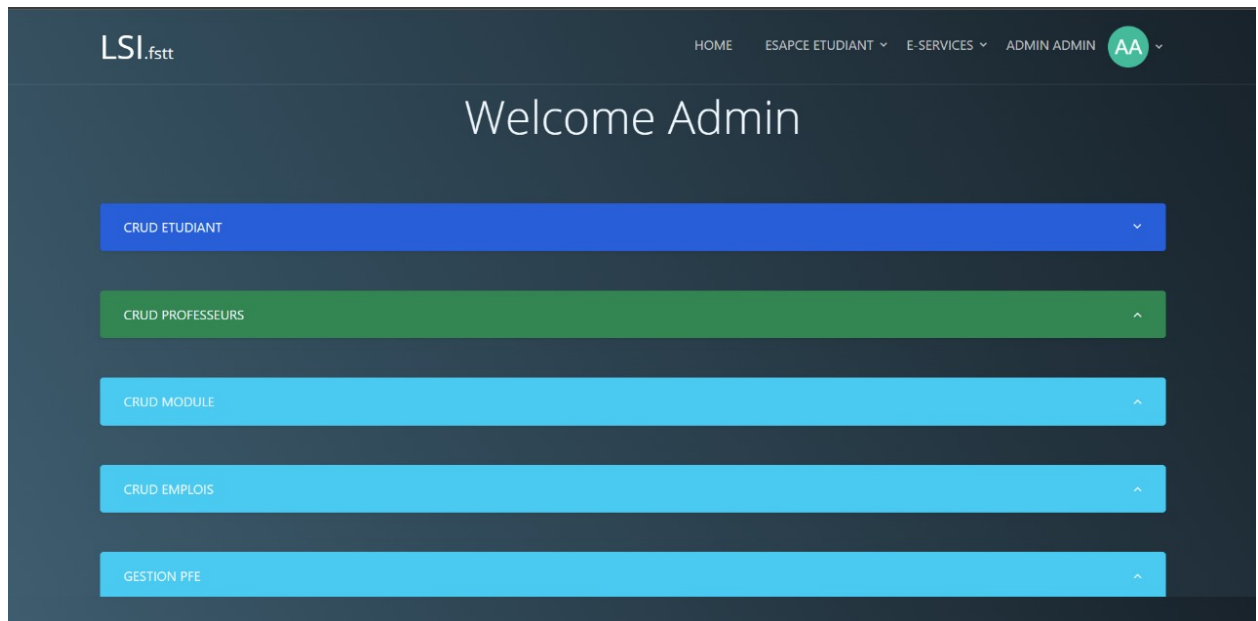
### 4.3.1 HOME Page



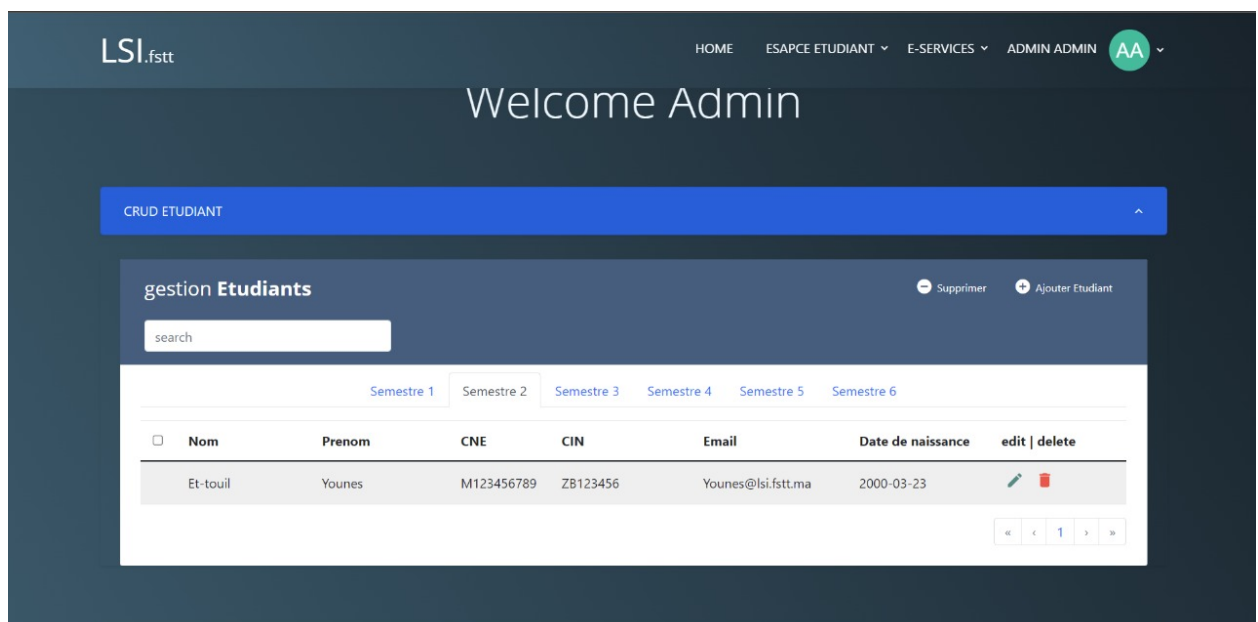
### 4.3.2 Login Page



### 4.3.3 ADMIN Page



### 4.3.4 CRUD Etudiant



### 4.3.5 CRUD Professeurs

LSI.fstt

HOME ESAPCE ETUDIANT E-SERVICES ADMIN ADMIN AA

CRUD ETUDIANT

CRUD PROFESSEURS

gestion Professeurs

Supprimer Ajouter un professeur

<input type="checkbox"/>	Id	Nom	Prenom	Email	
	5	Zili	Hassan	Hassan@lsi.fstt.ma	
	6	Fennan	Abdelhadi	Abdelhadi@lsi.fstt.ma	

