

Juin 2022

Rapport sur la réalisation d'une application de gestion des étudiants

Projet intégrant LINUX et PL/SQL

Réalisé par : Maysae Hmamouchi Et Abderrahim Fazazi

Encadré par : Prof. Mohamed Ghailani

Remerciements

Avant d'entamer notre présent rapport, nous tenons d'adresser nos sincères remerciements à Mr Mohamed Ghailani qui nous a offert l'opportunité de travailler sur ce projet qui va marquer notre parcours académique. Nous le remercions aussi pour son encadrement, ses conseils, sa patience et les efforts qu'il a fournis afin de nous présenter la meilleure formation que nous pouvions espérer.

Nous tenons également à remercier tous les enseignants qui nous ont car sans eux, nous n'aurions pas pu bien assimiler tous les prérequis nécessaires pour la réalisation de notre projet.

À propos de nous

Maysae Hmamouchi, élève ingénieure en première année cycle ingénieur, génie informatique à l'ENSA de Tanger, intéressée par les nouvelles technologies, l'intelligence artificielle, secrétaire générale-adjointe et coordinatrice du comité exécutif de l'Association Marocaine pour un Environnement Durable.

« Durant la réalisation de ce projet, j'ai pu apprendre des techniques de gestion des projets informatiques, j'ai développé mon esprit de créativité et le sens de partage d'informations avec mes camarades.»

Abderrahim Fazazi élève ingénieur en première année cycle ingénieur à l'ENSA de Tanger, passionné par l'entreprenariat, le domaine It et précisément le développement Web, free-lancer depuis une année, fondateur et président du club Ensatpress.

« Durant la réalisation du ce projet, j'ai développé plusieurs compétences tels que: l'esprit d'équipe, la prise de l'initiative, la gestion du temps et du travail et la communication.»

Mission et Vision

Nous visons offrir une application web attractive et qui est capable de permettre au cadre administratif de l'ENSAT, les moyens de gestion des données des étudiants : profil, notes, validation... Nous visons également la création d'un espace étudiant, à travers lequel, nous allons permettre aux élèves ingénieurs, la consultation de leurs notes, le contact des professeurs par email...

Durant toutes les phases de la réalisation du projet, nous avons comme mission le déploiement des différents outils et langages de programmation que nous avons pu manipuler le long de cette année académique ou bien à travers nos auto-formations et également les compétences de gestion de travail en équipe, communication et collaboration.

Code QR pour accéder à
notre site :



Résumé

Dans le but de la réalisation de ce projet, nous avons commencé par la répartition des tâches entre les membres du groupe (voir planning des taches), chose qui n'était pas facile vu que nous sommes que deux personnes dans l'équipe. Nous avons donné la priorité au début à l'établissement d'un environnement de travail sous lequel nous avons développé l'application. Nous avons effectué les installations demandées dans le cahier de charge ainsi que des tests de fonctionnement sur ces derniers. Par la suite, nous avons commencé le développement Backend de l'application et tous les tests sont effectués grâce à l'outil PostMan (voir chapitre 2). Nous avons décidé de travail avec cet outil avant de passer à la réalisation des interfaces nécessaires, dans l'objectif d'effectuer tous les tests sur les Apis afin éviter les erreurs qui peuvent se générer après et pour savoir précisément la source de chaque erreur, chose qui est difficile et presque ingérable lors du travail directement avec le FronEnd. Généralement, le travail avec un tel outil est une bonne habitude que les développeurs doivent apprendre.

Afin d'assurer que notre application soit interactive et responsive, nous avons utilisé ReactJS et d'autres outils pour intégrer le code Javascript (voir chapitre 2) et aboutir à un développement FrontEnd souple avec des interfaces attractives pour les différents utilisateurs de l'application (voir chapitre 3-4-5-6).

Dans le but d'assurer un bon fonctionnement de l'application, nous avons développé différentes procédures et fonctions ainsi que triggers (voir chapitre 3), mais vu que nous avons travaillé avec Laravel et dans le but d'approfondir notre maîtrise de ce Framework puissant, nous avons tiré profit de l'outil Observers: événements programmés qui se déclenchent selon le besoin. Et dans le cadre du travail sous une architecture MVC assurée par ce Framework, nous avons organisé notre code suivant des contrôleurs, des modèles et des vues.

Comme il était indiqué dans le cahier de charge, nous avons conçu la possibilité d'échange des emails entre les étudiants et leurs professeurs, chose qui est assurée grâce à l'utilisation de Sendinblue SMTP et la configuration du protocole SMTP (voir chapitre 3).

Dans le cadre d'amélioration du produit que nous souhaitons vous délivrer, nous avons ajouté des fonctionnalités, à savoir : la possibilité de l'exportation des notes des étudiants sous des fichiers Excel et la sécurisation des informations de connexion des utilisateurs par l'envoi des mots de passe générés automatiquement au moment de l'ajout d'un nouveau étudiant ou responsable de filière via l'email institutionnel (voir chapitre 8). Lors d'une première connexion, l'utilisateur est amené à modifier son mot de passe pour plus de sécurité. De plus, nous avons hébergé notre application sur un serveur réel en créant un compte payant sur DigitalOceans (voir chapitre 9).

Afin d'assurer la sécurité de la connexion à notre application, nous avons utilisé le protocole HTTPS(voir chapitre 9) sur le serveur d'hébergement de l'application et dans lequel nous avons effectué toutes les installations qui ont été déjà faites sur la machine locale.

Au fil de la réalisation du projet, nous avons organisé plusieurs réunions pour discuter les états d'avancement de chaque tâche, les modifications et améliorations à faire... Nous avons également développé un fort esprit d'équipe, de curiosité et de communication non seulement au sein de l'équipe elle-même mais également avec nos camarades avec lesquels nous avons eu des échanges à propos des différentes difficultés rencontrées. Nous pouvons dire que ce projet n'avait pas seulement un impact important sur notre parcours académiques vu les nombreuses techniques et connaissances que n'avons pu appliquer mais également, il nous a présenté une opportunité pour développer nos personnalités entant que futurs ingénieurs.

Contenu

Code QR pour accéder à notre site :	4
Résumé	5
Introduction :.....	12
Chapitre 1.....	13
Installations et outils déployés pour la réalisation de l'application.....	13
1-Installation PHP et ses packages sous Ubuntu :.....	14
2-Installation d'Apache2:.....	16
3-Installation, configuration d'Oracle et création d'un utilisateur :.....	17
Pré-installation :	17
Installation Oracle :.....	20
Configuration d'Oracle et création d'un utilisateur:.....	20
4-Installation du Composer et Laravel:.....	23
Instalation de Composer :.....	24
.....	26
Installation de Laravel en utilisant Composer:	26
5- Installation et configuration de Nginx:	28
6-Installation et configuration du serveur DNS:	32
7-Installation et configuration de PostFix:.....	35
.....	35
Chapitre 2	38
Autres outils déployés pour une meilleure qualité de l'application	38
1- Postman:	39
Présentation générale :.....	39
Fonctionnement :	39
Architecture et cycle de développement d'une Api :	40
Les différentes étapes du cycle de développement d'une Api :	41
Installation	41

2- NodeJS:.....	42
Présentation générale:.....	42
Architecture :	42
Installation de Node et NPM:	43
3-React Js :	45
Présentation générale:.....	45
Intérêt de son utilisation :	45
Chapitre 3	46
Présentation de l'application	46
1-Les Procédeurs nécessaires au fonctionnement de l'application:	47
Création des procédures (Controllers) sous Laravel:	47
Fonctions pour gérer l'authentification:	48
.....	49
.....	50
Fonctions pour gérer les étudiants	51
Fonctions pour gérer les modules:.....	54
Fonctions pour gérer les éléments des modules:	57
.....	57
Fonctions pour gérer les notes:.....	60
Fonctions pour gérer les moyennes:	63
Fonctions pour gérer les responsables des filières	65
2-Les déclencheurs et observers utilisés:	68
Création des tables grace Aux migrations:.....	68
La table responsable_filière:	69
La table Eleve	70
La table Filière:	71
La table element_modules :	72
La table moyennes	72
La table notes	72
Les triggers:	73
Utilisation des événements (Observers) dans Laravel:	77
Fonction Created:.....	78
Fonction Updated:.....	79
Fonction Deleted:.....	79

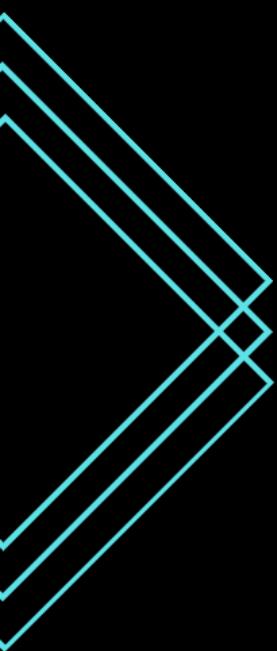
3-Utilisation de Laravel Rest Api:	80
4-Utilisation des Midelwares:	82
5-Les modèles (Relationships) sous Laravel:.....	83
Modèle pour Responsable Filière:.....	84
Modèle pour Moyenne:.....	84
Modèle pour Note	85
Modèle pour Eleve:	85
Modèle pour Utilisateur:.....	86
Modèle pour Module:	87
6-Présentation générale du site:	87
Page d'accueil:.....	88
Barre de navigation et Footer:	89
Page About us :	90
Page de Log in:.....	92
7-Échange des e-mails entre les étudiants et leurs professeurs:.....	93
Configuration d'un serveur SMTP:	93
C'est pour cela on a décidé d'utiliser Sendinblue smtp :	93
Création d'un compte sendinblue :	93
Configuration du fichier .env :	94
Controller :.....	95
Test de la requête par Postman :	95
Test d'Email :	96
Chapitre 4	97
Espace Étudiant	97
1-Cosultation et modification du profil	98
.....	99
2-Consultation des notes et modules	99
3-Envoi des emails aux chefs de filières	100
Chapitre 5	101
Espace Chèf de filière	101
1-Saisie et modification des notes :	102
saisie des notes.....	102
Affichage des notes	102
modification des notes :	103
Chapitre 6	104

Espace Admin	104
1-Manipulations sur les étudiants :	105
Recherche d'un étudiant ou d'une classe :	105
Recherche d'un étudiant par nom, prénom ou code :.....	106
Ajout d'un étudiant :	106
Modification d'un élève :.....	106
2-Manipulations sur les notes :	107
3-Manipulations sur les modules et éléments de modules :	107
Ajout d'un module :.....	107
Modification d'un module :.....	108
.....	108
L'affichage de la liste des modules:.....	108
4-Manipulations sur les responsables de filières :	108
Ajout d'un responsable :	109
Modification des informations d'un responsable :	109
Affichage des informations des responsables :.....	109
Chapitre 7	111
Tests et Comparaisons	111
1-Test connexion PHP et Oracle sous Apache:	112
Installation des packages pour connecter le code php avec Oracle:	112
Test simple de connexion :	115
2-Test Nginx:	116
Test de connexion :	116
3-Comparaison entre Apache et Nginx:	117
Caractéristiques des deux serveurs :.....	117
Caractéristiques d'Apache:.....	118
Caractéristiques de Nginx :.....	118
Comparaison entre les deux au niveau de l'architecture et la rapidité:	118
Comparaison d'efficacité en termes d'exigences système :	120
4-Test Postfix:.....	122
5-Comparaison entre Sendmail et Postfix:	123
Les caractéristiques de SendMail :	123
Caractéristiques de Postfix :	124
Comparaison au niveau de l'architecture:	124
Comparaison au niveau de la vitesse et la performance :	125

Comparaison au niveau de la sécurité :	126
Comparaison au niveau du nombre des utilisateurs :.....	126
Chapitre 8	127
Fonctionnalités ajoutées à l'application.....	127
1-Exportation/Importation des notes via des fichiers Excel :.....	128
2-Renforcement de sécurité des mots de passe des étudiants et responsables de filières:	131
3-Génération d'un code QR pour entrer dans le site :	133
Chapitre 9	134
Déploiement de l'application sur un serveur réel et configuration d'un nom de domaine	134
1-Creation d'un sever Ubuntu via DigitalOceans	135
2-Creation d'une clé ssh dans la machine locale pour accéder au serveur via le protocole ssh.....	136
3-Connexion du serveur via la machine locale :	137
.....	137
Et nous avons envoyé le projet vers le serveur pour l'héberger par la commande :	138
4-Configuration DNS.....	138
5-Installlation de SSL (HTTPS) :.....	138
Planning des tâches	142
Références	143

Introduction :

A travers ce projet, nous visons la création d'une application qui va permettre la gestion des notes et d'autres informations des étudiants de l'ENSAT, l'échange des emails entre les étudiants et leurs professeurs ainsi que l'extraction des notes des étudiants. Nous allons commencer par la présentation des différentes installations que nous avons réalisé et qui entrent dans le cadre des outils demandés dans le cahier de charges. Ensuite, nous allons introduire les différents outils que nous avons déployé afin d'améliorer la qualité de notre application, surtout du côté organisation du code et développement FrontEnd. Nous allons progresser par la présentation générale de notre application avec les différents triggers, fonctions et procédures utilisées, l'établissement de la possibilité d'échange des emails et la présentation des différents espaces conçus pour les différents types des utilisateurs (Etudiant, chef de filière et administration). Puis, nous allons citer les tests que nous avons effectués sur les différents outils utilisés. Ensuite, nous allons vous présenter les différentes fonctionnalités que nous avons ajouté et qui sortent du cahier de charge suivie d'un chapitre consacré pour la présentation des étapes que nous avons suivi afin de pouvoir héberger notre application sur un serveur réel. Et nous allons terminer par une conclusion suivie d'un planning des tâches réalisées dans le cadre du projet et les différentes références qui nous ont été utiles pour délivrer notre produit.



Chapitre 1

Installations et outils
déployés pour la réalisation
de l'application

1-Installation PHP et ses packages sous Ubuntu :



Avant de commencer les installations, on cherche les updates :

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt update
[sudo] password for maysae2710:
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Installation de **PHP**(La dernière version disponible) :

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt install php
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.1
 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap php-common
 php8.1 php8.1-cli php8.1-common php8.1-opcache php8.1-readline
Suggested packages:
 apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
 php-pear
The following NEW packages will be installed:
 apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.1
 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap php php-common
 php8.1 php8.1-cli php8.1-common php8.1-opcache php8.1-readline
0 upgraded, 16 newly installed, 0 to remove and 84 not upgraded.
Need to get 7,022 kB of archives.
After this operation, 29.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libapr1 amd64 1.7.0-8build1 [107 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4 [92.4 kB]
Get:3 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4 [11.3 kB]
Get:4 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4 [9,162 B]
Get:5 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 apache2-bin amd64 2.4.52-1ubuntu4 [1,344 kB]
Get:6 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 apache2-data all 2.4.52-1ubuntu4 [165 kB]
Get:7 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 apache2-utils amd64 2.4.52-1ubuntu4 [88.8 kB]
Get:8 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 apache2 amd64 2.4.52-1ubuntu4 [97.8 kB]
Get:9 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:92ubuntu1 [12.4 kB]
Get:10 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 php8.1-common amd64 8.1.2-1ubuntu2 [1,123 kB]
Get:11 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 php8.1-opcache amd64 8.1.2-1ubuntu2 [365 kB]
Get:12 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 php8.1-readline amd64 8.1.2-1ubuntu2 [13.5 kB]
Get:13 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 php8.1-cli amd64 8.1.2-1ubuntu2 [1,825 kB]
```

Installation des packages : **Php8.1-mbstring**, **php8.1-xml** et **php8.1-zip** qui permettent, successivement d'identifier les texte sous format non Ascii, utiliser le module XML et sauvegarder les fichiers ou dossiers compressés :

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt install php8.1-mbstring php8.1-xml php8.1-zip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbonig5 libzip4
The following NEW packages will be installed:
  libbonig5 libzip4 php8.1-mbstring php8.1-xml php8.1-zip
0 upgraded, 5 newly installed, 0 to remove and 84 not upgraded.
Need to get 857 kB of archives.
After this operation, 2,465 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libbonig5 amd64 6.9.7.1-2build1 [172 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu jammy/universe amd64 libzip4 amd64 1.7.3-1ubuntu2 [55.0 kB]
Get:3 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 php8.1-mbstring amd64 8.1.2-1ubuntu2 [484 kB]
Get:4 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 php8.1-xml amd64 8.1.2-1ubuntu2 [118 kB]
Get:5 http://ma.archive.ubuntu.com/ubuntu jammy/universe amd64 php8.1-zip amd64 8.1.2-1ubuntu2 [27.1 kB]
Fetched 857 kB in 1s (1,160 kB/s)
Selecting previously unselected package libbonig5:amd64.
(Reading database ... 196138 files and directories currently installed.)
Preparing to unpack .../libbonig5_6.9.7.1-2build1_amd64.deb ...
Unpacking libbonig5:amd64 (6.9.7.1-2build1) ...
Selecting previously unselected package libzip4:amd64.
Preparing to unpack .../libzip4_1.7.3-1ubuntu2_amd64.deb ...
Unpacking libzip4:amd64 (1.7.3-1ubuntu2) ...
Selecting previously unselected package php8.1-mbstring.
Preparing to unpack .../php8.1-mbstring_8.1.2-1ubuntu2_amd64.deb ...
Unpacking php8.1-mbstring (8.1.2-1ubuntu2) ...
Selecting previously unselected package php8.1-xml.
Preparing to unpack .../php8.1-xml_8.1.2-1ubuntu2_amd64.deb ...
```

Installation de **Curl**(il s'agit d'une extension qui permet l'échange des fichiers sous les protocoles HTTP et FTP) :

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl4
1 upgraded, 1 newly installed, 0 to remove and 83 not upgraded.
Need to get 194 kB/484 kB of archives.
After this operation, 454 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.2 [194 kB]
Fetched 194 kB in 1s (209 kB/s)
(Reading database ... 196186 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.81.0-1ubuntu1.2_amd64.deb ...
Unpacking libcurl4:amd64 (7.81.0-1ubuntu1.2) over (7.81.0-1) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.81.0-1ubuntu1.2_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.2) ...
Setting up libcurl4:amd64 (7.81.0-1ubuntu1.2) ...
Setting up curl (7.81.0-1ubuntu1.2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
```

2-Installation d'Apache2:



Installation via la commande sudo apt install **Apache2** :

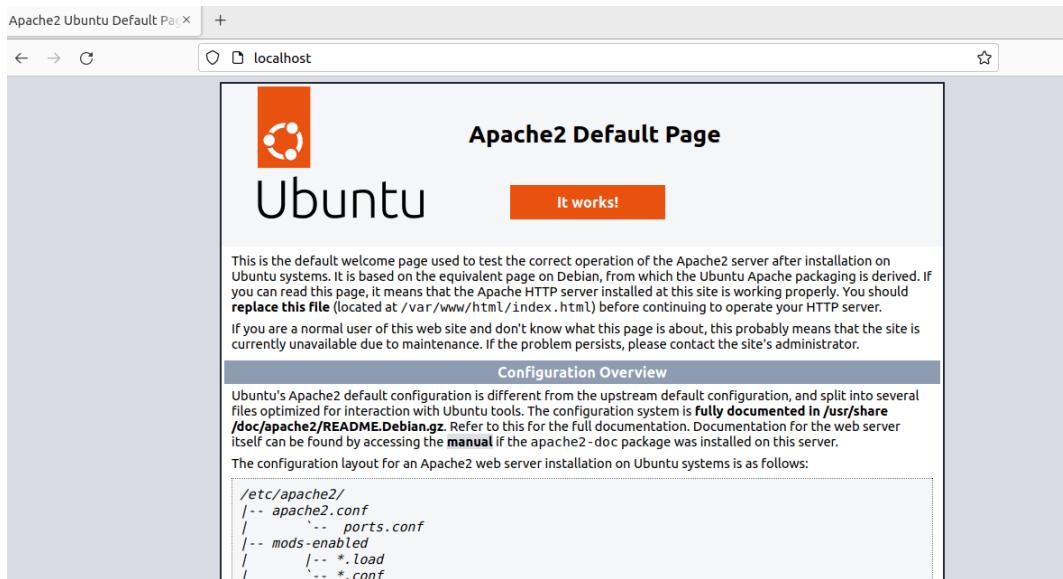
```
maysae2710@maysae2710-virtual-machine:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4).
```

Vérification du status : Apache2 status :

```
maysae2710@maysae-vm:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pres>
  Active: active (running) since Sun 2022-05-08 20:40:06 +01; 50s ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 1017 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
 Main PID: 1062 (apache2)
    Tasks: 55 (limit: 7605)
   Memory: 7.8M
      CPU: 97ms
     CGroup: /system.slice/apache2.service
             ├─1062 /usr/sbin/apache2 -k start
             ├─1064 /usr/sbin/apache2 -k start
             └─1065 /usr/sbin/apache2 -k start

20:40:06 08 مـ naysae-vm systemd[1]: Starting The Apache HTTP Server...
20:40:06 08 مـ naysae-vm apachectl[1037]: AH00558: apache2: Could not reliably>
20:40:06 08 مـ naysae-vm systemd[1]: Started The Apache HTTP Server.
lines 1-17/17 (END)
```

On tape localhost dans la barre d'adresse du navigateur :



Nous remarquons que dans notre cas toutes les configurations nécessaires sont faites par défaut et apache prend le port 80 .

3-Installation, configuration d'Oracle et création d'un utilisateur :



Pré-installation :

Avant d'installer Oracle, nous avons besoin d'effectuer des pré-Installations:

On télécharge oracle 11g à partir du site officiel d'Oracle www.oracle.com.

L'extraction du fichier zip :

```
maysae2710@maysae2710-virtual-machine:~$ cd Downloads
maysae2710@maysae2710-virtual-machine:~/Downloads$ unzip oracle-xe-11.2.0-1.0.x86_64.rpm.zip
Archive: oracle-xe-11.2.0-1.0.x86_64.rpm.zip
  creating: Disk1/
  creating: Disk1/upgrade/
  inflating: Disk1/upgrade/gen_inst.sql
  creating: Disk1/response/
  inflating: Disk1/response/xe.rsp
  inflating: Disk1/oracle-xe-11.2.0-1.0.x86_64.rpm
```

Installation de alien libaio1 unixodbc

```
maysae2710@maysae2710-virtual-machine:~/Downloads$ sudo apt-get install alien libaio1 unixodbc
[sudo] password for maysae2710:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  autoconf automake autopoint autotools-dev binutils binutils-common binutils-x86-64-linux-gnu build-essential debhelper debugedit
  dh-autoreconf dh-strip-nondeterminism dpkg-dev dwz fakeroot g++-4.11 gcc gcc-11 gettext inttool-debian libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libarchive-cpio-perl libarchive-zip-perl libasan6 libatomici libbinutils libc-dev-bin
  libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdebsupport-perl libdpkg-perl libfakeroot libfile-fcntllock-perl
  libfile-stripnondeterminism-perl libfsverity0 libgcc-11-dev libitm1 liblsan0 libltdl-dev libmail-sendmail-perl libnsl-dev libodbc2
  libodbcinst2 libquadmath0 librpm9 librpmbuild9 librpmi09 librpmisign9 libsigsegv2 libstdc++-11-dev libsub-override-perl
  libsys-hostname-long-perl libtirpc-dev libtool libtsan0 libubsan1 linux-libc-dev lto-disabled-list m4 make manpages-dev po-debconf
  rpcsvc-proto rpm rpm-common rpm2cpio unixodbc-common
Suggested packages:
  lintian autoconf-archive gnu-standards autoconf-doc binutils-doc dh-make debian-keyring g++-multilib g++-11-multilib gcc-11-doc
  gcc-multilib flex bison gcc-doc gcc-11-multilib gcc-11-locales gettext-doc libasprintf-dev libgettextxpo-dev libgc-doc git bzr libtool-doc
  odbc-postgresql tdsodbc libstdc++-11-doc gfortran | fortran95-compiler gcj-jdk m4-doc make-doc libmail-box-perl elfutils rpm-lint rpm-i18n
The following NEW packages will be installed:
  alien autoconf automake autopoint autotools-dev binutils binutils-common binutils-x86-64-linux-gnu build-essential debhelper debugedit
  dh-autoreconf dh-strip-nondeterminism dpkg-dev dwz fakeroot g++-4.11 gcc gcc-11 gettext inttool-debian libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libarchive-cpio-perl libarchive-zip-perl libasan6 libatomici libbinutils libc-dev-bin
  libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdebsupport-perl libdpkg-perl libfakeroot libfile-fcntllock-perl
  libfile-stripnondeterminism-perl libfsverity0 libgcc-11-dev libitm1 liblsan0 libltdl-dev libmail-sendmail-perl libnsl-dev libodbc2
  libodbcinst2 libquadmath0 librpm9 librpmbuild9 librpmi09 librpmisign9 libsigsegv2 libstdc++-11-dev libsub-override-perl
  libsys-hostname-long-perl libtirpc-dev libtool libtsan0 libubsan1 linux-libc-dev lto-disabled-list m4 make manpages-dev po-debconf
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo alien --scripts -d oracle-xe-11.2.0-1.0.x86_64.rpm
oracle-xe_11.2.0-2_amd64.deb generated
```

Création de **scriptshell** pour la configuration de **oracle** et changement des droits de permission :

```
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo pico /sbin/chkconfig
[sudo] password for maysae2710:
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo chmod 755 /sbin/chkconfig
```

On entre des paramètres supplémentaires pour le noyau (**kernel unix**)

```
GNU nano 6.2                                         /etc/sysctl.d/60-oracle.conf *
# Oracle 11g XE kernel parameters
#fs.file-max=6815744
#net.ipv4.ip_local_port_range=9000 65000
fkernel.sem=250 32000 100 128
ikernel.shmmax=536870912
e
e
echo '# Provides: OracleXE' >> $file
echo '# Required-Start: $remote_fs $syslog' >> $file
echo '# Required-Stop: $remote_fs $syslog' >> $file
echo '# Default-Start: 2 3 4 5' >> $file
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo pico /etc/sysctl.d/60-oracle.conf
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo cat /etc/sysctl.d/60-oracle.conf
# Oracle 11g XE kernel parameters
fs.file-max=6815744
net.ipv4.ip_local_port_range=9000 65000
kernel.sem=250 32000 100 128
kernel.shmmax=536870912
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$
```

On vérifie si les modifications ont été bien établies:

On doit charger les paramètres kernel :

```
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo service procps start
```

On a bien la file-max valeur entrée au début, ce qui prouve le succès de

```
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo sysctl -q fs.file-max
fs.file-max = 9223372036854775807
```

l'installation :

Ensuite, on va créer un script Shell qu'on va utiliser pour l'installation du système des fichiers montés et on va changer les permissions d'accès au fichier :

```
GNU nano 6.2                                         /etc/rc2.d/S01shm_load *
#!/bin/sh
case "$1" in
start) mkdir /var/lock/subsys 2>/dev/null
        touch /var/lock/subsys/listener
        rm /dev/shm 2>/dev/null
        mkdir /dev/shm 2>/dev/null
        mount -t tmpfs shmfs -o size=2048m /dev/shm ;;
*) echo error
   exit 1 ;;
esac
```

On redémarre le système.

Installation Oracle :

```
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo dpkg --install oracle-xe_11.2.0-2_amd64.deb
[sudo] password for maysae2710:
Selecting previously unselected package oracle-xe.
(Reading database ... 203490 files and directories currently installed.)
Preparing to unpack oracle-xe_11.2.0-2_amd64.deb ...
Unpacking oracle-xe (11.2.0-2) ...
Setting up oracle-xe (11.2.0-2) ...
Executing post-install steps...
You must run '/etc/init.d/oracle-xe configure' as the root user to configure the database.

Processing triggers for libc-bin (2.35-0ubuntu3) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
```

On installe **Oracle DBMS**, en utilisant la commande suivante :

Configuration d'Oracle et création d'un utilisateur:

```
Oracle Database 11g Express Edition Configuration
-----
This will configure on-boot properties of Oracle Database 11g Express
Edition. The following questions will determine whether the database should
be starting upon system boot, the ports it will use, and the passwords that
will be used for database accounts. Press <Enter> to accept the defaults.
Ctrl-C will abort.
```

Specify the HTTP port that will be used for Oracle Application Express [8080]:

Port http : 8080

Port listener de la base de données : 1521

Specify a port that will be used for the database listener [1521]:

Configuration du mot de passe pour SYS et SYSTEM

```
Specify a password to be used for database accounts. Note that the same
password will be used for SYS and SYSTEM. Oracle recommends the use of
different passwords for each database account. This can be done after
initial configuration:
The password you entered contains invalid characters. Enter password:
Confirm the password:
```

L'installation s'est bien effectuée ...

```
Starting Oracle Net Listener...Done
Configuring database...Done
Starting Oracle Database 11g Express Edition instance...Done
Installation completed successfully.
```

Configuration des variables d'environnement en modifiant le fichier .bashrc et on charge les changements en exécutant le profile.

On ajoute des paramètres supplémentaires à la configuration d'Oracle à la fin du fichier.bashrc :

```
GNU nano 6.2                               /home/maysae2710/.bashrc *
# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$( [ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*\[\s*0\]/[<--> /'\'')'
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/xe
export ORACLE_SID=XE
export NLS_LANG=$ORACLE_HOME/bin/nls_lang.sh
export ORACLE_BASE=/u01/app/oracle
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export PATH=$ORACLE_HOME/bin:$PATH

^G Help      ^O Write Out   ^W Where Is   ^K Cut          ^T Execute   ^C Location   M-U Undo   M-A Set Mark   M-[ To B
^X Exit      ^R Read File   ^\ Replace    ^U Paste        ^J Justify   ^/ Go To Line M-E Redo   M-6 Copy    ^Q Where
```

```
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ pico ~/.bashrc
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ . ~/.profile
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ █
```

On démarre Oracle et on crée un utilisateur :

```
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo service oracle-xe start
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo usermod -a -G dba maysae2710
```

```
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sudo service oracle-xe start
maysae2710@maysae2710-virtual-machine:~/Downloads/Disk1$ sqlplus sys as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Tue May 17 23:54:21 2022

Copyright (c) 1982, 2011, Oracle. All rights reserved.

Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL>
```

On crée un utilisateur :

```
SQL> create user admin identified by 123456
      2 ;
User created.
```

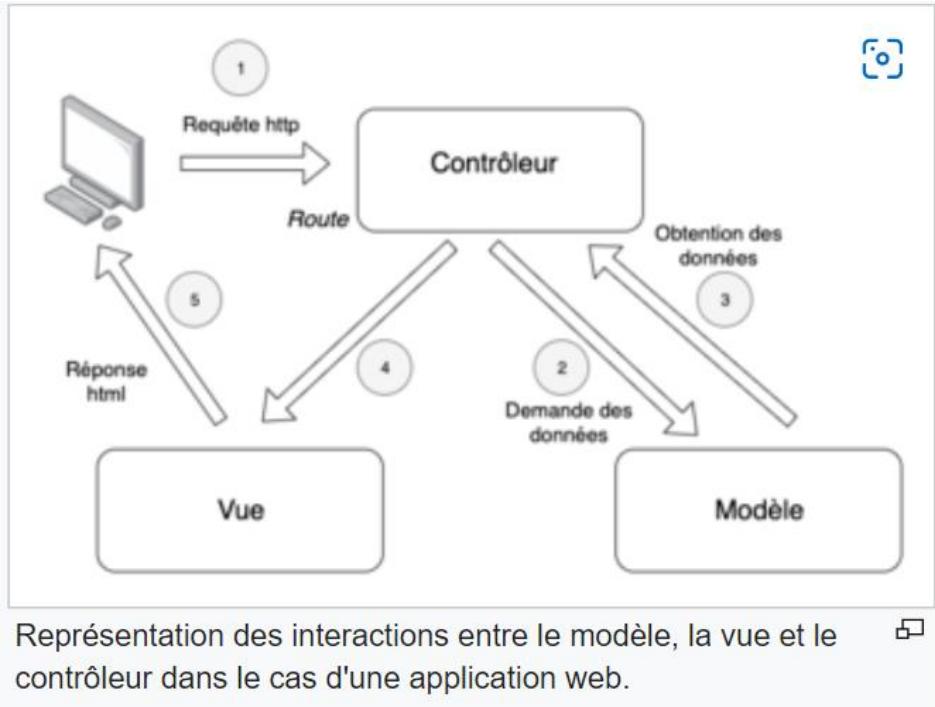
4-Installation du Composer et Laravel:

Afin de bien structurer notre code nous avons décidé de travailler avec Laravel : un Framework qui permet **PHP** multi-Plateforme permettant de créer des applications web. Il offre plusieurs avantages, tels que : une architecture MVC (Modèle, Vue et Contrôleur), une sécurité de haut niveau, une grande communauté et une documentation riche et valable en ligne et gratuitement , il offre également un outil de ligne de commande qui s'appelle Artisan et qui aide à créer une architecture de code « squelette » et de base de données, ainsi que leurs migrations. La gestion de base de données en devient beaucoup plus facile.

Mais avant d'installer **Laravel**, il nous faut un autre outil qui s'appelle **Composer** dont nous allons détailler le fonctionnement et l'installation dans ce qui suit. Nous allons également expliquer l'architecture MVC et son utilité pour notre application :

Le modèle MVC, ses avantages et ses limites :

Le modèle MVC décrit une manière d'architecturer d'une application informatique en la décomposant en trois sous-parties de manière à ce que plusieurs groupes de personnes puissent bosser chacun sur leur partie, sans même qu'ils connaissent les personnes qui bossent les autres parties : la partie Modèle, la partie Vue et la partie Contrôleur.



Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées.

Le MVC montre ses limites dans le cadre des applications utilisant les technologies du web, bâties à partir de serveurs d'applications. Des couches supplémentaires sont alors introduites ainsi que les mécanismes d'inversion de contrôle et d'injection de dépendances.

Instalation de Composer :

Installation du composer à partir du site officiel: Composer (getcomposer.org)



A Dependency Manager for PHP

Latest: **2.3.6** ([changelog](#))

Getting Started

[Download](#)

Documentation

Browse Packages

```
maysae2710@maysae2710-virtual-machine:~$ curl -s https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.3.5) successfully installed to: /home/maysae2710/composer.phar
Use it: php composer.phar
```

On vérifie que l'installation est bien effectuée par la commande **composer** :

On ajoute l'emplacement de **composer** dans les variables système :

```

GNU nano 6.2
.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *t*) ;;
    *) return;;
esac
export PATH="$PATH:$HOME/.config/composer/vendor/bin" ----->

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

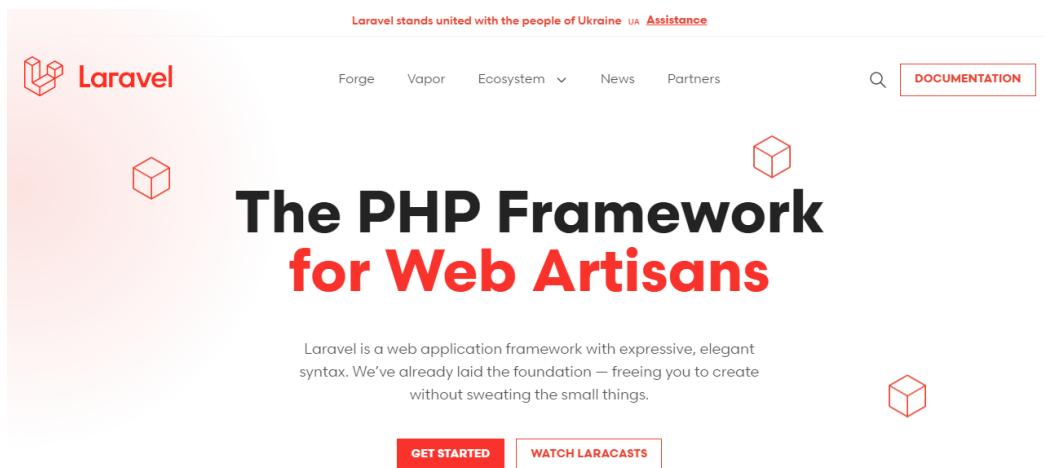
# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "/*" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

```

Installation de Laravel en utilisant Composer:

Site officiel: [Laravel - The PHP Framework For Web Artisans](https://laravel.com)



Dans documentation, on cherche l'installation via le **Composer** :

On utilise la commande : `composer global require laravel/installer`

```
maysae2710@maysae-vm:~/example-app$ composer global require laravel/installer
Changed current directory to /home/maysae2710/.config/composer
Info from https://repo.packagist.org: #StandWithUkraine
Using version ^4.2 for laravel/installer
./composer.json has been created
Running composer update laravel/installer
Loading composer repositories with package information
Updating dependencies
Lock file operations: 10 installs, 0 updates, 0 removals
- Locking laravel/installer (v4.2.10)
- Locking psr/container (2.0.2)
- Locking symfony/console (v6.0.8)
- Locking symfony/polyfill-ctype (v1.25.0)
- Locking symfony/polyfill-intl-grapheme (v1.25.0)
- Locking symfony/polyfill-intl-normalizer (v1.25.0)
- Locking symfony/polyfill-mbstring (v1.25.0)
- Locking symfony/process (v6.0.8)
- Locking symfony/service-contracts (v3.0.1)
- Locking symfony/string (v6.0.8)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 10 installs, 0 updates, 0 removals
- Downloading symfony/process (v6.0.8)
```

Puis on crée un dossier Laravel et on vérifie que **Laravel** s'est bien installé en créant un nouveau projet:

```
maysae2710@maysae-vm:~/example-app$ mkdir Laravel
maysae2710@maysae-vm:~/example-app$ cd Laravel
maysae2710@maysae-vm:~/example-app/Laravel$ laravel new newprojet

Creating a "laravel/laravel" project at "./newprojet"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v9.1.7)
- Installing laravel/laravel (v9.1.7): Extracting archive
Created project in /home/maysae2710/example-app/Laravel/newprojet
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
```

5- Installation et configuration de Nginx:



Suppression d' Apache2 :

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt-get purge apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  apache2-data apache2-utils
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  apache2*
0 upgraded, 0 newly installed, 1 to remove and 11 not upgraded.
After this operation, 545 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 207634 files and directories currently installed.)
Removing apache2 (2.4.52-1ubuntu4) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4build1) ...
Rules updated for profile 'Nginx HTTP'
Firewall reloaded
(Reading database ... 207584 files and directories currently installed.)
Purging configuration files for apache2 (2.4.52-1ubuntu4) ...
Processing triggers for ufw (0.36.1-4build1) ...
Rules updated for profile 'Nginx HTTP'
```

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt-get autoclean
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Del gnome-control-center-data 1:41.4-1ubuntu13 [344 kB]
Del gnome-control-center 1:41.4-1ubuntu13 [1,785 kB]
Del gnome-control-center-faces 1:41.4-1ubuntu13 [1,218 kB]
maysae2710@maysae2710-virtual-machine:~$ sudo apt-get autoremove
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  apache2-data apache2-utils
0 upgraded, 0 newly installed, 2 to remove and 11 not upgraded.
After this operation, 1,281 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 207421 files and directories currently installed.)
Removing apache2-data (2.4.52-1ubuntu4) ...
Removing apache2-utils (2.4.52-1ubuntu4) ...
Processing triggers for man-db (2.10.2-1) ...
maysae2710@maysae2710-virtual-machine:~$
```

Pour installer **Nginx**, on commence par faire un update du gestionnaire de packages

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt update
[sudo] password for maysae2710:
Hit:1 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
11 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

On installe **Nginx** et toutes les dépendances requises sur notre serveur :

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx-common nginx-core
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx nginx-common nginx-core
0 upgraded, 9 newly installed, 0 to remove and 11 not upgraded.
Need to get 696 kB of archives.
After this operation, 2,395 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nginx-common
all 1.18.0-6ubuntu14.1 [40.1 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-
http-geoip2 amd64 1.18.0-6ubuntu14.1 [11.9 kB]
```

Nous devons faire une configuration de notre logiciel de pare-feu pour autoriser l'accès au service.

Nginx s'enregistre en tant que service avec **ufw** afin d'en faciliter l'accès.

Énumérons les configurations d'application avec lesquelles **ufw** sait travailler en saisissant :

```
maysae2710@maysae2710-virtual-machine:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
```

Comme le montre le résultat, il existe trois profils pour **Nginx** :

- **Nginx Full** : Ce profil ouvre à la fois le port 80 (trafic web normal, non crypté) et le port 443 (trafic crypté TLS/SSL).

- **Nginx HTTP** : Ce profil n'ouvre que le port 80 (trafic web normal, non crypté).

- **Nginx HTTPS** : Ce profil n'ouvre que le port 443 (trafic crypté TLS/SSL).