« < 1 > »

### 4.3.6 CRUD Modules

LSI.fstt

HOME ESAPCE ETUDIANT E-SERVICES ADMIN ADMIN AA

CRUD MODULE

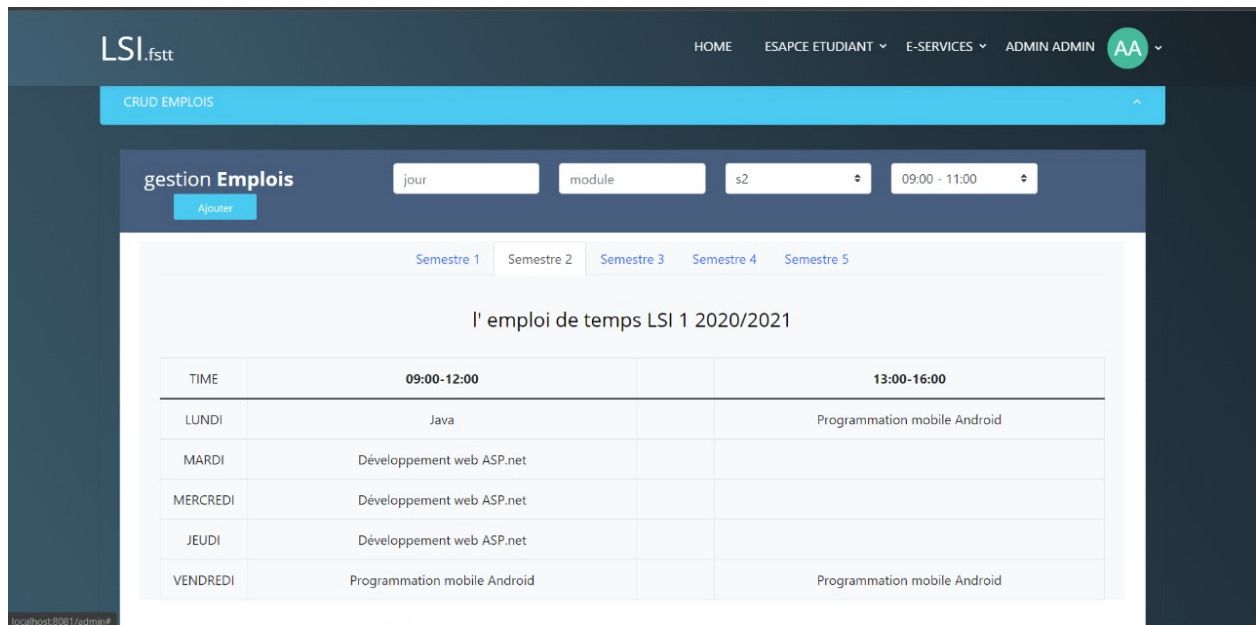
gestion Modules

module --semestre-- --Prof-- confirmer

Module	professeure	Semestre	action
Java	Fennan Abdelhadi	Semestre 2	
Théorie des graphes	Ghadi Abderrahim	Semestre 1	
Programmation mobile Android	Zili Hassan	Semestre 2	
Développement web ASP.net	Zili Hassan	Semestre 2	
Base de données	Zili Hassan	Semestre 1	