Nous avons choisis d'activer **Nginx http** :

```
maysae2710@maysae2710-virtual-machine:~$ sudo ufw allow 'Nginx HTTP'
Rules updated
Rules updated (v6)
```

Activation de **ufw** et vérification de l'établissement des changements :

```
maysae2710@maysae2710-virtual-machine:~$ sudo ufw enable
Firewall is active and enabled on system startup
maysae2710@maysae2710-virtual-machine:~$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
Nginx HTTP                  ALLOW       Anywhere
Nginx HTTP (v6)              ALLOW       Anywhere (v6)
```

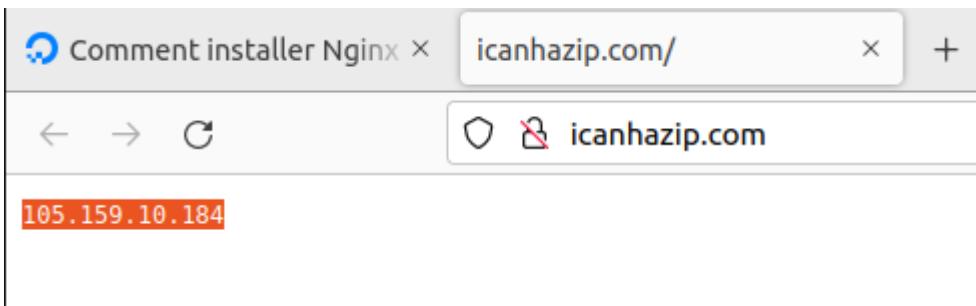
On vérifie le fonctionnement du serveur :

```
maysae2710@maysae2710-virtual-machine:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-05-19 21:23:36 +01; 4min 7s ago
     Docs: man:nginx(8)
  Process: 5940 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process
  Process: 5941 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (c>
 Main PID: 5942 (nginx)
    Tasks: 5 (limit: 9419)
   Memory: 5.3M
      CPU: 63ms
     CGroup: /system.slice/nginx.service
             ├─5942 "nginx: master process /usr/sbin/nginx -g daemon on; master_p>
             ├─5943 "nginx: worker process" " " " " " " " " " " " " " " " " " " >
             ├─5944 "nginx: worker process" " " " " " " " " " " " " " " " " " " >
             ├─5945 "nginx: worker process" " " " " " " " " " " " " " " " " " " >
             └─5946 "nginx: worker process" " " " " " " " " " " " " " " " " " " >

21:23:36 19 مـ maysae2710-virtual-machine systemd[1]: Starting A high performan>
21:23:36 19 مـ maysae2710-virtual-machine systemd[1]: Started A high performan>
lines 1-19/19 (END)
```

Comme cela le confirme, le service a bien démarré. Cependant, la meilleure façon de le vérifier consiste à demander une page à partir de **Nginx** :

En utilisant le service : <http://icanhazip.com/> , nous trouvons notre adresse ip :



Et la page par défaut de **Nginx** :



6-Installation et configuration du serveur DNS:

Installation de bind9 :

```
maysae2710@maysae2710-virtual-machine:~$ sudo apt install bind9
[sudo] password for maysae2710:
Reading package lists... Done
Building dependency tree... Done
Processing triggers for syslog (3.2.1-12.0+deb9u22.2) ...
maysae2710@maysae2710-virtual-machine:~$
```

Installation de DNSUtilities :

```
abderrahimfizazi@abderrahimfizazi:~$ sudo apt-get install dnsutils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dnsutils is already the newest version (1:9.18.1-1ubuntu1.1).
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
abderrahimfizazi@abderrahimfizazi:~$
```

Configuration sur /etc/bind/named.con :

```
abderrahimfizazi@abderrahimfizazi:~$ sudo nano /etc/bind/named.conf.options
abderrahimfizazi@abderrahimfizazi:~$
```

```
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
    };

    //========================================================================
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //================================================================
    dnssec-validation auto;

[ Read 24 lines ]
^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line
```

Après la saisie de tous les paramètres de configuration nous allons redémarrer le serveur pour les activer :

```
abderrahimfizazi@abderrahimfizazi:~$ sudo systemctl restart bind9
abderrahimfizazi@abderrahimfizazi:~$
```

Test du temps de réponse des requêtes, en utilisant dig google.com :

```
abderrahimfizazi@abderrahimfizazi:~$ dig google.com

; <>> DiG 9.18.1-1ubuntu1.1-Ubuntu <>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45970
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.           IN      A

;; ANSWER SECTION:
google.com.        70      IN      A      142.250.200.78

;; Query time: 212 msec
;; SERVER: 192.168.137.1#53(192.168.137.1) (UDP)
;; WHEN: Wed Jun  8 17:33:02 WEST 2022
;; MSG SIZE  rcvd: 55

abderrahimfizazi@abderrahimfizazi:~$
```

L'ajout d'une zone DNS dans /etc/bind/named.conf.local

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
};
```

Pour activer les nouvelles modifications, on redémarre le serveur et on effectue les tests de checkzone :

```

abderrahimfizazi@abderrahimfizazi:~$ sudo nano /etc/bind/named.conf.local
abderrahimfizazi@abderrahimfizazi:~$ sudo systemctl restart bind9
abderrahimfizazi@abderrahimfizazi:~$ named-checkzone example.com /etc/bind/db.example.com
zone example.com/IN: loaded serial 2
OK
abderrahimfizazi@abderrahimfizazi:~$ named-checkzone 192.168.0.0/32 /etc/bind/db.10
zone 192.168.0.0/32/IN: loaded serial 1
OK
abderrahimfizazi@abderrahimfizazi:~$ named-checkconf /etc/bind/named.conf.local
abderrahimfizazi@abderrahimfizazi:~$ named-checkconf /etc/bind/named.conf
abderrahimfizazi@abderrahimfizazi:~$ █

```

On remarque que tous les tests sont bien effectués et notre serveur est bien configuré.

7-Installation et configuration de PostFix:

Installation :

```

maysae2710@maysae2710-virtual-machine:~$ sudo apt install postfix
[sudo] password for maysae2710:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libblockfile-bin libblockfile1 lockfile-progs sendmail-base sendmail-cf
  sensible-mda
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  postfix-mysql postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb
  postfix-sqlite sasl2-bin | dovecot-common resolvconf postfix-cdb
  postfix-mta-sts-resolver postfix-doc
The following packages will be REMOVED:

```

```

abderrahimfizazi@abderrahimfizazi:~$ dig google.com

; <>> DiG 9.18.1-1ubuntu1.1-Ubuntu <>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45970
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.           IN      A

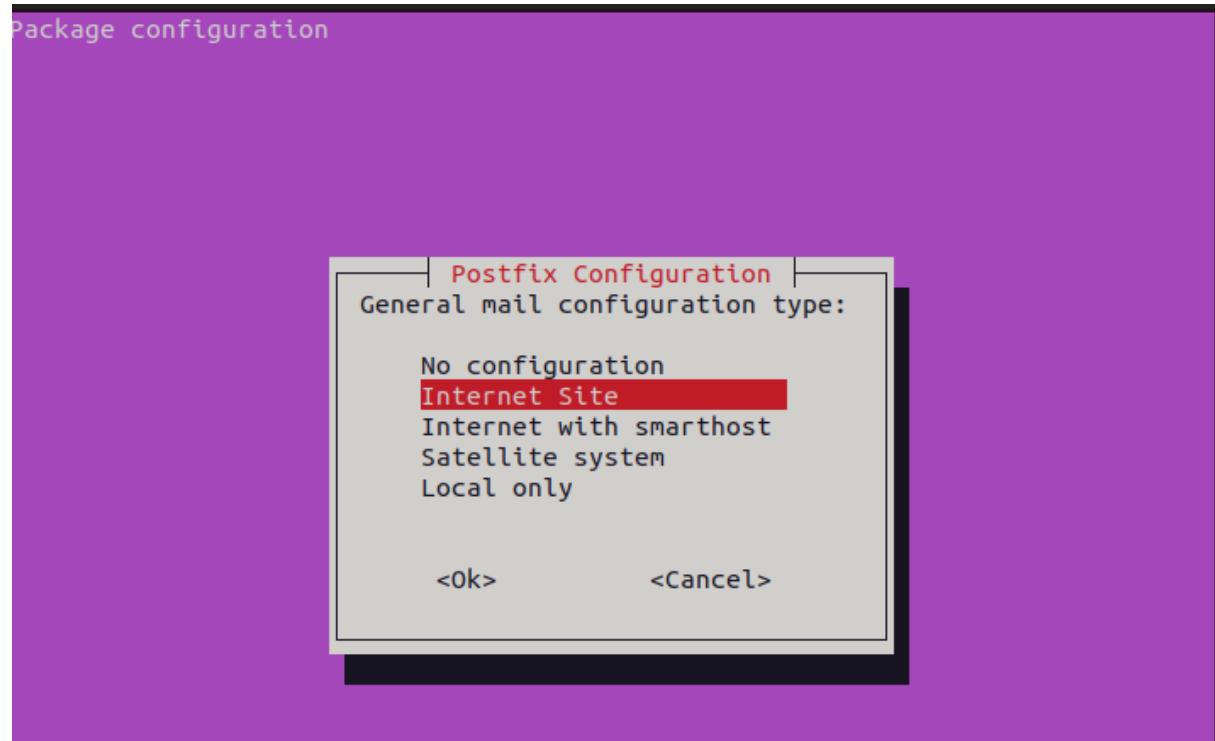
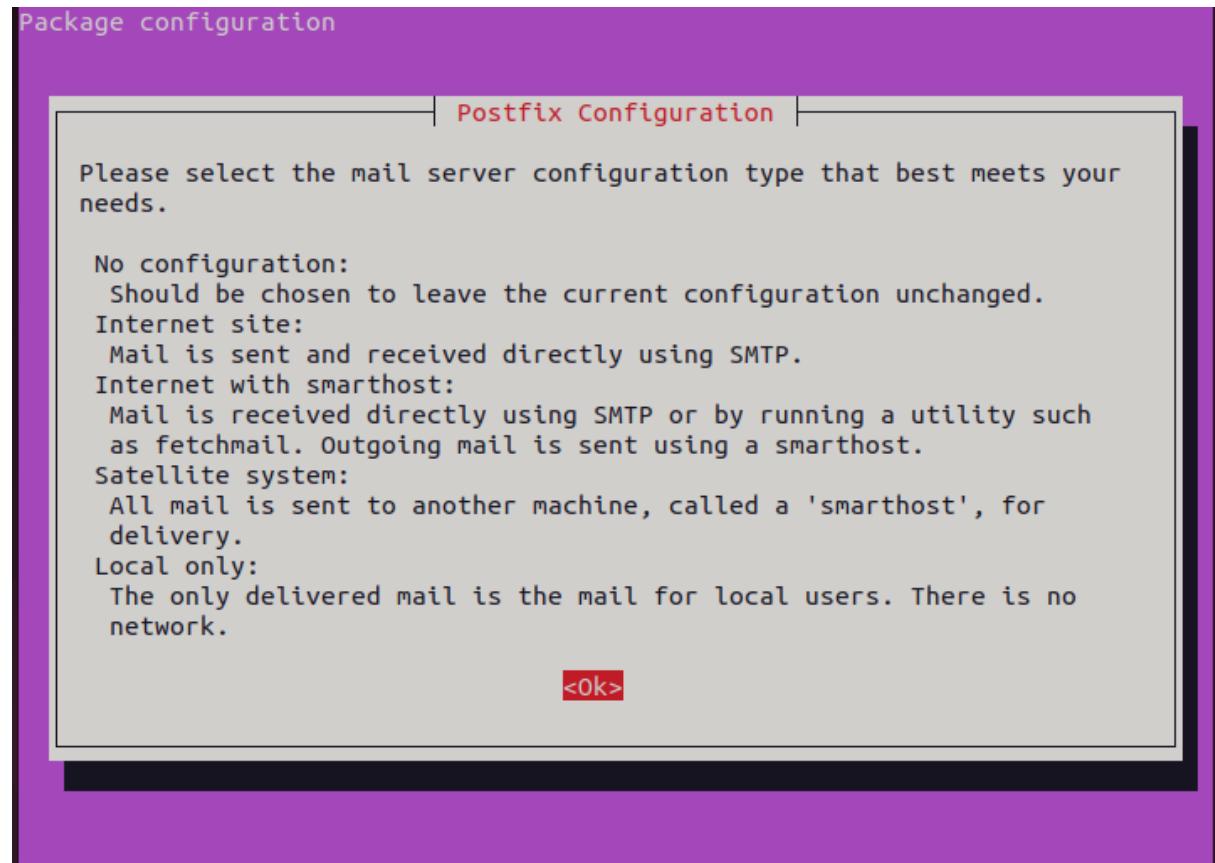
;; ANSWER SECTION:
google.com.        70      IN      A      142.250.200.78

;; Query time: 212 msec
;; SERVER: 192.168.137.1#53(192.168.137.1) (UDP)
;; WHEN: Wed Jun  8 17:33:02 WEST 2022
;; MSG SIZE  rcvd: 55

abderrahimfizazi@abderrahimfizazi:~$ █

```

Configuration :



Package configuration

Postfix Configuration
The 'mail name' is the domain name used to 'qualify' _ALL_ mail addresses without a domain name. This includes mail to and from <root>; please do not make your machine send out mail from root@example.org unless root@example.org has told you to.

This name will also be used by other programs. It should be the single, fully qualified domain name (FQDN).

Thus, if a mail address on the local host is foo@example.org, the correct value for this option would be example.org.

System mail name:

maysae2710-virtual-machine

<Ok>

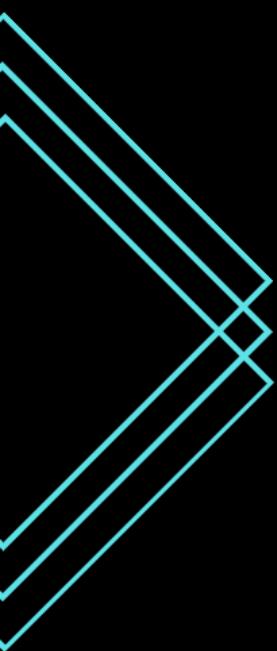
<Cancel>

Configuration supplémentaire sur le fichier de configuration :

```
# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CApth=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = abderrahimfizazi
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, abderrahimfizazi, localhost.localdomain, , localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_tls_CAfile = /etc/postfix/cacert.pem
smtp_use_tls = yes
```



Chapitre 2

Autres outils déployés pour
une meilleure qualité de
l'application

1- Postman:

Présentation générale :



Postman est une plate-forme API pour la création et l'utilisation d'API. Il simplifie chaque étape du cycle de vie des API et rationalise la collaboration afin de créer de meilleures API plus rapidement. Il permet également de tester des services Web que ce soit pour du développement PHP, Node.js, Ruby on Rails ou Python.

Fonctionnement :

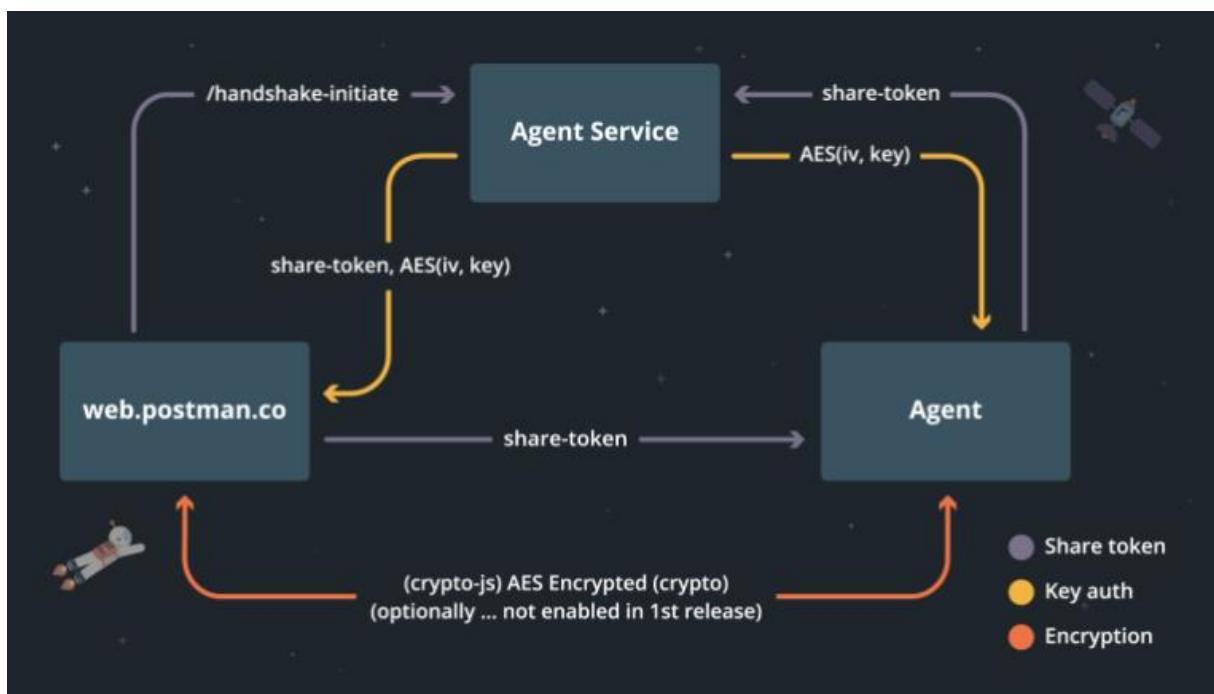
Parmi les nombreuses solutions pour interroger ou tester webservices et API, Postman propose de nombreuses fonctionnalités, une prise en main rapide et une interface graphique agréable.

Il permet de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en Collections.

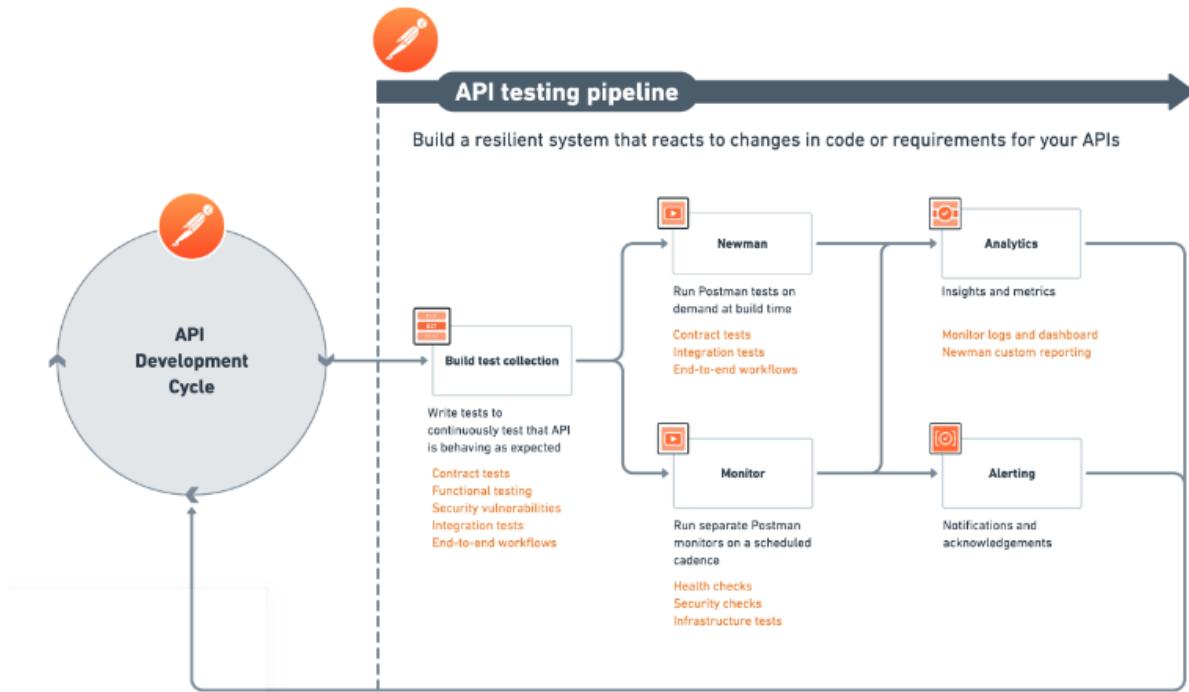
Les requêtes http représentent la fonctionnalité la plus triviale à laquelle répond cet outil. L'interface graphique rend l'exercice aussi facile que surfer internet sur un navigateur moderne. Il suffit de renseigner l'URL et la méthode http : Get, Post, Put, Patch et Delete.

L'interface de Postman :

Architecture et cycle de développement d'une Api :
L'illustration suivante montre l'architecture de l'outil Postman :



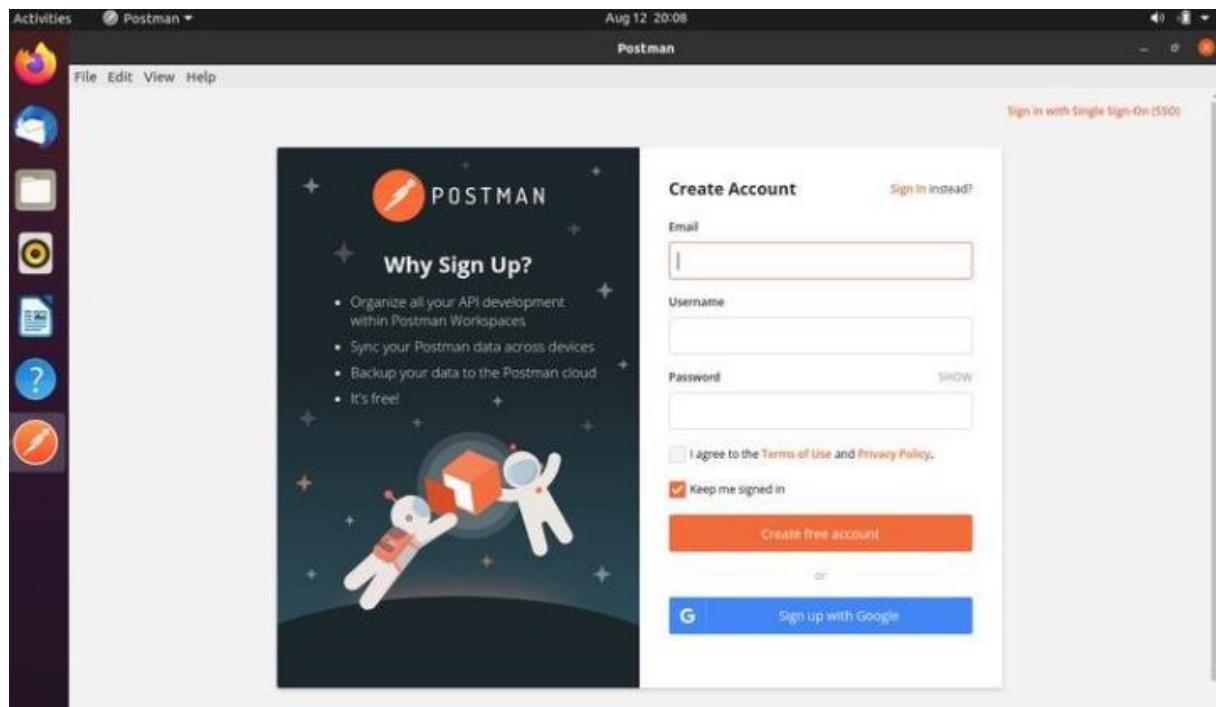
Les différentes étapes du cycle de développement d'une API :



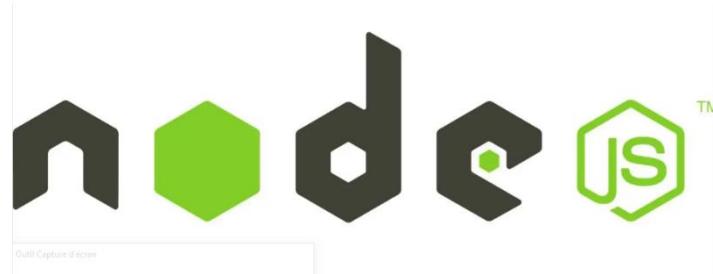
Installation :

```
maysae2710@maysae2710-virtual-machine:~$ sudo snap install postman
[sudo] password for maysae2710:
postman (v9/stable) 9.19.0 from Postman, Inc. (postman-inc✓) installed
```

On vérifie que Postman s'est bien installé en ouvrant la page principale:



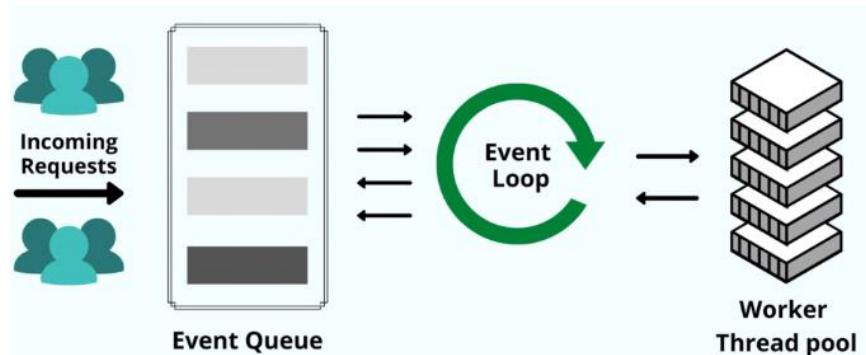
2- NodeJS:



Présentation générale:

Node.js est un environnement d'exécution open source basé sur JavaScript. Il est construit sur le V8 de Chrome et utilise un modèle d'entrée/sortie piloté par les événements, qui est responsable de son efficacité et de son évolutivité. Introduit en 2009 lors de la JSConf européenne, Node.js permet aux développeurs d'exécuter rapidement des codes en utilisant une boucle d'événement monofilaire.

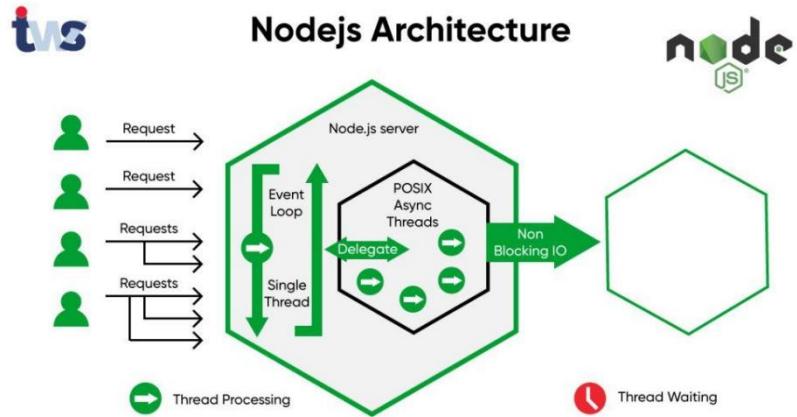
Dans une boucle d'événement monofilaire, le serveur et le client peuvent travailler simultanément sur les demandes ou les informations grâce à un libre échange de données :



Architecture :

Node.js utilise l'architecture « Single Threaded Event Loop » pour gérer plusieurs clients en même temps. Pour comprendre en quoi cela est différent des autres runtimes, nous devons comprendre comment les clients concurrents multi-threads sont gérés dans des langages comme Java.

Dans un modèle requête-réponse multi-thread, plusieurs clients envoient une requête, et le serveur traite chacune d'entre elles avant de renvoyer la réponse. Cependant, plusieurs threads sont utilisés pour traiter les appels simultanés. Ces threads sont définis dans un pool de threads, et chaque fois qu'une requête arrive, un thread individuel est affecté à son traitement.



L'utilisation du Node package manager (NPM) avec Laravel :

Bien que Laravel ne dicte pas les préprocesseurs JavaScript ou CSS que vous utilisez, il fournit un point de départ de base utilisant Bootstrap et Vue qui sera utile pour de nombreuses applications. Par défaut, Laravel utilise NPM pour installer ces deux packages frontaux.

Installation de Node et NPM:

Nous devons d'abord appeler le package d'interaction avec l'utilisateur de Laravel :

```
PS C:\Users\hp\OneDrive\Bureau\LARAVEL\projet> composer require laravel/ui
Using version ^3.4 for laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking laravel/ui (v3.4.6)
```

On installe Node à partir du site officiel : [Download | Node.js \(nodejs.org\)](https://nodejs.org)

The screenshot shows the Node.js website's download section. At the top, there are two tabs: 'LTS Recommended For Most Users' (green background) and 'Current Latest Features' (light green background). Below these are download links for Windows, macOS, and Source Code. A table provides download links for Windows, macOS, and Source Code in both 32-bit and 64-bit formats.

	32-bit	64-bit
Windows Installer (.msi)	node-v16.15.1-x64.msi	node-v16.15.1-x64.msi
Windows Binary (.zip)	node-v16.15.1.zip	node-v16.15.1.zip
macOS Installer (.pkg)	node-v16.15.1.pkg	node-v16.15.1.pkg
	64-bit / ARM64	64-bit / ARM64

Installation de npm :

```
PS C:\Users\hp\OneDrive\Bureau\LARAVEL\projet> npm install
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
added 746 packages, and audited 747 packages in 41s
76 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 8.8.0 → 8.12.1
```

Vérification de l'installation de node :

```
PS C:\Users\hp\OneDrive\Bureau\LARAVEL\projet> node -v
v18.1.0
```

Compiler tous les éléments actif : npm run dev

```
PS C:\Users\hp\OneDrive\Bureau\LARAVEL\projet> npm run dev
> dev
> npm run development

> development
> mix

  Additional dependencies must be installed. This will only take a moment.

  Running: npm install resolve-url-loader@^5.0.0 --save-dev --legacy-peer-deps

  Finished. Please run Mix again.
```

Et on peut commencer le travail avec NodeJs sur Laravel !

3-React Js :



Présentation générale:

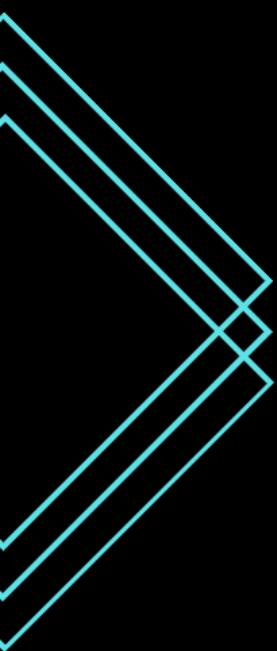
React est une bibliothèque permettant de construire des interfaces utilisateur composites. Elle encourage la création de composants d'interface utilisateur réutilisables, qui présentent des données qui évoluent dans le temps. Beaucoup de gens utilisent React comme le V de MVC (Modèle-Vue-Contrôleur).

React fait abstraction du DOM, offrant un modèle de programmation plus simple et de meilleures performances. React peut également effectuer un rendu sur le serveur à l'aide de Node, et il peut alimenter des applications natives à l'aide de React Native.

React met en œuvre un flux de données réactif unidirectionnel, ce qui réduit le nombre d'erreurs et facilite le raisonnement par rapport à la liaison de données traditionnelle.

Intérêt de son utilisation :

Nous avons décidé d'utiliser React vu sa grande simplicité et le cadre professionnel qui nous offre: ReactJS est tout simplement plus simple à saisir tout de suite. L'approche basée sur les composants, le cycle de vie bien défini et l'utilisation de JavaScript simple rendent React très simple à apprendre, à construire un site web professionnel (et des applications mobiles) et à le prendre en charge. React utilise une syntaxe spéciale appelée JSX qui vous permet de mélanger HTML et JavaScript. Ce n'est pas une obligation ; le développeur peut toujours écrire en JavaScript simple, mais JSX est beaucoup plus facile à utiliser.



Chapitre 3

Présentation de l'application

1-Les Procédeurs nécessaires au fonctionnement de l'application:

Création des procédures (Controllers) sous Laravel:

Php artisan make:controller API/controller_name

(en creant automatiquement un dossier avec le API pour faciliter le travail)

Ex:

Php artisan make:controller API/EleveController

Php artisan make:controller API>NoteController

Ces contrôleurs sont des classes qui contiennent des fonctions qui permettent de gérer toutes les fonctionnalités de l'application telles que : l'indexation, l'enregistrement, la suppression et la modification des données de toutes les parties intervenantes.

De base, ces fonctions sont utilisées pour tous les contrôleurs et elles ont comme appellation générale **CRUD**:

Store() : permet de créer un enregistrement depuis les informations passées par une requête.

Index() : pour l'affichage des données liées à l'élément représenté par le modèle lié au contrôleur correspondant. Ex : index() dans Eleve_controller affiche les informations d'un élève.

Delete() : pour supprimer les données, elle prend comme argument l'id correspondant à l'enregistrement à supprimer, s'il existe bien sûr.

Edit() : permet de récupérer toutes les données liées à un enregistrement, elle prend comme argument l'id correspondant à l'enregistrement à supprimer, s'il existe bien sûr.

Update() : permet de modifier les informations d'un enregistrement, elle fait appel à edit().

Cependant, souvent, nous avons besoin de d'autres fonctions selon le besoin.

Fonctions pour gérer l'authentification:

```
App / Controllers / API / AuthController.php
public function login(Request $request){
    $validator = Validator::make($request->all(), [
        'login' => 'required|max:191',
        'password' => 'required|min:8|max:191'
    ]);
    if($validator->fails()){
        return response()->json([
            "validation_errors" => $validator->messages(),
            "status" => 401
        ]);
    }
    else {
        $user = User::where('login', $request->login)->first();
        if (! $user || !($request->password === $user->password)) {
            return response()->json([
                "status" => 403,
                "message" => "invalid Credentials"
            ]);
        }
        else{
            if($user->role == 0){

                $token = $user->createToken($user->login.'EleveToken', [ '' ])->plainTextToken;
                $role = 'eleve';

            }
            elseif($user->role == 1){
                $token = $user->createToken($user->login.'Responsable_Token',[ 'server:respo' ])->plainTextToken;
                $role = 'respo';
            }
            elseif($user->role == 999){
                $token = $user->createToken($user->login.'Responsable_Token',[ 'server:admin' ])->plainTextToken;
                $role = 'Admin';
            }
        }
        if($user->verified == 0){
            return response()->json([
                'status' => 201,
                'login' =>$user->login,
                'role' => $role,
                'token'=>$token,
                'message' => 'Login successfully',
                'verified' => 0
            ]);
        }
        else{
            return response()->json([
                'status' => 200,
                'login' =>$user->login,
                'role' => $role,
                'token'=>$token,
                'message' => 'Login successfully',
            ]);
        }
    }
}
```

```

protected function destroy($id){
    $user = user::find($id);
    if($user){
        $user->delete();
        return response()->json([
            "message" => "user Deleted successfully",
            "status" => 200
        ]);
    }
    else{
        return response()->json([
            "message" => "user Not found",
            "status" => 404
        ]);
    }
}

protected function index(){
    $eleve = User::has('eleve')->get();
    $respo = User::has('responsable_filiere')->get();

    return response()->json([
        "status" => 200,
        "eleve" => $eleve,
        "responsable_filiere" => $respo
    ]);
}

protected function getLastId(){
    $last = User::latest('id')->select("id")->first();
    return response()->json([
        "status" => 200,
        "user_id" => $last
    ]);
}

```

```

protected function change(Request $request){
    $validator = Validator::make($request->all(), [
        "oldPassword" => "required| max:191|min:8",
        "newPassword" => "required| max:191|min:8",
    ]);
    if($validator->fails()){
        return response()->json([
            "status" => 400,
            "errors" => $validator->messages(),
        ]);
    }
    else{
        if($request->oldPassword == auth()->user()->password){
            if($request->oldPassword == $request->newPassword ){
                return response()->json([
                    "status" => 401,
                    "message" => "plz set a different password"
                ]);
            }
            else{
                $user = auth()->user();
                $user->fill([
                    'password' => $request->newPassword,
                    'verified' => 1
                ])->save();

                return response()->json([
                    "status" => 200,
                    "message" => "Password changed succesfully"
                ]);
            }
        }
        return response()->json([
            "status" => 403,
            "message" => "Invalid password"
        ]);
    }
}

```

Fonctions pour gérer les étudiants :

```
}

protected function destroy($id){
    $eleve = Eleve::find($id);
    if[$eleve]{
        $eleve->delete();
        return response()->json([
            "message" => "Eleve Deleted successfully",
            "status" => 200
        ]);
    }
    else{
        return response()->json([
            "message" => "Eleve Not found",
            "status" => 404
        ]);
    }
}

protected function index(){
    $eleve = eleve::All();
    return response()->json([
        "status" => 200,
        "eleve" => $eleve
    ]);
}

protected function get($filiere , $niveau){
    if($filiere === "All"){
        if($niveau == "All"){
            $eleve = eleve::All();
        }
        else{
            $eleve = eleve::where("niveau" , $niveau)->get();
        }
    }
    else{
        if($niveau == 'All'){
            $eleve = eleve::where("filiere" , $filiere)->get();
        }
        else{
            $eleve = eleve::where("filiere" , $filiere)->where("ni");
        }
    }
    return response()->json([
        "status" => 200,
        "eleve" => $eleve
    ]);
}

protected function search($search){
    $eleve = eleve::where("nom" , "LIKE" , '%' . $search . '%')->orwl;
    return response()->json([
        "status" => 200,
        "eleve" => $eleve
    ]);
}
```

```
protected function edit($id){
    $eleve = Eleve::find($id);
    if($eleve){
        return response()->Json([
            "status" => 200,
            "eleve" => $eleve,
        ]);
    }
    else{
        return response()->Json([
            "status" => 422,
            "message" => "Eleve not found"
        ]);
    }
}

protected function update(Request $request , $id){
    $eleve = Eleve::find($id);
    if($eleve){
        $validator = Validator::make($request->all(), [
            "nom" => "required| max:191",
            "prenom" => "required| max:191",
            "filiere" => "required | max:191",
            "niveau"=> "required",
            "code"=> "required"
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $eleve->nom = $request->input("nom");
            $eleve->prenom = $request->input("prenom");
            $eleve->code = $request->input("code");
            $eleve->niveau = $request->input("niveau");
            $eleve->filiere = $request->input("filiere");
            $eleve->sex = $request->input("sex");
            $eleve->phone = $request->input("phone");
            $eleve->about = $request->input("about");
            $eleve->adrresse = $request->input("adrresse");
            $eleve->update();
            return response()->json([
                "status" => 200,
                "message" => "Eleve Updated successfully"
            ]);
        }
    }
}
else{
    return response()->Json([
        "status" => 422,
        "message" => "Eleve not found"
    ]);
}
```

```

}

protected function profile(){}
    $eleve = Eleve::where('user_id',auth()->user()->id )->first();
    if($eleve){
        return response()->json([
            "status" => 200,
            "eleve" => $eleve
        ]);
    }
    else{
        return response()->json([
            "message" => "Eleve Not found",
            "status" => 404
        ]);
    }
}

protected function change(Request $request){
    $eleve = Eleve::where('user_id',auth()->user()->id )->first();
    if($eleve){
        $eleve->phone = $request->phone;
        $eleve->adresse = $request->adresse;
        $eleve->date_nais = $request->date_nais;
        $eleve->about = $request->about;
        $eleve->save();
```

```

        return response()->json([
            "status" => 200,
            "message" => "info changed successfully"
        ]);

    }
    else{
        return response()->json([
            "message" => "Eleve Not found",
            "status" => 404
        ]);
    }
}

```

Fonctions pour gérer les modules:

```
class ModuleController extends Controller
{
    protected function store(Request $request){
        $validator = Validator::make($request->all(), [
            "code" => "required| max:191 | unique:modules,code",
            "designation" => "required| max:191",
            "niveau" => "required| max:191",
            "filiere" => "required | max:191",
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $module = new Module ;
            $module->code = $request->input("code");
            $module->designation = $request->input("designation");
            $module->filiere = $request->input("filiere");
            $module->semestre = $request->input("niveau");
            if($request->input("niveau") === "S1" || $request->input("niveau") === "S2"){
                $module->niveau = 1;
            }
            else if($request->input("niveau") === "S3" || $request->input("niveau") === "S4"){
                $module->niveau = 2;
            }
            else if($request->input("niveau") === "S5" || $request->input("niveau") === "S6"){
                $module->niveau = 3;
            }

            $module->save();
            return response()->json([
                "status" => 200,
                "message" => "Module Added successfully"
            ]);
        }
    }
    protected function index(){
        $module = Module::All();
        return response()->json([
            "status" => 200,
            "module" => $module
        ]);
    }
    protected function edit($id){
        $module = Module::find($id);
        if($module){
            return response()->json([
                "status" => 200,
                "module" => $module
            ]);
        }
    }
}
```

```

    }
    protected function destroy($id){
        $module = Module::find($id);
        if($module){
            $module->delete();
            return response()->json([
                "message" => "Module Deleted successfully",
                "status" => 200
            ]);
        } else{
            return response()->json([
                "status" => 404,
                "message" => "Module Not found"
            ]);
        }
    }
    protected function get($filiere , $niveau){
        $module = Module::where("filiere" , $filiere)->where("niveau" , $niveau)->with('element')->get();
        if($module){
            return response()->json([
                "module" => $module,
                "status" => 200
            ]);
        } else{
            return response()->json([
                "status" => 404,
                "message" => "Module Not found"
            ]);
        }
    }
}
```

```

protected function update(Request $request , $id){
    $module = Module::find($id);
    if($module){
        $validator = Validator::make($request->all(), [
            "designation" => "required| max:191",
            "niveau" => "required| max:191",
            "filiere" => "required | max:191",
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
    } else{
        $module->code = $request->input("code");
        $module->designation = $request->input("designation");
        $module->filiere = $request->input("filiere");
        $module->semestre = $request->input("niveau");
        if($request->input("niveau") === "S1" || $request->input("niveau") === "S2"){
            $module->niveau = 1;
        }
        else if($request->input("niveau") === "S3" || $request->input("niveau") === "S4"){
            $module->niveau = 2;
        }
        else if($request->input("niveau") === "S5" || $request->input("niveau") === "S6"){
            $module->niveau = 3;
        }

        $module->update();
        return response()->json([
            "status" => 200,
            "message" => "Module Updated successfully"
        ]);
    }
} else{
    return response()->json([
        "status" => 404,
        "message" => "Module Not found"
    ]);
}
}

```

Fonctions pour gérer les éléments des modules:

```
class ElementModuleController extends Controller
{
    protected function store(Request $request){
        $validator = Validator::make($request->all(), [
            "code" => "required| max:191",
            "designation" => "required| max:191",
            "vh" => "required| max:191",
            "poids" => "required| max:191",
            "id_module" => "required | max:191",
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $element = new Element_module ;
            $element->code = $request->input("code");
            $element->designation = $request->input("designation");
            $element->id_module = $request->input("id_module");
            $element->vh = $request->input("vh");
            $element->poids = $request->input("poids");

            $element->save();
            return response()->json([
                "status" => 200,
                "message" => "Element Added successfully"
            ]);
        }
    }
    protected function index(){
        $element = Element_module::All();
        return response()->json([
            "status" => 200,
            "element" => $element
        ]);
    }
    protected function edit($id){
        $element = Element_module::find($id);
        if($element){
            return response()->json([
                "status" => 200,
                "element" => $element
            ]);
        }
        else{
            return response()->json([
                "status" => 404,
                "message" => "Element module Not found"
            ]);
        }
    }
}
```

```

protected function update(Request $request , $id){
    $element = Element_module::find($id);
    if($element){
        $validator = Validator::make($request->all(), [
            "code" => "required| max:191",
            "designation" => "required| max:191",
            "vh" => "required| max:191",
            "poids" => "required| max:191",
            "id_module" => "required | max:191",
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $element->code = $request->input("code");
            $element->designation = $request->input("designation");
            $element->id_module = $request->input("id_module");
            $element->vh = $request->input("vh");
            $element->poids = $request->input("poids");

            $element->update();
            return response()->json([
                "status" => 200,
                "message" => "Element Updated successfully"
            ]);
        }
    }
    else{
        return response()->json([
            "status" => 404,
            "message" => "Element Not found"
        ]);
    }
}
protected function destroy($id){
    $element = Element_module::find($id);
    if($element){
        $element->delete();
        return response()->json([
            "message" => "Element Deleted successfully",
            "status" => 200
        ]);
    }
    else{
        return response()->json([
            "status" => 404,
            "message" => "Element Not found"
        ]);
    }
}

```

```
protected function get($module_id){
    $element = Element_module::where("id_module" , $module_id)->get();
    if($element){
        return response()->json([
            "element" => $element,
            "status" => 200
        ]);
    }
    else{
        return response()->json([
            "status" => 404,
            "message" => "Element Not found"
        ]);
    }
}
```

Fonctions pour gérer les notes:

```
class NoteController extends Controller
{
    protected function store(Request $request){
        $validator = Validator::make($request->all(), [
            "element_module_id" => "required| max:191",
            "eleve_id" => "required| max:191",
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $note = new note;
            $note->element_module_id = $request->input("element_module_id");
            $note->eleve_id = $request->input("eleve_id");
            $note->note = $request->input("note");

            $note->save();
            return response()->json([
                "status" => 200,
                "message" => "Note Added successfully"
            ]);
        }
    }

    protected function index(){
        $note = note::All();
        return response()->json([
            "status" => 200,
            "note" => $note
        ]);
    }

    protected function edit($id){
        $note = note::find($id);
        if($note){
            return response()->json([
                "status" => 200,
                "note" => $note
            ]);
        }
        else{
            return response()->json([
                "status" => 404,
                "message" => "Note Not found"
            ]);
        }
    }
}
```