« < 1 > »

### 4.3.7 CRUD Emplois



**gestion Emplois**

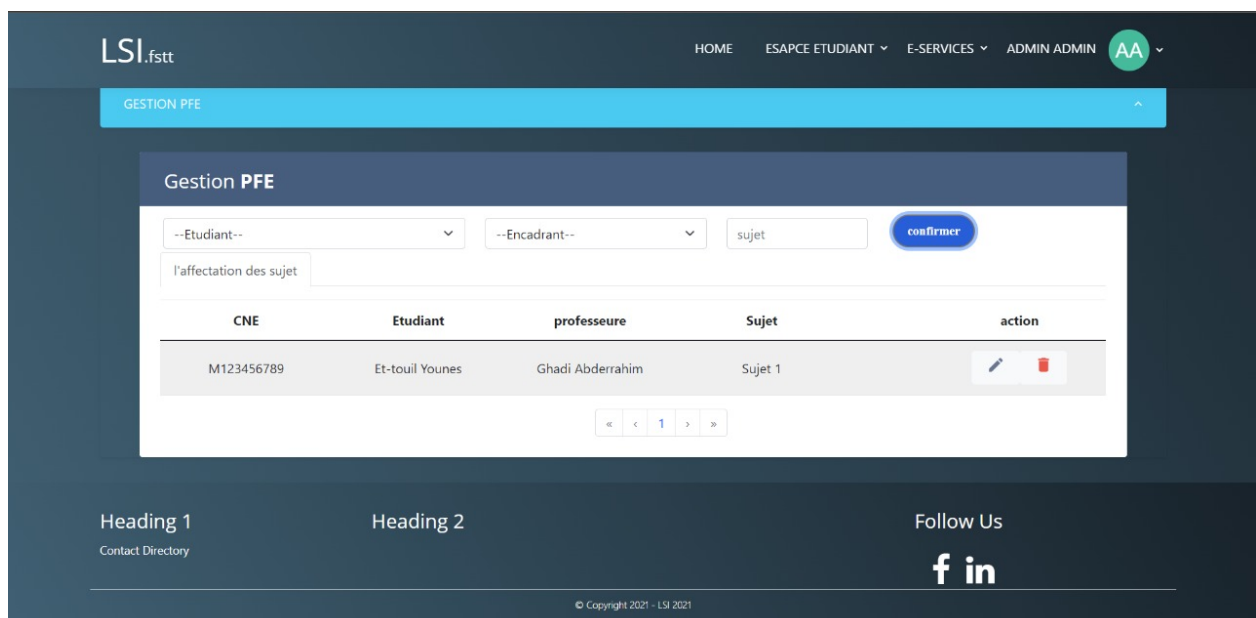
Ajouter

Semestre 1 Semestre 2 Semestre 3 Semestre 4 Semestre 5

**l' emploi de temps LSI 1 2020/2021**

TIME	09:00-12:00	13:00-16:00
LUNDI	Java	Programmation mobile Android
MARDI	Développement web ASP.net	
MERCREDI	Développement web ASP.net	
JEUDI	Développement web ASP.net	
VENDREDI	Programmation mobile Android	Programmation mobile Android



### 4.3.8 CRUD PFE



**Gestion PFE**

--Etudiant-- --Encadrant-- sujet **confirmer**

**l'affectation des sujet**

CNE	Etudiant	professeure	Sujet	action
M123456789	Et-touil Younes	Ghadi Abderrahim	Sujet 1	 

« < 1 > »