```
}

protected function get($element){
    $note = note::where('element_module_id' , $element )->get();
    if($note){
        return response()->json([
            "note" => $note,
            "status" => 200
        ]);
    }
    else{
        return response()->json([
            "status" => 404,
            "message" => "note Not found"
        ]);
    }
}

protected function getEleve(){
    $note = note::where('eleve_id' , auth()->user()->id )->get();
    if($note){
        return response()->json([
            "note" => $note,
            "status" => 200
        ]);
    }
    else{
        return response()->json([
            "status" => 404,
            "message" => "note Not found"
        ]);
    }
}
```

```

}

protected function update(Request $request , $id){
    $note = note::find($id);
    if($note){
        $validator = Validator::make($request->all(), [
            "element_module_id" => "required| max:191",
            "eleve_id" => "required| max:191",
            "note" => "required"
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $note->element_module_id = $request->input("element_module_id");
            $note->eleve_id = $request->input("eleve_id");
            $note->note = $request->input("note");

            $note->update();
            return response()->json([
                "status" => 200,
                "message" => "Note Updated successfully"
            ]);
        }
    }
    else{
        return response()->json([
            "status" => 404,
            "message" => "note Not found"
        ]);
    }
}

protected function destroy($id){
    $note = note::find($id);
    if($note){
        $note->delete();
        return response()->json([
            "message" => "note Deleted successfully",
            "status" => 200
        ]);
    }
    else{
        return response()->json([
            "status" => 404,
            "message" => "note Not found"
        ]);
    }
}

```

Fonctions pour gérer les moyennes:

```
class MoyenneController extends Controller
{
    protected function store(Request $request){
        $validator = Validator::make($request->all(), [
            "filiere" => "required| max:191",
            "eleve_id" => "required| max:191",
            "moyenne" => "required",
            "niveau" => "required"
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $moyenne = new Moyenne;
            $moyenne->filiere = $request->input("filiere");
            $moyenne->eleve_id = $request->input("eleve_id");
            $moyenne->moyenne = $request->input("moyenne");
            $moyenne->niveau = $request->input("niveau");

            $moyenne->save();
            return response()->json([
                "status" => 200,
                "message" => "Moyenne Added successfully"
            ]);
        }
    }
    protected function index(){
        $moyenne = Moyenne::All();
        return response()->json([
            "status" => 200,
            "moyenne" => $moyenne
        ]);
    }
    protected function edit($id){
        $moyenne = Moyenne::find($id);
        if($moyenne){
            return response()->json([
                "status" => 200,
                "moyenne" => $moyenne
            ]);
        }
        else{
            return response()->json([
                "status" => 404,
                "message" => "Moyenne Not found"
            ]);
        }
    }
    protected function update(Request $request, $id){
```

```

    protected function update(Request $request , $id){
        $moyenne = Moyenne::find($id);
        if($moyenne){
            $validator = Validator::make($request->all(), [
                "filiere" => "required| max:191",
                "eleve_id" => "required| max:191",
                "moyenne" => "required",
                "niveau" => "required"
            ]);
            if($validator->fails()){
                return response()->json([
                    "status" => 400,
                    "errors" => $validator->messages(),
                ]);
            }
            else{
                $moyenne->filiere = $request->input("filiere");
                $moyenne->eleve_id = $request->input("eleve_id");
                $moyenne->moyenne = $request->input("moyenne");
                $moyenne->niveau = $request->input("niveau");

                $moyenne->update();
                return response()->json([
                    "status" => 200,
                    "message" => "Moyenne updated successfully"
                ]);
            }
        }
        else{
            return response()->json([
                "status" => 404,
                "message" => "Moyenne Not found"
            ]);
        }
    }

    protected function destroy($id){
        $moyenne = Moyenne::find($id);
        if($moyenne){
            $moyenne->delete();
            return response()->json([
                "message" => "Moyenne Deleted successfully",
                "status" => 200
            ]);
        }
        else{
            return response()->json([
                "status" => 404,
                "message" => "Moyenne Not found"
            ]);
        }
    }
}

```

Fonctions pour gérer les responsables des filières :

```
class Responsable_filiere_controller extends Controller
{
    protected function store(Request $request){
        $validator = Validator::make($request->all(), [
            "nom" => "required| max:191",
            "prenom" => "required| max:191",
            "departement" => "required | max:191",
            "filiere"=>"required | max:191 | unique:responsable_filiere,filiere",
            "user_id" => "required",
            'image' => 'required| image| mimes:jpeg,png,jpg',
            "sex" => "required",
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $respo = new Responsable_filiere ;
            $respo->nom = $request->input("nom");
            $respo->sex = $request->input("sex");
            $respo->prenom = $request->input("prenom");
            $respo->user_id = $request->input("user_id");
            $respo->filiere = $request->input("filiere");
            $respo->departement = $request->input("departement");
            if($request->hasFile("image")){
                $file = $request->file("image");
                $extention = $file->getClientOriginalExtension();
                $filename = time() . '.' . $extention;
                $file->move("uploads/respo/" , $filename);
                $respo->image = 'uploads/respo/' . $filename;
            }
            $respo->save();
            return response()->json([
                "status" => 200,
                "message" => "Responsable_filiere Added successfully"
            ]);
        }
    }

    protected function edit($id){
        $respo = Responsable_filiere::find($id);
        if($respo){
            return response()->Json([
                "status" => 200,
                "respo" => $respo,
            ]);
        }
        else{
            return response()->Json([
                "status" => 422,
                "message" => "Responsable_filiere not found"
            ]);
        }
    }
}
```

```

protected function update(Request $request , $id){
    $respo = Responsable_filiere::find($id);
    if($respo){
        $validator = Validator::make($request->all(), [
            "nom" => "required| max:191",
            "prenom" => "required| max:191",
            "departement" => "required | max:191",
        ]);
        if($validator->fails()){
            return response()->json([
                "status" => 400,
                "errors" => $validator->messages(),
            ]);
        }
        else{
            $respo->nom = $request->input("nom");
            $respo->filiere = $request->input("filiere");
            $respo->prenom = $request->input("prenom");
            $respo->departement = $request->input("departement");
            $respo->update();
            return response()->json([
                "status" => 200,
                "message" => "Responsable_filiere Updated successfully"
            ]);
        }
    }
    else{
        return response()->json([
            "message" => "Responsable_filiere Not found",
            "status" => 404
        ]);
    }
}

protected function destroy($id){
    $respo = Responsable_filiere::find($id);
    if($respo){
        $respo->delete();
        return response()->json([
            "message" => "Responsable_filiere Deleted successfully",
            "status" => 200
        ]);
    }
    else{
        return response()->json([
            "message" => "Responsable_filiere Not found",
            "status" => 404
        ]);
    }
}

```

```

protected function index(){
    $respo = Responsable_filiere::all();
    return response()->json([
        "status" => 200,
        "respo" => $respo
    ]);
}

protected function profile(){
    $respo = Responsable_filiere::where('user_id',auth()->user()->id)->first();
    if($respo){
        return response()->json([
            "status" => 200,
            "Respo" => $respo
        ]);
    }
    else{
        return response()->json([
            "message" => "Respo Not found",
            "status" => 404
        ]);
    }
}

protected function change(Request $request){
    $respo = Responsable_filiere::where('user_id',auth()->user()->id)->first();
    if($respo){
        $respo->phone = $request->phone;
        $respo->adresse = $request->adresse;
        $respo->date_nais = $request->date_nais;
        $respo->about = $request->about;
        $respo->save();

        return response()->json([
            "status" => 200,
            "message" => "info changed successfully"
        ]);
    }
    else{
        return response()->json([
            "message" => "Respo Not found",
            "status" => 404
        ]);
    }
}

```

2-Les déclencheurs et observers utilisés:

Création des tables grace Aux migrations:

On commence par créer les tables nécessaires de notre base de données afin de manipuler toutes les informations nécessaires pour notre application. Laravel Nous propose un outil simple qui permet de créer les tables très facilement: Les migrations

Création de la table users(utilisateurs):

On note que Nous avons ajouté quelque attributs pour enrichir la table:

L'attribut **Verified** pour vérifier l'activation du compte.

L'attribut **Role**, pour déterminer le role de l'utilisateur(Admin, Chèf de filière, étudiant)

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->integer('role')->default(0)->nullable();
            $table->integer('verified')->default(0)->nullable();
            $table->string('login')->unique();
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}

```

Et pour les tables Etudiant et chef de filière, nous avons ajouté quelques informations personnelles: L'image, le sexe, date de naissance, numéro de téléphone et about(un paragraphe descriptif de la personne).

La table responsable_filière:

```
use Illuminate\Support\Facades\Schema;

class CreateResponsableFiliereTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('responsable_filiere', function (Blueprint $table)
        {
            $table->id();
            $table->string('nom');
            $table->string('prenom');
            $table->string('filiere');
            $table->string('departement');
            $table->integer("user_id");
            $table->string("sex");
            $table->string("phone")->nullable();
            $table->string("adresse")->nullable();
            $table->longText("about")->nullable();
            $table->string("date_nais")->nullable();
            $table->string("image")->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('responsable_filiere');
    }
}
```

La table Eleve :

```
public function up()
{
    Schema::create('eleves', function (Blueprint $table) [
        $table->id();
        $table->string("code");
        $table->string("nom");
        $table->string("prenom");
        $table->string("sex");
        $table->string("phone")->nullable();
        $table->string("adresse")->nullable();
        $table->longText("about")->nullable();
        $table->string("date_nais")->nullable();
        $table->string("niveau");
        $table->string("filiere");
        $table->string("user_id");
        $table->string("image");
        $table->timestamps();
    ]);
}
```

La table Filière:

```
    /**
     * @return void
     */
    public function up()
    {
        Schema::create('filieres', function (Blueprint $table) {
            $table->id();
            $table->string("code");
            $table->string("designation");
            $table->integer("id_responsable");
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('filieres');
    }
}
```

La table modules:

```
public function up()
{
    Schema::create('modules', function (Blueprint $table) {
        $table->id();
        $table->string("code");
        $table->string("designation");
        $table->string("semestre");
        $table->string("niveau");
        $table->string("filiere");
        $table->timestamps();
    });
}
```

La table element_modules :

```
public function up()
{
    Schema::create('element_modules', function (Blueprint $table) {
        $table->id();
        $table->string("code");
        $table->string("designation");
        $table->string("vh");
        $table->string("poids");
        $table->string("id_module");
        $table->timestamps();
    });
}
```

La table moyennes :

```
public function up()
{
    Schema::create('moyennes', function (Blueprint $table) {
        $table->id();
        $table->string('eleve_id');
        $table->string("filiere");
        $table->string("niveau");
        $table->decimal("moyenne");
        $table->timestamps();
    });
}
```

La table notes :

```
public function up()
{
    Schema::create('notes', function (Blueprint $table) {
        $table->id();
        $table->string("eleve_id");
        $table->string("element_module_id");
        $table->decimal('note')->default(0);
        $table->timestamps();
    });
}
```

Les triggers:

(Les fonctionnalités du triggers sont décrites dans les commentaires):

Pour une meilleur visibilité, Nous avons décidé de présenter le script directement ici, au lieu d'utiliser une capture d'écran:

```
create or replace trigger gestionMoyenne
after insert or update or delete on note
for each row
declare
poids_total int;
sum_note float;
fil varchar(4);
niv int;
moy float;
newMoy float;
test_existantse int ;
begin
select filiere , niveau into fil,niv  from eleve where id_eleve=:NEW.elv;

-- 1-) cas d'insertion
if inserting then
--- on test si la somme du poids des tout les elements isnere dans la table note est egale a 12
(nbr du module par annee)
select sum(poids) into poids_total from element_mod where id_element in (select id_elm from
note where elv=:NEW.elv );
if poids_total=12 then
select sum(note) into sum_note from note where :NEW.elv=elv;
moy:=sum_note/12;
```

-- Insertion de la moyenne

insert into moyenne values(:NEW.elv , :NEW.elv , fil , niv , moy); --- puisque id_moyenne ne peut pas etre null on l'affecte id_elv comme valeur ce qui va chnage rien

--- Incrementation du niveau

--- le cas si un etudiant etude en 2eme annee cycle prepa

if fil='AP' or niv=2 then

-- on lui affecte aleatoirement a une des filiere a la 1ere annee cycle ing

niv := 1;

case round(dbms_random.value(1,5))

WHEN 1 THEN fil:='GINF' ;

WHEN 2 THEN fil:='GIL' ;

WHEN 3 THEN fil:='GSEA' ;

WHEN 4 THEN fil:='G3EI' ;

WHEN 5 THEN fil:='GSTR' ;

end case;

elsif niv!=3 then niv:=niv+1;

end if ;

end if;

end if;

-- 2-) cas du suppression

if deleting then

--- si on suprime une note on suprime la moyenne si elle est deja inserer

select case

when exists(select * from moyenne where id_elv=:OLD.elv

)

then 1

else 0

end into test_existantse

from dual;

if test_existantse=1 then

select moyenne into moy from moyenne where id_elv=:OLD.elv ;

```

-- on decremente le niveau si la moyenne > 10 pour cycle prepa et si la moyenne > 12 pour
cycle ing

-- Si l'eleve etudie en 1er anne cycle ing on lui ramene au 2eme annee cycle prepa

if fil!='AP' and niv=1 and moy>=12 then

fil:='AP';

niv:=2;

end if;

elsif fil='AP' and niv=2 and moy>=10 then niv:=niv-1;

elsif fil!= 'AP' and moy >=12 then niv:=niv-1;

    delete from moyenne where id_elv=:OLD.elv;

end if;

end if;

-- 3-) cas du mdoification

-- on test si la moyenne est deja inseree

if updating then

select case

when exists(select * from moyenne where id_elv=:OLD.elv

)

then 1

else 0

end into test_existantse

from dual;

if test_existantse=1 then

select moyenne into moy from moyenne where id_elv=:OLD.elv ;

newMoy := moy + (:NEW.note - :OLD.note)/12 ;

update moyenne set moyenne = newMoy where id_elv=:NEW.elv;

-- moy : l'ancien moyenne , neuMoy : la nouvelle moyenne

-- on test si l'annclien < 12 (10) et la nouvelle > 12 (10)

if (moy < 12 and newMoy > 12 and fil!='AP' and niv!=3) or (moy < 10 and newMoy > 10 and
fil='AP' and niv=1) then

niv:=niv+1;

```

```

elsif moy<10 and newMoy > 10 and fil='AP' and niv=2 then
niv:=1;
case round(dbms_random.value(1,5))
WHEN 1 THEN fil:='GINF' ;
WHEN 2 THEN fil:='GIL' ;
WHEN 3 THEN fil:='GSEA' ;
WHEN 4 THEN fil:='G3EI' ;
WHEN 5 THEN fil:='GSTR' ;
end case;
end if;

-- on test si l'annien > 12 (10) et la novelle < 12 (10)

if (moy > 12 and newMoy <12 and fil!='AP' and niv!= 1) or (moy > 10 and newMoy < 10 and fil='AP' and niv=2) then
niv:=niv-1;
elsif moy> 12 and newMoy < 12 and fil!='AP' and niv=1 then
niv:=2;
fil:='AP';
end if;
end if;
end if;

end;

```

Utilisation des événements (Observers) dans Laravel:

Afin d'entamer une architecture bien structurée au niveau de gestion de base de données et des déclencheurs, Nous avons décidé d'utiliser outil très puissant présenté par le Framework Laravel: Les événements.

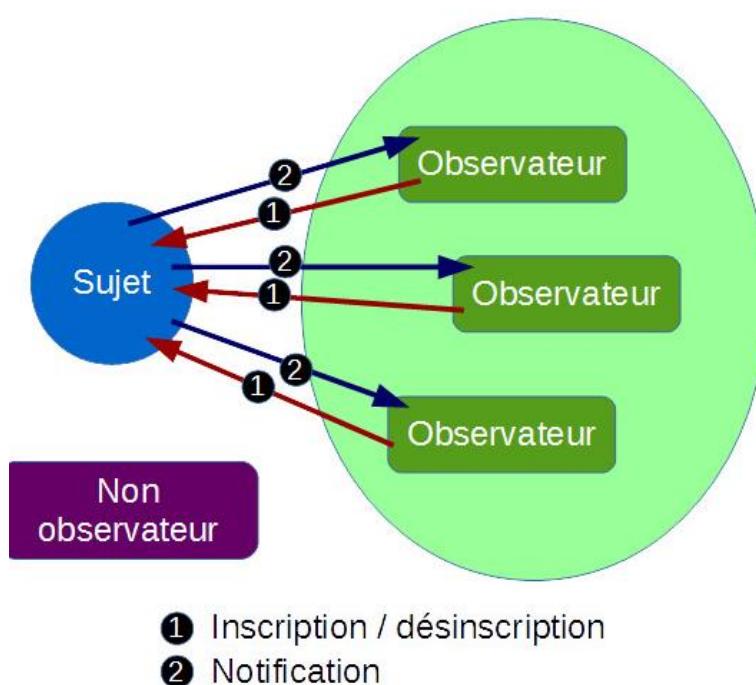
En effet, il y a plusieurs façons de programmer. En général la plus simple et lisible consiste à procéder de façon séquentielle : on a une suite d'instructions (modulables par des conditions) qui sont toujours les mêmes. Le flux est facile à suivre.

Mais on a aussi l'approche événementielle. On va réagir à des événements susceptibles de se produire. On peut mettre ainsi en place un ensemble constitué par une partie qui détecte des événements et une autre qui les gère.

Le design pattern Observateur (**observer**) établit une relation de type **1:n** entre des objets. Si l'objet côté **1** change d'état il en informe les objets côté **n** pour qu'ils puissent agir en conséquence. Ça implique une intendance et les objets doivent pouvoir s'inscrire et se mettre à l'écoute des événements du sujet.

Voici une visualisation de ce design pattern :

On a un sujet et des observateurs. Les observateurs doivent s'inscrire



(on dit aussi s'abonner) auprès du sujet. Lorsque le sujet change d'état il notifie tous ses abonnés. Un observateur peut aussi se désinscrire, il ne recevra alors plus les notifications.

Pour notre application, Nous avons profité de trois fonctions prédéfinies dans le Framework Laravel: Created(ce qui est équivalent à after insert), Updated(ce qui est équivalent à after insert) , Deleted(ce qui est équivalent à after insert).

Les illustrations suivantes montrent le code utilisé:

Fonction Created:

```
class noteObserver
{
    public function created(note $note)
    {
        $poidsTotal = 0 ;
        $noteTotal = 0;
        $notes = note::where('eleve_id' , $note->eleve_id)->get();
        $filiereAP = Filiere::where('designation' , "AP")->first();
        foreach ($notes as $n){
            $poidsTotal = $n->Element_module['poids'] + $poidsTotal;
        }
        if($poidsTotal == 2){
            foreach ($notes as $n){
                $noteTotal = $n->Element_module['poids'] * $n->note + $noteTotal;
            }
            $moyenne = new Moyenne;
            $moyenne->filiere_id = $note->Eleve['filiere_id'];
            $moyenne->eleve_id = $note->Eleve['id'];
            $moyenne->niveau = $note->Eleve['niveau'];
            $moyenne->moyenne = $noteTotal/2 ;
            $eleve = Eleve::where('id' , $note->eleve_id)->first();
            if($eleve && $moyenne->moyenne > 12){
                if($eleve->filiere_id == $filiereAP->id && $eleve->niveau == '1') $eleve->niveau = '2';
                elseif ($eleve->filiere_id == $filiereAP->id && $eleve->niveau == '2') {
                    /*ici on va definir une fonction qui affecter d'une maniere aleatoire une filiere pour chaque etudiant
                    qui a bien passee la 2eme annee du cyle prepa ; bientot on va redefinir cette fonction plus tard ;*/
                    $Filières = [1, 2 , 3, 4, 5];
                    $randomFiliere = Arr::random($Filières);
                    $eleve->filiere_id = $randomFiliere ;
                    $eleve->niveau = '1';
                }
                elseif ($eleve->filiere_id != $filiereAP->id && $eleve->niveau < 3 ) {
                    $eleve->niveau++;
                }
            }
            $eleve->save();
            $moyenne->save();
        }
    }
}
```

Fonction Updated:

```
}

public function updated(note $note)
{
    $moyenne = Moyenne::where('eleve_id', $note->eleve_id)->first();
    $filiereAP = Filiere::where('designation', 'AP')->first();
    if($moyenne){
        $noteTotal = 0;
        $notes = note::where('eleve_id', $note->eleve_id)->get();
        $filiereAP = Filiere::where('designation', 'AP')->first();

        foreach ($notes as $n){
            $noteTotal += $n->element_module['poids'] * $n->note + $noteTotal;
        }
        $moyenne->filiere_id = $note->Eleve['filiere_id'];
        $moyenne->eleve_id = $note->Eleve['id'];
        $moyenne->niveau = $note->Eleve['niveau'];
        $oldMoyenne = $moyenne->moyenne;

        //on va garder l'ancien valeur de la moyenne pour teste si les possintites entre l'antien valeur et la nouvelle
        $moyenne->moyenne = $noteTotal/2;
        $moyenne->update();
        $eleve = Eleve::where('id', $note->eleve_id)->first();

        //si l'eleve a réussi apres la modification
        if( $eleve && $moyenne->moyenne >= 12 && $oldMoyenne < 12){
            if($eleve->filiere_id == $filiereAP->id && $eleve->niveau == '1') $eleve->niveau = '2';
            elseif ($eleve->filiere_id == $filiereAP->id && $eleve->niveau == '2') {
                /*ici on va definir une fonction qui affecter d'une maniere aleatoire une filiere pour chaque etudiant
                qui a bien passe la 2eme annee du cycle prepa ; bésure on va redéfinir cette fonction plus tard ;*/
                $filieres = [1, 2, 3, 4, 5];
                $randomFiliere = Arr::random($filieres);
                $eleve->filiere_id = $randomFiliere;
                $eleve->niveau = '1';
            }
            elseif ($eleve->filiere_id != $filiereAP->id || $eleve->niveau < 3 ) {
                $eleve->niveau++;
            }
            $eleve->save();
        }

        //si l'eleve va redoubler apres la modification
        if( $eleve && $moyenne->moyenne < 12 && $oldMoyenne >= 12){
            if($eleve->niveau > 1) $eleve->niveau--;
            else{
                $eleve->niveau = '2';
                $eleve->filiere_id = $filiereAP->id;
            }
            $eleve->save();
        }
    }
}
```

Fonction Deleted:

```

public function deleted(note $note)
{
    $moyenne = Moyenne::where('eleve_id' , $note->eleve_id)->first();
    $filiereAP = Filiere::where('designation' , 'AP')->first();

    //si une personne a passé à l'année suivante alors il doit revenir à l'année d'origine;
    if($moyenne){
        $eleve = Eleve::where('id' , $note->eleve_id)->first();
        if( $eleve && $moyenne->moyenne >= 12){
            if($eleve->niveau > 1) $eleve->niveau--;
            else{

                $eleve->niveau = '2';
                $eleve->filiere_id = $filiereAP->id ;
            }
            $eleve->save();
        }
        $moyenne->delete();
    }
}

```

3-Utilisation de Laravel Rest Api:

Laravel permet de bâtir facilement des API à partir de ressource Eloquent.

REST (Representational State Transfer) est un style d'architecture pour constituer des API. Pour Laravel une API REST est :

- organisée autour de ressources
- sans persistance (aucune session)
- orientée client-serveur
- pouvant être mis en cache (donc une requête doit retourner toujours les mêmes résultats)
- retourner du JSON...

Les illustrations suivantes montrent les apis utilisées pour notre application:

```

<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\API\AuthController;
use App\Http\Controllers\API\Responsable_filiere_controller;
use App\Http\Controllers\API\ModuleController;
use App\Http\Controllers\API\ElementModuleController;
use App\Http\Controllers\API\EleveController;
use App\Http\Controllers\API>NoteController;
use App\Http\Controllers\API\MoyenneController;

Route::post('login' , [AuthController::class , 'login' ]);

Route::middleware(['auth:sanctum'])->group(function (){
    //Eleve =====
    Route::get('CheckEleveAuth' , function (){
        return response()->Json(["message" => "You are in" , "status" => 200] , 200);
    });
    Route::get('profileeleve' , [EleveController::class , 'profile']);
    Route::put('change_info_eleve' , [EleveController::class , 'change']);
    Route::get('getNotesEleve' , [NoteController::class , 'getEleve']);
    Route::post('logout' , [AuthController::class , 'logout']);
    Route::put('change_pass' , [AuthController::class , 'change']);
    Route::get('index_element' , [ElementModuleController::class , 'index']);
    Route::get('getElements/{filiere}/{niveau}' , [ElementModuleController::class , 'getElements']);
    Route::get('getModules/{filiere}/{niveau}' , [ModuleController::class , 'get']);
    Route::get('index_module' , [ModuleController::class , 'index']);

    //responsable de filiere=====
    Route::middleware('isApiRespo')->group(function (){
        Route::get('CheckRespoAuth' , function (){
            return response()->Json(["message" => "You are in" , "status" => 200] , 200);
        });
        Route::get('index' , [AuthController::class , 'index']);
        Route::get('getLastId' , [AuthController::class , 'getLastId']);
        Route::get('profilerespo' , [Responsable_filiere_controller::class , 'profile']);
        Route::put('change_info_respo' , [Responsable_filiere_controller::class , 'change']);
        Route::get('getElements/{module_id}' , [ElementModuleController::class , 'get']);

        //CRUD Note
        Route::post('store_note' , [NoteController::class , 'store']);
        Route::post('store_notes' , [NoteController::class , 'storemultiply']);
        Route::delete('destroy_note/{id}' , [NoteController::class , 'destroy']);
        Route::get('edit_note/{id}' , [NoteController::class , 'edit']);
        Route::put('update_note/{id}' , [NoteController::class , 'update']);
        Route::get('index_note' , [NoteController::class , 'index']);
        Route::get('getNote/{element}' , [NoteController::class , 'get']);

        Route::delete('destroy/{id}' , [AuthController::class , 'destroy']);
        Route::get('edit_element/{id}' , [ElementModuleController::class , 'edit']);

        Route::get('eleves/{filiere}/{niveau}' , [EleveController::class , 'get']);
    });
});

```

```

//Admin=====
Route::middleware('isApiAdmin')->group(function (){
    Route::get('CheckAdminAuth' , function (){
        return response()->Json(["message" => "You are in" , "status" => 200] , 200);
    });
    Route::post('register' , [AdminController::class , 'register']);

    //CRUD Responsable filiere
    Route::post('store_respo' , [Responsable_filiere_controller::class , 'store']);
    Route::delete('destroy_respo/{id}' , [Responsable_filiere_controller::class , 'destroy']);
    Route::get('edit_respo/{id}' , [Responsable_filiere_controller::class , 'edit']);
    Route::put('update_respo/{id}' , [Responsable_filiere_controller::class , 'update']);
    Route::get('index_respo' , [Responsable_filiere_controller::class , 'index']);

    //CRUD Module
    Route::post('store_module' , [ModuleController::class , 'store']);
    Route::delete('destroy_module/{id}' , [ModuleController::class , 'destroy']);
    Route::get('edit_module/{id}' , [ModuleController::class , 'edit']);
    Route::put('update_module/{id}' , [ModuleController::class , 'update']);

    //CRUD Element Module
    Route::post('store_element' , [ElementModuleController::class , 'store']);
    Route::delete('destroy_element/{id}' , [ElementModuleController::class , 'destroy']);
    Route::put('update_element/{id}' , [ElementModuleController::class , 'update']);

    //CRUD Eleve
    Route::post('store_eleve' , [EleveController::class , 'store']);
    Route::delete('destroy_eleve/{id}' , [EleveController::class , 'destroy']);
    Route::get('edit_eleve/{id}' , [EleveController::class , 'edit']);
    Route::put('update_eleve/{id}' , [EleveController::class , 'update']);
    Route::get('index_eleve' , [EleveController::class , 'index']);
    Route::get('search/{search}' , [EleveController::class , 'search']);

    //CRUD Moyenne
    Route::post('store_moyenne' , [MoyenneController::class , 'store']);
    Route::delete('destroy_moyenne/{id}' , [MoyenneController::class , 'destroy']);
    Route::get('edit_moyenne/{id}' , [MoyenneController::class , 'edit']);
    Route::put('update_moyenne/{id}' , [MoyenneController::class , 'update']);
    Route::get('index_moyenne' , [MoyenneController::class , 'index']);
})
});

```

4-Utilisation des Midelwares:

Laravel Nous offre un autre outil très puissant, cette fois-ci, pour assurer que chaque utilisateur va avoir accès vers les Apis qui lui correspondent en se basant sur son rôle(Elève, Chef de Filière, Admin). Cet outil s'appelle: Les Midelwares.

Les midelwares sont des fonctions qui permettent d'enrichir ou de vérifier une requête (Api) Avant qu'elle arrive au contrôleur.

```

class ApiAdminMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(Auth::check()){
            if(auth()->user()->tokenCan('server:admin')){
                return $next($request);
            }
            else{
                return response()->Json([
                    'status' => 403,
                    'message'=> 'Access denied as you are not an admin'
                ]);
            }
        }
        else{
            return response()->Json([
                'status' => 401,
                'message'=> 'Please Log in First'
            ]);
        }
    }
}

```

```

public function handle(Request $request, Closure $next)
{
    if(Auth::check()){
        if(auth()->user()->tokenCan('server:respo') || auth()->user()->tokenCan('server:admin')){
            return $next($request);
        }
        else{
            return response()->Json([
                'status' => 403,
                'message'=> 'Access denied as you are not a responsable'
            ]);
        }
    }
    else{
        return response()->Json([
            'status' => 401,
            'message'=> 'Please Log in First'
        ]);
    }
}

```

5-Les modèles (Relationships) sous Laravel:

Dans le cadre du travail sous une architecture MVC assurée par Laravel, Nous avons utilisé les modèles au sein desquels nous avons déployé les relations. Ces derniers représentent un autre avantage qui

vient avec le Framework Laravel et permettent de gérer les relations et joitures des différentes tables de notre application:

Dans les illustrations qui suivent, Nous présentons les différents modèles utilisés:

Modèle pour Responsable Filière:

```
9  class Responsable_filiere extends Model
10 {
11     use HasFactory;
12     protected $fillable = [
13         'nom',
14         'prenom',
15         'departement',
16         'user_id',
17         'image'
18     ];
19     protected $with = ['user'];
20
21     public function user(){
22         return $this->belongsTo(User::class);
23     }
24 }
```

Modèle pour Moyenne:

```
1  use App\Models\Eleve;
2  class Moyenne extends Model
3  {
4      use HasFactory;
5      protected $fillable = [
6          'eleve_id',
7          'filiere_id',
8          'moyenne',
9          'niveau'
10     ];
11     protected $with = ['eleve' , 'filiere'];
12     public function eleve(){
13         return $this->belongsTo(Eleve::class);
14     }
15 }
```

Modèle pour Note :

```
1  use App\Widgets\Element_Module;
2  class note extends Model
3  {
4      use HasFactory;
5      protected $fillable = [
6          'note',
7          'eleve_id',
8          'element_module_id'
9      ];
10     protected $with = ['element_module' , 'eleve'];
11     public function eleve(){
12         return $this->belongsTo(Eleve::class );
13     }
14     public function element_module(){
15         return $this->belongsTo(Element_Module::class);
16     }
17 }
```

Modèle pour Eleve:

```
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use \App\Models\User;
8  use \App\Models\Moyenne;
9
10 class Eleve extends Model
11 {
12     use HasFactory;
13     protected $fillable = [
14         'nom',
15         'prenom',
16         'niveau',
17         'filiere',
18         'user_id',
19         'image',
20         'code',
21         'sex',
22         'date_nais',
23         'adresse'
24     ];
25     protected $with = ['user'];
26     public function user(){
27         return $this->belongsTo(User::class);
28     }
29
30     public function moyenne(){
31         return $this->hasOne(Moyenne::class);
32     }
33
34 }
```

Modèle pour Utilisateur:

```
3  class User extends Authenticatable
4  {
5      use HasApiTokens, HasFactory, Notifiable;
6
7      /**
8      * The attributes that are mass assignable.
9      *
10     * @var array<int, string>
11     */
12    protected $fillable = [
13        'email',
14        'password',
15        'login',
16        'verified'
17    ];
18
19    /**
20     * The attributes that should be hidden for serialization.
21     *
22     * @var array<int, string>
23     */
24    protected $hidden = [
25        'password',
26        'remember_token',
27    ];
28
29    /**
30     * The attributes that should be cast.
31     *
32     * @var array<string, string>
33     */
34    protected $casts = [
35        'email_verified_at' => 'datetime',
36    ];
37
38    public function eleve(){
39        return $this->hasOne(Eleve::class );
40    }
41    public function responsable_filiere(){
42        return $this->hasOne(Responsable_filiere::class );
43    }
44 }
```

Modèle pour Module:

```
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use \App\Models\Element_module;
8 class Module extends Model
9 {
10     use HasFactory;
11     protected $fillable = [
12         "code",
13         "designation",
14         "niveau",
15         "semestre",
16         "id_filiere"
17     ];
18     protected $casts = ['element'];
19     public function Element(){
20         return $this->hasMany(Element_module::class , 'id_module' , 'id');
21     }
22 }
23
```

6-Présentation générale du site:

Grace à l'utilisation de tous les outils présentés dans les chapitres précédents. Nous avons pu concevoir un site interactif, responsive et dynamique avec différents espaces qui répondent aux besoins qui offre également une version mobile sur téléphone.

Pour le reste du rapport Nous allons nous consacrer pour la présentation de la version sur ordinateur sachant que la version mobile contient exactement les mêmes fonctionnalités.

Page d'accueil:
Le master Head:



Espace de contact:

A screenshot of the contact form page. The top navigation bar is orange with the text "EnsatApp" on the left and "About", "Contact", "Notes", and "mohamed" with a dropdown arrow on the right. The main title "Pour nous contacter" is centered above a form. The form consists of four input fields: "Full name", "Email address", "Phone number", and "Message", each with a placeholder text. Below the message field is a large text area. At the bottom of the form is an orange "SUBMIT" button. At the very bottom of the page, there is a small line of text "+1 (555) 123-4567".

Une explication des différentes fonctionnalités dans le site:



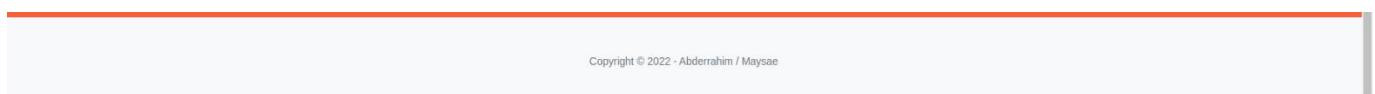
Barre de navigation et Footer:

Ce sont deux éléments présents sur toutes les pages du site, sauf pour l'espace Admin pour lequel nous avons donné une présentation complètement différente:

Barre de navigation contient un menu qui permet de naviguer le site:



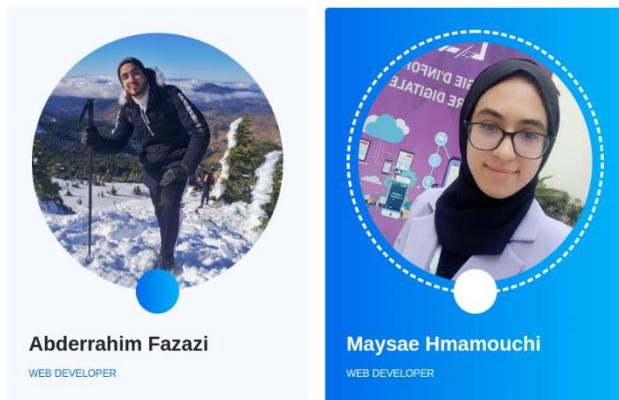
Un footer qui contient les Copyrights



Page About us :

The screenshot shows a top navigation bar with the text "EnsatApp" on the left and "About", "Contact", "Notes", and "abderrahim.fazazi" with a dropdown arrow on the right.

Notre équipe :



Encadré par :

The screenshot shows a top navigation bar with the text "EnsatApp" on the left and "About", "Contact", "Notes", and "abderrahim.fazazi" with a dropdown arrow on the right.

Encadré par :



Abderrahim fazazi

Abderrahim Fazazi élève ingénieur en première année cycle ingénieur à l'ENSA de Tanger, passionné par l'entrepreneuriat, le domaine IT et précisément le développement Web, free-lancer depuis une année, fondateur et président du club Ensappress. « Durant la réalisation de ce projet, j'ai développé plusieurs compétences telles que l'esprit d'équipe, la prise de l'initiative, la gestion du temps et du travail et la communication et la priorisation. »

Maysae Hmamouchi, élève ingénierie en première année cycle Ingénieur, génie informatique à l'ENSA de Tanger, intéressée par les nouvelles technologies, l'intelligence artificielle, secrétaire générale-adjointe et coordinatrice du comité exécutif de l'Association Marocaine pour un Environnement Durable. « Durant la réalisation de ce projet, j'ai pu apprendre des techniques de gestion des projets informatiques, j'ai développé mon esprit de créativité et le sens de partage d'informations avec mes camarades. »

Maysae Hmamouchi,

Notre Vision :

Nous visons à offrir une application web attractive , capable de permettre au cadre administratif de l'ENSAT,d'user des moyens de gestion, des données des étudiants : profil, notes, validation... Nous visons également à la création d'un espace étudiant, à travers lequel, nous allons permettre aux élèves ingénieurs, la consultation de leurs notes, le contact des professeurs par email... Durant toutes les phases de la réalisation du projet, nous avons comme mission le déploiement des différents outils et langages de programmation que nous avons pu appréhender le long de cette année académique ou bien à travers nos auto-formations et également les compétences de gestion de travail en équipe, communication et collaboration.

About US



EnsApp

5 raisons pour utiliser EnsApp...



Qui sommes nous ?

Maysae Hmamouchi / Abderrahim fazazi élèves ingénieurs en première année cycle ingénieur, génie informatique à l'ENSA de Tanger...



Notre vision

Nous visons à offrir une application web attractive et qui est capable...

EnsApp

Notre Slogan ici

Pourquoi cette plateforme ?

1. Un espace pour le cadre administratif de l'ENSAT afin de gérer les informations des étudiants.
2. Un espace pour les chefs de filières pour saisir les notes des étudiants et les délibérations.
3. Un espace pour les étudiants afin de consulter leur profils scolaires et leurs notes.
4. Un espace pour permettre aux étudiants de contacter leurs chefs de filières.

Page de Log in:

Demande à l'utilisateur de saisir son Log in et son mot de passe qui lui sont envoyés par mail institutionnel vu des contraintes de sécurité.

Connectez-vous à votre compte



The page features a collage of four images: a curved wooden bookshelf filled with books, a multi-level study area with tables and chairs under a glass roof, a white building with blue-framed windows, and a view of a modern building complex with a swimming pool in the foreground.

SUIVEZ-NOUS SUR

Toujours dans le cadre d'assurance de sécurité de notre site, nous demandons aux utilisateurs d'activer leur comptes en changeant le mot de passe en un qui est personnel, bien sur en leur demandant de saisir le mot de passe qui leur a été communiquer par email institutionnel.

Veuillez choisir une mot de passe forte



The page includes three input fields for password entry: "Current Password..", "New Password..", and "Confirmed New Password..". Below these is a red "Submit" button.

SUIVEZ-NOUS SUR

L'utilisateur est par la suite redirigé vers l'espace qui lui est dédié selon son type de login, les différents espaces disponibles sont présentés dans les chapitres suivants.

7-Échange des e-mails entre les étudiants et leurs professeurs:

Configuration d'un serveur SMTP:

Au début, nous avons décidé de travailler avec SMTP gmail mais malheureusement Google a arrêté cette option pour des raisons de sécurité.

Less secure apps & your Google Account

To help keep your account secure, from **May 30, 2022**, Google no longer supports the use of third-party apps or devices which ask you to sign in to your Google Account using only your username and password.

Important: This deadline does not apply to Google Workspace or Google Cloud Identity customers. The enforcement date for these customers will be announced on the Workspace blog at a later date.

For more information, continue to read.

C'est pour cela on a décidé d'utiliser Sendinblue smtp :

Création d'un compte sendinblue :

The screenshot shows the Sendinblue dashboard with the following sections:

- Statistics:** Displays monthly calendar grids for May and June 2022, showing the number of days. Below are three performance metrics:
 - Open Rate:** 0
 - Click Rate:** 0
 - Unsubscription Rate:** 0
- Email Campaigns:** A table with columns: ID, NAME, RECIPIENTS, OPENED, CLICKED, UNSUBSCRIBED, COMPLAINTS, BOUNCES, SENT DATE, ACTIONS. It shows "No Stats Found".
- SMS Campaigns:** A table with columns: ID, NAME, RECIPIENTS, SMS DELIVERED, HARD + SOFT BOUNCES, REPLIED, SENT DATE, ACTIONS. It shows "No Stats Found".