Heading 1  
Contact Directory

Heading 2

Follow Us  
**f in**

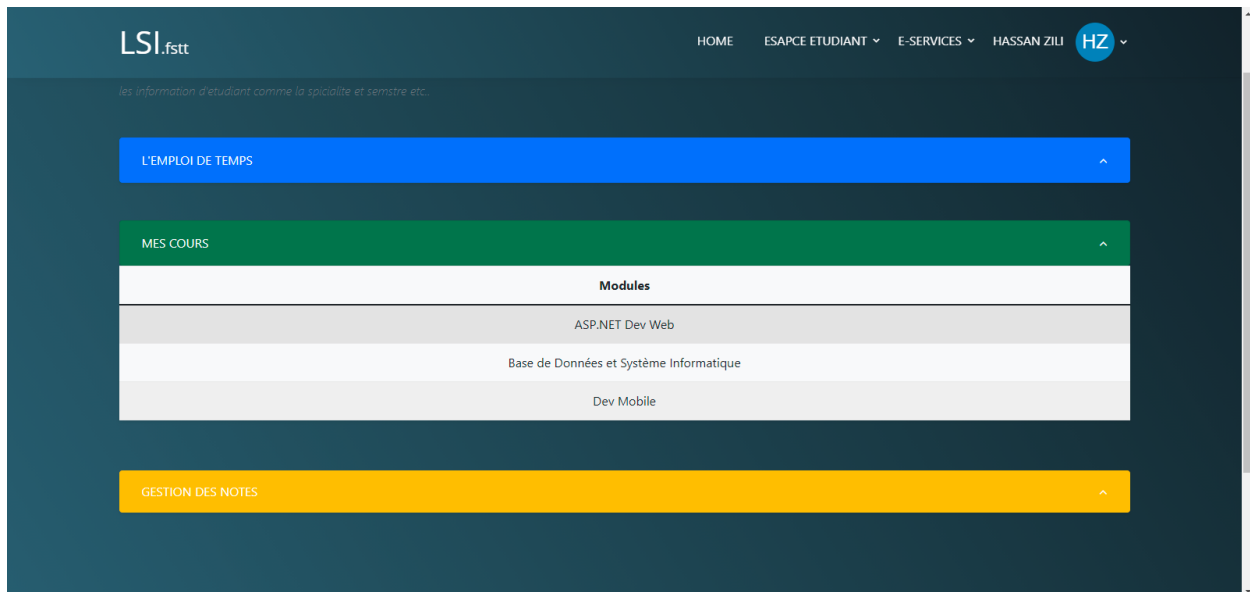
© Copyright 2021 - LSI 2021

### 4.3.9 Professeur Page

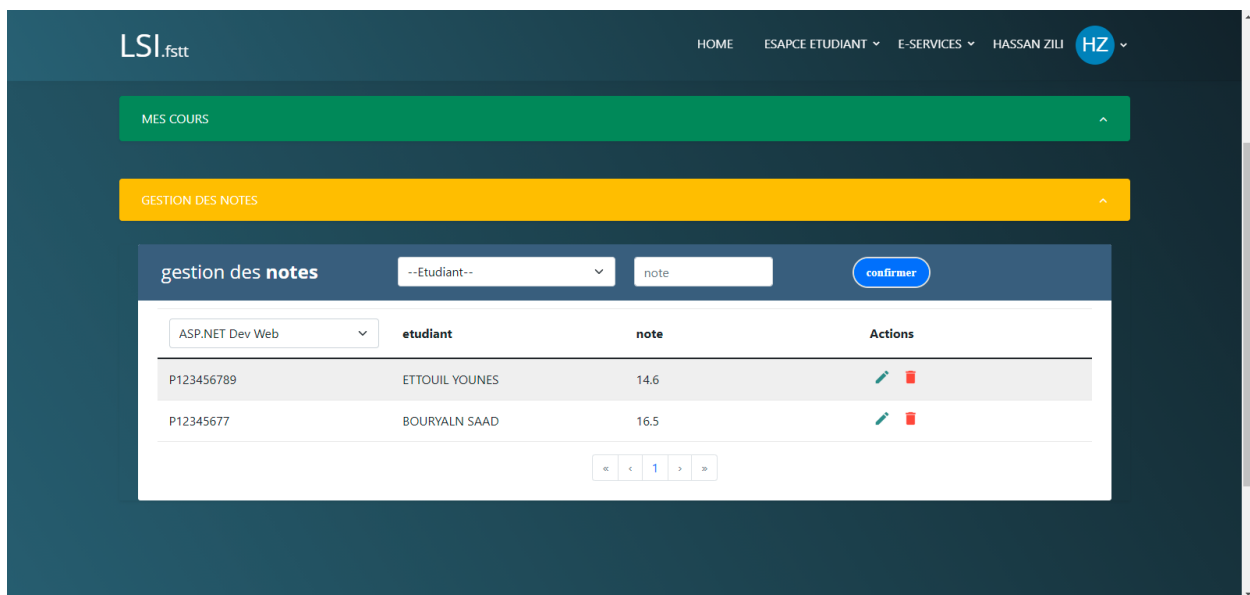


### 4.3.10 CRUD Emplois

### 4.3.11 Cours



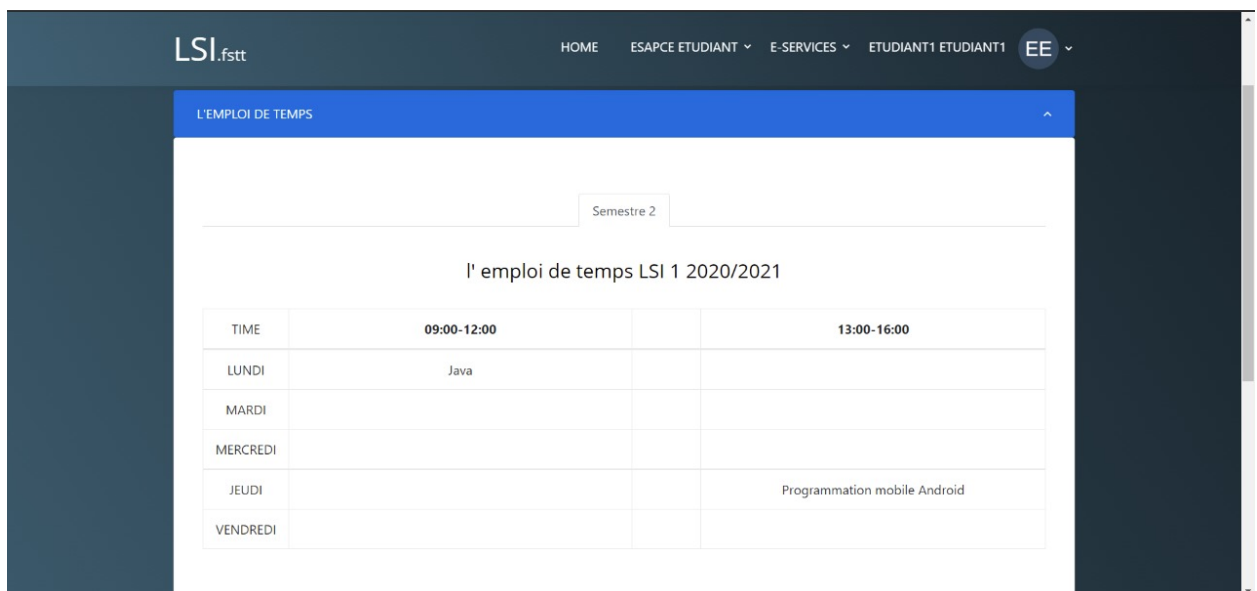
### 4.3.12 CRUD Notes



### 4.3.13 Etudiants Page



### 4.3.14 Emplois





#### 4.3.15 Cours

The screenshot shows the LSI.fstt student portal. The top navigation bar includes the logo, a home link, and dropdown menus for 'ESAPCE ETUDIANT', 'E-SERVICES', and 'ETUDIANT1 ETUDIANT1'. A user profile icon 'EE' is on the right. Below the navigation bar, there are two expandable sections: 'MES COURS' (green) and 'RELEVÉ DE NOTES' (yellow). The 'MES COURS' section is expanded, showing a table titled 'Liste des Cours' for 'Semestre S2'.

Module	Prof
Java	Fennan Abdelhadi
Programmation mobile Android	Zili Hassan
Développement web ASP.net	Zili Hassan

#### 4.3.16 notes

The screenshot shows the LSI.fstt student portal. The top navigation bar is identical to the previous one. Below it, the 'RELEVÉ DE NOTES' (yellow) section is expanded, showing a table titled 'Notes des Modules' for 'Semestre S2'.

Module	Note
Développement web ASP.net	16
Programmation mobile Android	20

## 5 Conclusion

Ce projet nous a permis de bien comprendre le fonctionnement des frameworks, et nous a donné l'opportunité de travailler et de collaborer en binôme.

Lien vers **Backend Laravel** :

<https://github.com/saadsb20/BackEnd-Gestion-LSI-.git>

Lien vers **Frontend Vue js** :

<https://github.com/saadsb20/FrontEnd-Gestion-LSI-.git>