Consulter notre **smtp** page pour prendre les infos nécessaires :

Configuration du fichier .env :

```

> resources
> routes
> storage
> tests
> vendor
> .editorconfig
> .env
> .env.example
> .gitattributes
> .gitignore
! styleci.yml
> artisan
composer.json
composer.lock
package.json
26  REDIS_HOST=127.0.0.1
27  REDIS_PASSWORD=null
28  REDIS_PORT=6379
29
30
31  MAIL_MAILER=smtp
32  MAIL_HOST=smtp-relay.sendinblue.com
33  MAIL_PORT=587
34  MAIL_USERNAME=abderrahimfizazi2019@gmail.com
35  MAIL_PASSWORD=mot de passe ici
36  MAIL_ENCRYPTION=ssl
37  MAIL_FROM_ADDRESS=abderrahimfizazi2019@gmail.com
38  MAIL_FROM_NAME="walo@gmail.com"
39
40  AWS_ACCESS_KEY_ID=
41  AWS_SECRET_ACCESS_KEY=
42  AWS_DEFAULT_REGION=us-east-1

```

Controller :

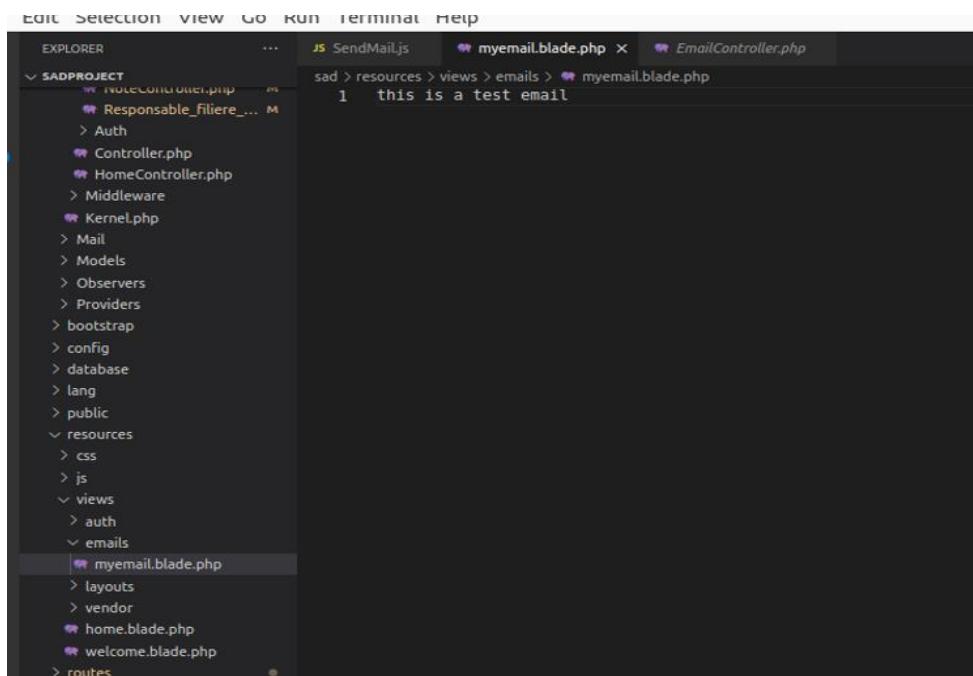
```
namespace App\Http\Controllers\API;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Mail\MyEmail;
use App\Models\Email;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Facades\Validator;

class EmailController extends Controller
{
    public function SendEmail(Request $request)
    {

        $data = [
            'Name' => $request->username,
            'Email' => $request->myEmail,
            'Query' => $request->body
        ];
        $subject = $request->subject;
        $dest = $request->dest;
        $username = $request->username;
        //Mail Function
        Mail::send('emails.myemail', ['data' => $data], function ($m) use ($subject, $dest, $username) {
            $m->from($dest, $username)->to($dest)->subject($subject);
        });
        //Json Response For Angular frontend
        return response()->json(["message" => "Email sent successfully."]);
    }
}
```

Création d'un fichier view :



Test de la requête par Postman :

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman Upgrade

Overview http://127.0.0.1:8000/ POST http://127.0.0.1:8000/ + ... No Environment Save Send

POST http://127.0.0.1:8000/api/sendMail?username=abdo&myEmail=abdo@fizazi.com&body=hahahah&subject=nothing&dest=abderrahim... Save Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
username	abdo		x	
myEmail	abdo@fizazi.com			
body	hahahah			
subject	nothing			
dest	abderrahim.fazazi@etu.uae.ac.ma			
src	abderrahim.fazazi@etu.uae.ac.ma			

Body Cookies Headers (10) Test Results Status: 200 OK Time: 1226 ms Size: 327 B Save Response

Pretty Raw Preview Visualize JSON

```
1 "message": "Email sent successfully."
```

Find and Replace Console Cookies Capture requests Bootcamp Runner Trash

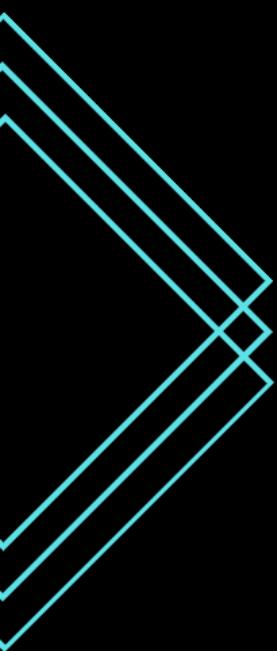
Test d'Email :

nothing Boîte de réception x

abdo via sendinblue.com
À moi ▾

this is a test email

Répondre Transférer



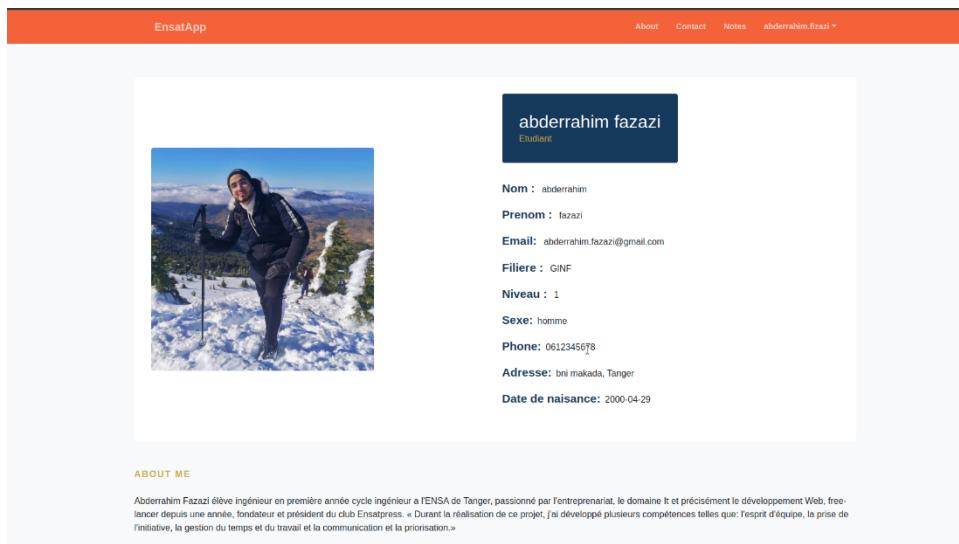
Chapitre 4

Espace Étudiant

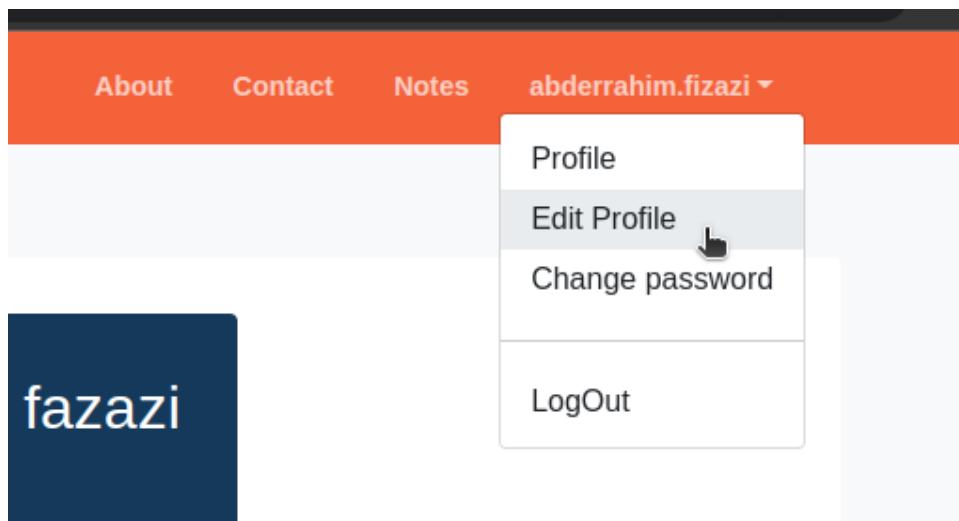
1-Consultation et modification du profil

Après sa connexion, l'étudiant est dirigé vers l'espace qui lui est conçu.

La première page qui s'affiche est la page d'accueil qui contient son profil :



Nous avons donné la possibilité aux étudiants de modifier leur profiles :



Il est également possible de modifier le mot de passe :

Orte

About Contact Notes abderrahim.fizazi ▾

Profile

Edit Profile

Change password

LogOut

EnsatApp

About Contact Notes mohamed ▾

Veuillez choisir une mot de passe forte

Current Password...
New Password...
Confirmed New Password...

Submit



SUITEZ VOTRE SUIVIEUR

Copyright © 2022 - Abderrahim / Maysaae

2-Consultation des notes et modules .

En ce qui concerne l'affichage des notes, il est présenté aux étudiants de la manière suivante :

EnsatApp

About Contact Notes abderrahim.fizazi ▾

Liste des Notes pour l'annee 2021/2022

Consulter vos modules pour cette année

Module	Element de module	Note
php	php poc	16.00
php	php tp	20.00

Copyright © 2022 - Abderrahim / Maysaae

Moyenne : 0

Il est aussi possible de consulter les modules qui lui sont enseignés :

The screenshot shows a web application interface with an orange header bar containing the text "EnsatApp". Below the header, a title "Liste des modules pour votre filière GINF Niveau 1" is displayed. A table follows, with columns "Module" and "Element module". The data in the table is as follows:

Module	Element module
php	php poo php tp
mysql	
linux	

At the bottom of the page, a copyright notice "Copyright © 2022 - Abderrahim / Maysae" is visible.

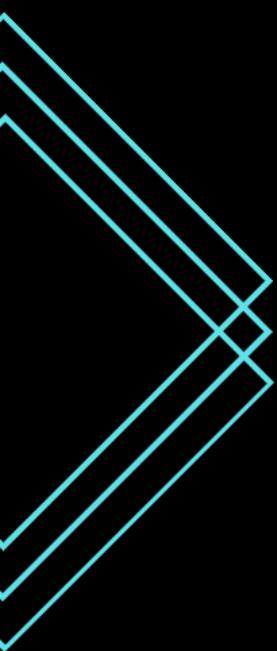
3-Envoi des emails aux chefs de filières

L'étudiant est dirigé vers cette interface pour envoyer un mail à la boîte mail de ses professeurs:

The form is titled "Pour envoyer des emails". It contains the following fields:

- Select Type destination
- Select Mail
- Sujet (Subject)
- Message (Message)

A large orange "SUBMIT" button is located at the bottom of the form.



Chapitre 5

Espace Chèf de filière

1-Saisie et modification des notes :

Au début, le chef de filière doit sélectionner la classe concernée par la

Veuillez choisir le niveau, le module et l'élément du module

Selectionnez la Niveau :

1

Selectionnez la Module :

Select Module

Selectionnez la Element :

Select Element

saisie des notes :

Insertion des notes

Filière : GINF
Niveau : 1
Module : php
Element : php poo
Pois : 0.25
VH : 10h

Nom	Prenom	Code	Note
patrick	guillemand	0123	13
abdo	fizazi	0123	18
abderrahim	fazazi	AP6500	020
Maysae	Hmamouchi	AP2456	12

Vous ne pouvez pas soumettre à moins d'avoir saisi toutes les valeurs!

Affichage des notes :

Les Notes de:

Filière: GINF
Niveau: 1
Module: php
Element: php poo
Pois: 0.25
VH: 10h

Nom	Prenom	Code	Note
patrick	guillemand	0123	14.00
abdo	fizazi	0123	15.00

modification des notes :

Les Notes de :

Filière : GINF
Niveau : 1
Module : php
Element : php poo
Poids : 0.25
VH : 10h

Nom	Prenom	Code	Note
patrick	guillemand	0123	<input type="text" value="14.00"/>
abdo	fizazi	0123	<input type="text" value="15.00"/>

vous ne pouvez pas soumettre à moins d'avoir saisi toutes les valeurs!



Chapitre 6

Espace Admin

1-Manipulations sur les étudiants :

Recherche d'un étudiant ou d'une classe :

The screenshot shows a user interface for searching students and classes. At the top right, there is a user profile icon with the email 'ana@ana.ana'. On the left, there is a sidebar menu with options like Home, Respo filieres, Eleves, Modules, Element Module, and Notes. The main area has three sections: 'Search for student' with a text input field and a green 'Search for student' button; 'Search for class' with two dropdown menus ('AP' and '1') and a green 'Search for class' button; and 'Add Eleve' with a green 'Add Eleve' button.

Recherche d'une classe par le nom de la filière et le niveau :

Il y a aussi possibilité de supprimer et modifier les étudiants affichés

The screenshot shows the same interface as above, but now displaying a table of student records. The table has columns for Nom, Prenom, Code, Filiere, Niveau, Edit, and Delete. The data is as follows:

Nom	Prenom	Code	Filiere	Niveau	Edit	Delete
patrick	guillemand	0123	GINF	1		
abdo	fazazi	0123	GINF	1		
abderrahim	fazazi	AP6500	GINF	1		
Maysae	Hmamouchi	AP2456	GINF	1		

Recherche d'un étudiant par nom, prénom ou code :

The screenshot shows a search interface for students. On the left, there's a sidebar with navigation links: Home, Respo filières, Eleves, Modules, Element Module, and Notes. The main area has three search fields: 'Search for student' (containing 'abdo'), 'Search for class' (containing 'GINF' and '1'), and 'Add Eleve'. A button labeled 'Search for student' is visible. To the right, a table titled 'Résultats de la recherche de : "abdo"' displays student information: Nom (abdo), Prénom (fizazi), Code (0123), Filière (GINF), Niveau (1), Edit (green pencil icon), and Delete (red trash icon). The table has columns for Nom, Prénom, Code, Filière, Niveau, Edit, and Delete.

Ajout d'un étudiant :

The screenshot shows a form titled 'Add Eleve'. It includes fields for Code, Nom, Prénom, Select Sexe (dropdown menu), Login, Email, Select Filière (dropdown menu), Select Niveau (dropdown menu), and Image (file input field with 'Choose File' placeholder). A note at the bottom states: 'En créant un étudiant on crée aussi un utilisateur avec le login saisi est une mot de passe qui va être envoyé automatiquement par un email !!!'. A 'Submit' button is located at the bottom right.

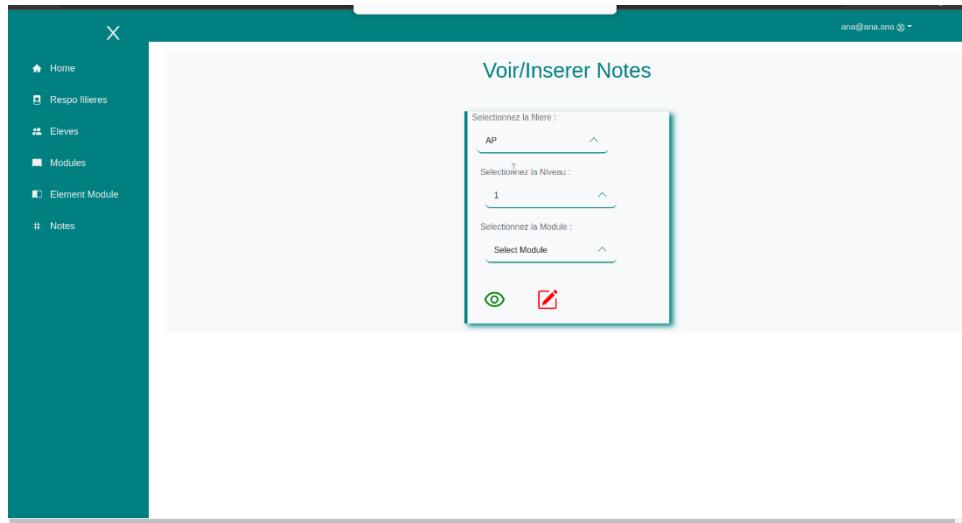
Modification d'un élève :

The screenshot shows a form titled 'Update Eleve'. It includes fields for Code (0123), Nom (patrick), Prénom (guillemand), Select Filière (dropdown menu containing 'GINF'), and Select Niveau (dropdown menu containing '1'). A 'Submit' button is located at the bottom right.

2-Manipulations sur les notes :

L'admin a les même privilège qu'un responsable de filière mais **il a accès à toutes les données de toutes les filières**.

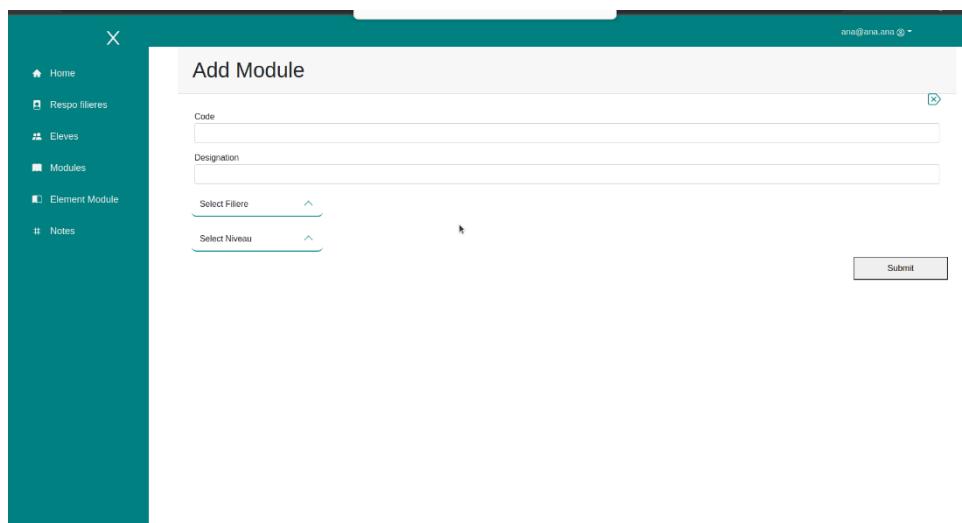
Choix du niveau, filière, module et élément de module pour la manipulation des notes :



(voir cette section dans le chapitre précédent)

3-Manipulations sur les modules et éléments de modules :

Ajout d'un module :



Modification d'un module :

The screenshot shows a 'Update Module' form. On the left is a sidebar with navigation links: Home, Respo filières, Eleves, Modules, Element Module, and Notes. The main area has a title 'Update Module'. It contains four input fields: 'Code' (1010), 'Designation' (php), 'Niveau' (GINF), and 'Semestre' (1). A 'Submit' button is at the bottom right.

L'affichage de la liste des modules:

Il y a aussi possibilité de supprimer et modifier les modules affichés

The screenshot shows a 'Modules list' page. On the left is a sidebar with navigation links: Home, Respo filières, Eleves, Modules, Element Module, and Notes. The main area has a title 'Modules list' and a table. The table has columns: Code, Designation, Niveau, Filiere, Semestre, Edit, and Delete. There are three rows of data:

Code	Designation	Niveau	Filiere	Semestre	Edit	Delete
1010	php	1	GINF	1		
1212	plusql	1	GINF	1		
1212	linux	1	GINF	1		

4-Manipulations sur les responsables de filières :

Ajout d'un responsable :

The screenshot shows a web application interface for adding a responsible person. The URL in the address bar is `localhost:3000/admin/add_respo`. The page title is "Add Responsable". On the left, there is a sidebar with navigation links: Home, Respo filieres, Eleves, Modules, Element Module, and Notes. The main content area contains fields for Nom, Prénom, Select Sexe, Login, Email, Département, Select Filière*, and Image (with a "Choose File" button). A note at the bottom states: "En creant un responsable d'une filiere on cree aussi un utilisateur avec le login saisi est une mot de passe qui va etre envoyee automatiquement par un email !!!". A "Submit" button is located on the right.

Modification des informations d'un responsable :

The screenshot shows a web application interface for editing a responsible person's information. The URL in the address bar is `localhost:3000/admin/update_respo/1`. The page title is "Edit Responsable". The sidebar on the left is identical to the one in the previous screenshot. The main content area displays fields for Nom (ghallani), Prénom (mohamed), Select Filière* (GINF), and Département (It). A "Submit" button is located on the right.

Affichage des informations des responsables :

Il y a aussi possibilité de supprimer et modifier les responsables affichés

X

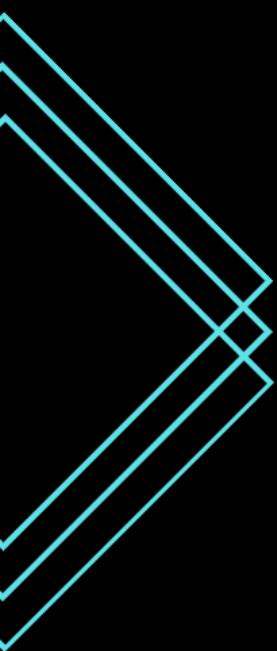
ana@ana.ana.az ▾

- [Home](#)
- [Respo filieres](#)
- [Eleves](#)
- [Modules](#)
- [Element Module](#)
- [Notes](#)

Responsables des filieres

[Add Responsable](#)

Image	Nom	Prenom	Login	Filiere	Departement	Edit	Delete	Created date
	ghalani	mohamed	mohamed	GINF	It			June 8th



Chapitre 7

Tests et Comparaisons

1-Test connexion PHP et Oracle sous Apache:

Installation des packages pour connecter le code php avec Oracle:

Installation de instantClient:

```
maysae2710@maysae2710-virtual-machine:~/Downloads$ sudo mv instantclient-basic-linux.x64-21.6.0.0.0dbru.zip /opt/oracle/
maysae2710@maysae2710-virtual-machine:~/Downloads$ sudo mv instantclient-sdk-linux.x64-21.6.0.0.0dbru.zip /opt/oracle/
maysae2710@maysae2710-virtual-machine:~/Downloads$ cd /opt/oracle
```

```
maysae2710@maysae2710-virtual-machine:/opt/oracle$ sudo unzip instantclient-basic-linux.x64-21.6.0.0.0dbru.zip
Archive: instantclient-basic-linux.x64-21.6.0.0.0dbru.zip
  inflating: instantclient_21_6/adrci
  inflating: instantclient_21_6/BASIC_LICENSE
  inflating: instantclient_21_6/BASIC_README
  inflating: instantclient_21_6/genezi
    linking: instantclient_21_6/libclntshcore.so -> libclntshcore.so.21.1
    linking: instantclient_21_6/libclntshcore.so.12.1 -> libclntshcore.so.21.1
    linking: instantclient_21_6/libclntshcore.so.18.1 -> libclntshcore.so.21.1
    linking: instantclient_21_6/libclntshcore.so.19.1 -> libclntshcore.so.21.1
    linking: instantclient_21_6/libclntshcore.so.20.1 -> libclntshcore.so.21.1
  inflating: instantclient_21_6/libclntshcore.so.21.1
    linking: instantclient_21_6/libclntsh.so -> libclntsh.so.21.1
    linking: instantclient_21_6/libclntsh.so.10.1 -> libclntsh.so.21.1
    linking: instantclient_21_6/libclntsh.so.11.1 -> libclntsh.so.21.1
    linking: instantclient_21_6/libclntsh.so.12.1 -> libclntsh.so.21.1
    linking: instantclient_21_6/libclntsh.so.18.1 -> libclntsh.so.21.1
    linking: instantclient_21_6/libclntsh.so.19.1 -> libclntsh.so.21.1
    linking: instantclient_21_6/libclntsh.so.20.1 -> libclntsh.so.21.1
  inflating: instantclient_21_6/libclntsh.so.21.1
  inflating: instantclient_21_6/libnznz21.so
    linking: instantclient_21_6/libocci_gcc53.so -> libocci_gcc53.so.21.1
  inflating: instantclient_21_6/libocci_gcc53.so.21.1
    linking: instantclient_21_6/libocci.so -> libocci.so.21.1
    linking: instantclient_21_6/libocci.so.10.1 -> libocci.so.21.1
    linking: instantclient_21_6/libocci.so.11.1 -> libocci.so.21.1
    linking: instantclient_21_6/libocci.so.12.1 -> libocci.so.21.1
    linking: instantclient_21_6/libocci.so.18.1 -> libocci.so.21.1
    linking: instantclient_21_6/libocci.so.19.1 -> libocci.so.21.1
    linking: instantclient_21_6/libocci.so.20.1 -> libocci.so.21.1
```

```
maysae2710@maysae2710-virtual-machine:/opt/oracle$ sudo unzip instantclient-sdk-linux.x64-21.6.0.0.0dbru.zip
Archive: instantclient-sdk-linux.x64-21.6.0.0.0dbru.zip
  inflating: instantclient_21_6/SDK README
  creating: instantclient_21_6/sdk/
  extracting: instantclient_21_6/sdk/ottclasses.zip
  inflating: instantclient_21_6/sdk/ott
  creating: instantclient_21_6/sdk/admin/
  inflating: instantclient_21_6/sdk/admin/oraaccess.xsd
  creating: instantclient_21_6/sdk/demo/
  inflating: instantclient_21_6/sdk/demo/demo.mk
  inflating: instantclient_21_6/sdk/demo/oraaccess.xml
  inflating: instantclient_21_6/sdk/demo/occidemo.sql
  inflating: instantclient_21_6/sdk/demo/occidemod.sql
  inflating: instantclient_21_6/sdk/demo/cdemo01.c
  inflating: instantclient_21_6/sdk/demo/occobj.cpp
  inflating: instantclient_21_6/sdk/demo/occidml.cpp
  inflating: instantclient_21_6/sdk/demo/occijobj.typ
  inflating: instantclient_21_6/sdk/demo/setuporamysql.sh
  creating: instantclient_21_6/sdk/include/
  inflating: instantclient_21_6/sdk/include/xmlurl.h
  inflating: instantclient_21_6/sdk/include/ocixmlldb.h
  inflating: instantclient_21_6/sdk/include/ort.h
  inflating: instantclient_21_6/sdk/include/oraxml.hpp
  inflating: instantclient_21_6/sdk/include/xmlotn.hpp
  inflating: instantclient_21_6/sdk/include/oriid.h
  inflating: instantclient_21_6/sdk/include/xmlsoapc.hpp
  inflating: instantclient_21_6/sdk/include/ociixstream.h
  inflating: instantclient_21_6/sdk/include/oci.h
  inflating: instantclient_21_6/sdk/include/nzerror.h
  inflating: instantclient_21_6/sdk/include/oraxsd.h
  inflating: instantclient_21_6/sdk/include/xmlev.h
```

On passe au root à cause des permissions pour effectuer l'installation de
php-dev php-pear build-essential libaio1 :

```
maysae2710@maysae2710-virtual-machine:/opt/oracle$ su
Password:
root@maysae2710-virtual-machine:/opt/oracle# sudo echo /opt/oracle/instantclient_21_6 > /etc/ld.so.conf.d/oracle-instantclient.conf
root@maysae2710-virtual-machine:/opt/oracle# ldconfig
root@maysae2710-virtual-machine:/opt/oracle# sudo apt install php-dev php-pear build-essential libaio1
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9ubuntu3).
libaio1 is already the newest version (0.3.112-13build1).
The following additional packages will be installed:
  libpcre2-16-0 libpcre2-dev libpcre2-posix3 libssl-dev libss13 php-json php8.1-dev pkg-config pkg-php-tools shtool
Suggested packages:
  libssl-doc dh-php dh-make
The following NEW packages will be installed:
  libpcre2-16-0 libpcre2-dev libpcre2-posix3 libssl-dev php-dev php-json php-pear php8.1-dev pkg-config pkg-php-tools shtool
The following packages will be upgraded:
  libssl3
1 upgraded, 11 newly installed, 0 to remove and 82 not upgraded.
Need to get 4,303 kB/6,202 kB of archives.
```

On active l'extension dans le fichier `php.ini`

```
root@maysae2710-virtual-machine:/etc/php/8.1/apache2# nano php.ini
```

On élimine « ; » pour activer l'extension `oci8_12c` :

```
GNU nano 6.2                                         php.ini *
extension=bz2
extension=curl
extension=ffl
extension=ftp
extension=fileinfo
extension=gd
extension=gettext
extension=gmp
extension=intl
extension=imap
extension=ldap
extension=mbstring
extension=exif      ; Must be after mbstring as it depends on it
extension=mysqli
extension=oci8_12c ; Use with Oracle Database 12c Instant Client
extension=oci8_19  ; Use with Oracle Database 19 Instant Client
extension=odbc
extension=openssl
extension=pdo_firebird
extension=pdo_mysql
extension=pdo_oci
extension=pdo_odbc
extension=pdo_pgsql
extension=pdo_sqlite
extension=pgsql
extension=shmop

The MIBS data available in the PHP distribution must be installed.
See https://www.php.net/manual/en/snmp.installation.php

G Help          A Write Out    W Where Is     K Cut           T Execute      C Location    U Undo        M-A Set Mark   M-
X Exit         R Read File   ^R Replace    ^U Paste       J Justify     G Go To Line  E Redo        M-6 Copy      Q

```

```
maysae2710@maysae2710-virtual-machine:/Laravel/projet$ composer require yajra/laravel-oci8 --ignore-platform-req=ext-oci8
PHP Warning: Module "curl" is already loaded in Unknown on line 0
Using version ^9.0 for yajra/laravel-oci8
./composer.json has been updated
Running composer update yajra/laravel-oci8
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
- Locking yajra/laravel-oci8 (v9.0.1)
- Locking yajra/laravel-pdo-via-oci8 (v3.2.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
- Downloading yajra/laravel-pdo-via-oci8 (v3.2.1)
- Downloading yajra/laravel-oci8 (v9.0.1)
- Installing yajra/laravel-pdo-via-oci8 (v3.2.1): Extracting archive
- Installing yajra/laravel-oci8 (v9.0.1): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
PHP Warning: Module "curl" is already loaded in Unknown on line 0
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: spatie/laravel-ignition
Discovered Package: yajra/laravel-oci8
Package manifest generated successfully.
79 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
```

Add to providers :

```
Illuminate\Session\SessionServiceProvider::class,
Illuminate\Translation\TranslationServiceProvider::class,
Illuminate\Validation\ValidationServiceProvider::class,
Illuminate\View\ViewServiceProvider::class,

/*
 * Package Service Providers...
 */

/*
 * Application Service Providers...
 */
App\Providers\AppServiceProvider::class,
App\Providers\AuthServiceProvider::class,
// App\Providers\BroadcastServiceProvider::class,
App\Providers\EventServiceProvider::class,
App\Providers\RouteServiceProvider::class,
Yajra\Oci8\Oci8ServiceProvider::class,
```

],

/*

```
maysae2710@maysae2710-virtual-machine:/Laravel/projet$ php artisan vendor:publish --tag=oracle
PHP Warning: Module "curl" is already loaded in Unknown on line 0
Copied File [/vendor/yajra/laravel-oci8/src/config/oracle.php] To [/config/oracle.php]
Publishing complete.
```

On fait les modifications suivantes au niveau du projet Laravel :
config/database.php

```
'oracle' => [
    'driver'      => 'oracle',
    'tns'         => env('DB_TNS', ''),
    'host'        => env('DB_HOST', ''),
    'port'        => env('DB_PORT', '1521'),
    'database'   => env('DB_DATABASE', ''),
    'username'   => env('DB_USERNAME', ''),
    'password'   => env('DB_PASSWORD', ''),
    'charset'    => env('DB_CHARSET', 'AL32UTF8'),
    'prefix'     => env('DB_PREFIX', ''),
    'prefix_schema' => env('DB_SCHEMA_PREFIX', ''),
    'edition'    => env('DB_EDITION', 'ora$base'),
],
],
```

Et au niveau du fichier : config/auth.php On ajoute :

```
'providers' => [
    'users' => [
        'driver' => 'oracle',
        'model'  => App\User::class,
    ],
]
```

Test simple de connexion :

Nous avons utilisé l'objet PDO

```

<?php

$pdo =new PDO('oci:dbname=//localhost:1521/orcl','ginf1','123456');

$sql="insert into ginf1.ensat(id,name,passwd) values(110,'Maysae','123456789')";
$test=$pdo->exec($sql);

$res=$pdo->query("select * from ginf1.ensat ");

while ($ligne=$res->fetch()){
    echo"=====\\n";
    echo $ligne['ID'];
    echo "\\n";
    echo $ligne['NAME'];
    echo "\\n";
    echo $ligne['PASSWD'];
    echo "\\n";
    echo"=====\\n";
}

```

Résultat:

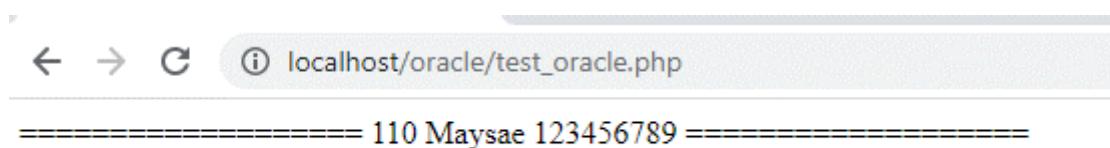
```

=====
110
Maysae
123456789
=====

[Done] exited with code=0 in 1.404 seconds

```

Résultat sur le navigateur:



2-Test Nginx:

Test de connexion :

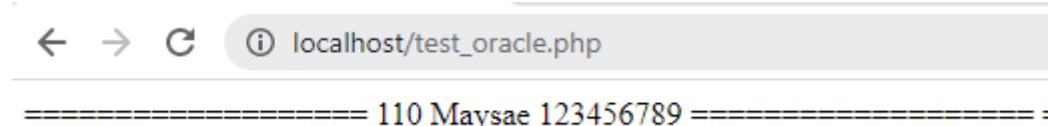
Nous avons utilisé l'objet PDO

```

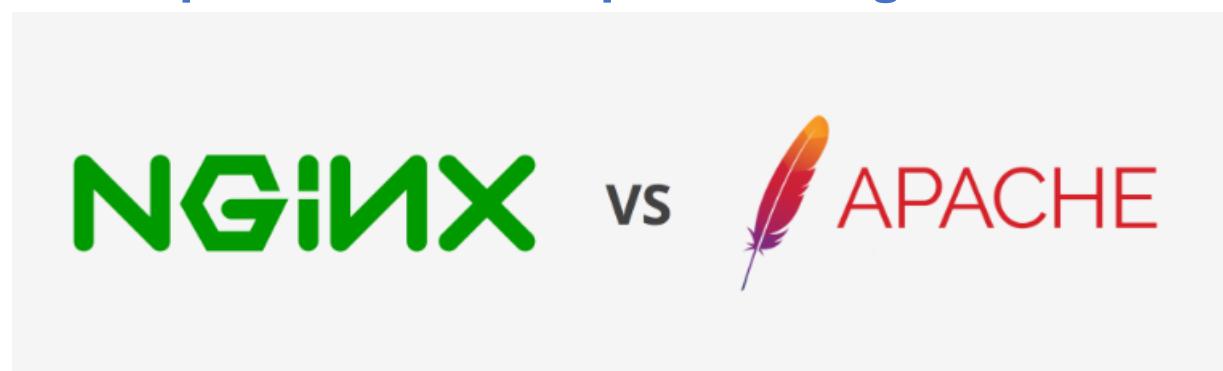
1  <?php
2  ^ try{
3      $pdo =new PDO('oci:dbname=//localhost:1521/orcl','ginf1','123456');
4  }
5  ^ catch(PDOException $e){
6      echo $e->getMessage();
7  }
8  $sql="insert into ginf1.ensat(id,name,passwd) values(111,'Maysae','123456789')" ;
9  $test=$pdo->exec($sql);
10
11 $res=$pdo->query("select * from ginf1.ensat ");
12
13
14 ^ while ($ligne=$res->fetch()){
15     echo"=====\\n";
16     echo $ligne['ID'];
17     echo "\\n";
18     echo $ligne['NAME'];
19     echo "\\n";
20     echo $ligne['PASSWD'];
21     echo "\\n";
22     echo"=====\\n";
23 }
24
25 ?>

```

Le résultat obtenu depuis le navigateur :



3-Comparaison entre Apache et Nginx:



Caractéristiques des deux serveurs :

Avant de faire une comparaison entre les deux, nous allons citer des points essentiels sur chacun :

Caractéristiques d'Apache:

- L'Apache Software Foundation (ASF) gère elle-même sa marque et ses services.
- Le serveur est développé, mais également maintenu par une communauté de développeurs sous la tutelle de la Fondation Apache ou ASF.
- Généralement viable sur LINUX, Apache dessert à peu près les 46 % des sites web du monde.
- C'est le composant-clé de la Pile LAMP (Linux, Apache, MySQL, PHP)

Caractéristiques de Nginx :

- Au début, NGINX était utilisé comme un complément d'Apache
- Il a été usuellement sollicité pour traiter des fichiers statiques, mais actuellement, il a évolué et est devenu un serveur web qui traite les spectres complets des tâches du serveur.
- Désormais, NGINX est devenu comme un proxy inversé, un équilibrEUR de charges, un proxy de messagerie et un outil de mise en cache http.

Comparaison entre les deux au niveau de l'architecture et la rapidité:

La plus grande différence entre Apache et NGINX repose sur le design architectural. Ils se discernent par leurs moyens de traiter des connexions, des données sur le trafic et de répondre aux différentes conditions des serveurs : pour Apache, il utilise une architecture qui produit une multitude de modules. Ces pré-modules sont au nombre de trois selon l'algorithme de traitement des requêtes. Chacun est utilisé différemment selon les besoins du serveur et chacun peut servir une seule requête à la fois. De l'autre côté, NGINX suit une approche basée sur les événements et traite les requêtes d'une manière asynchrone. Ainsi, il peut exécuter des milliers de requêtes en un seul thread. Une telle procédure modulaire permet à NGINX de réaliser très rapidement les opérations et de déployer le minimum de ressources.

Illustration qui explique l'architecture d'Apache :

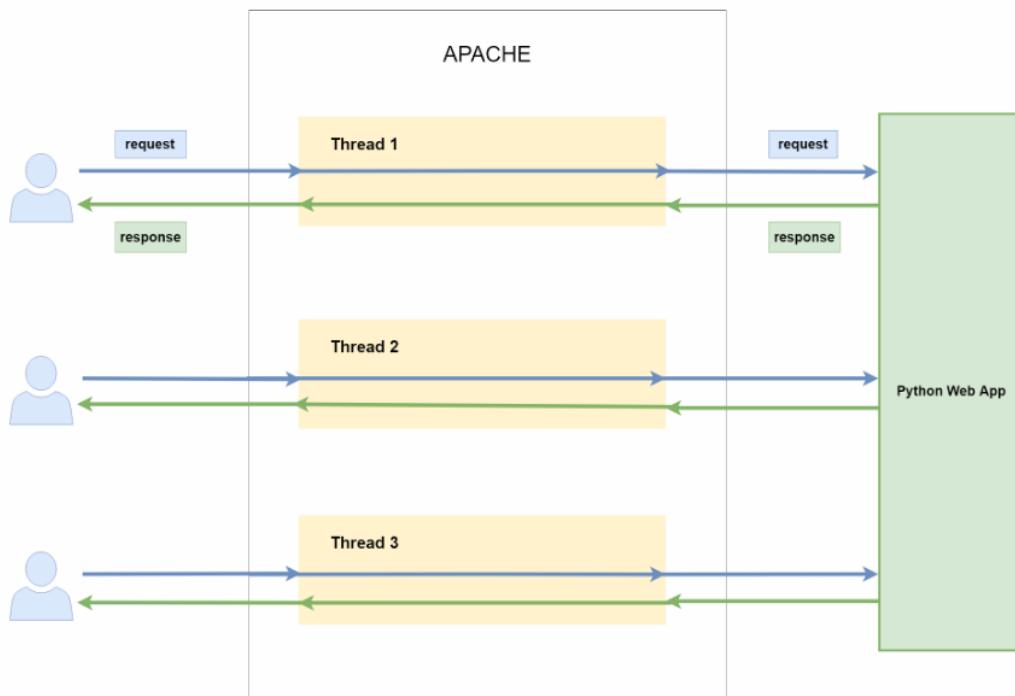
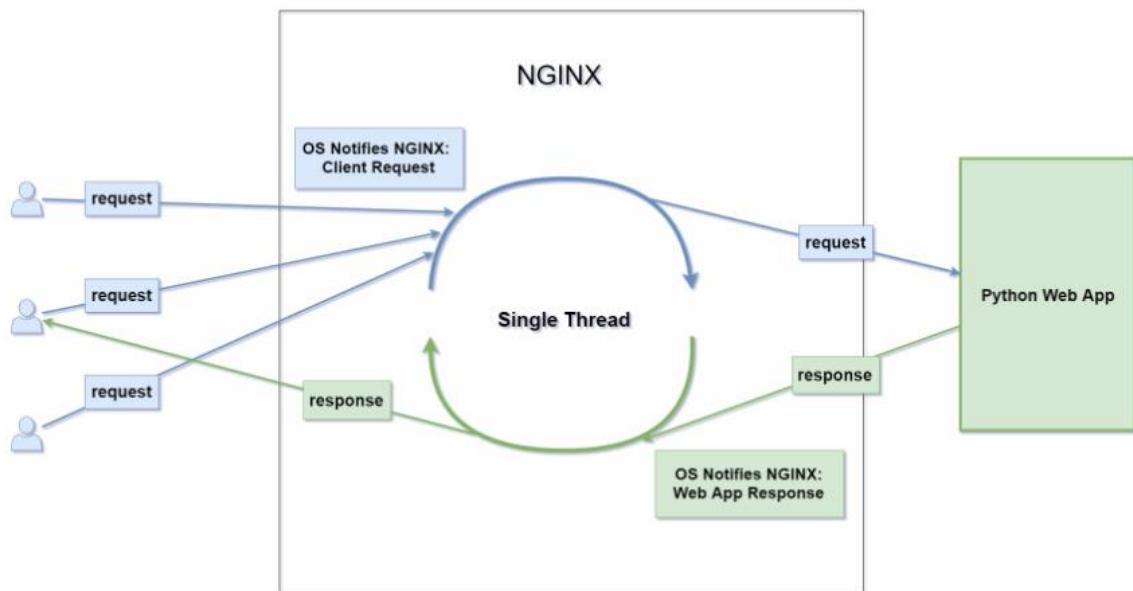


Illustration qui explique l'architecture de Nginx :



- Nous pouvons dire que Nginx a une structure légère et plus rapide que celle d'Apache.

De plus Nginx est plus performant pour le traitement statique mais il est incapable de faire un traitement dynamique, chose qu'Apache peut faire au sein du serveur lui-même sans avoir recours à des composants externes.

- Nous constatons qu'Apache a l'avantage d'être utilisé pour traiter les contenus dynamiques.

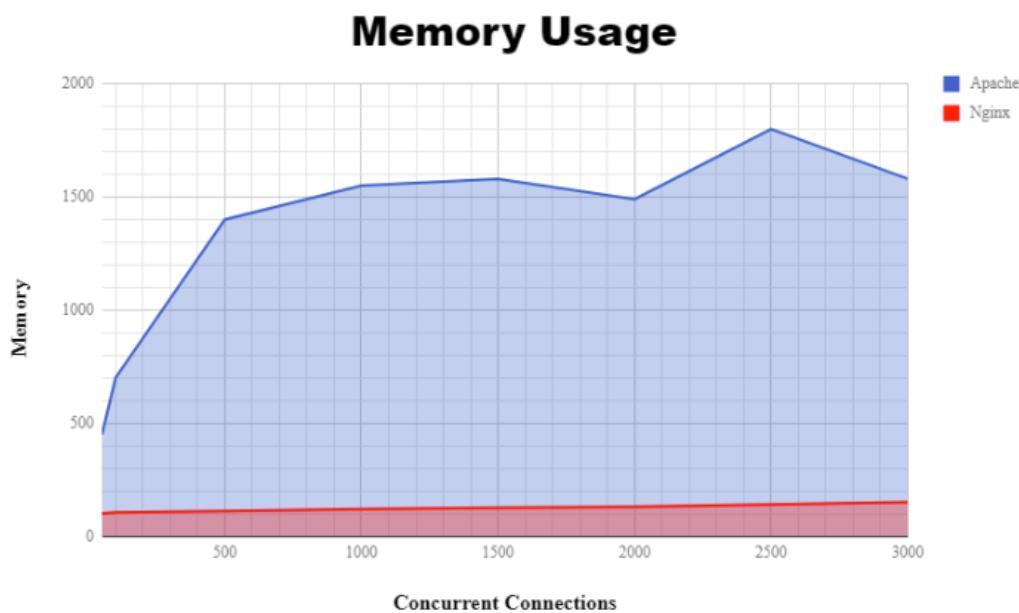
Comparaison d'efficacité en termes d'exigences système :

Nginx sert à traiter les ressources statiques sans l'intervention de PHP. Apache traite toutes ces demandes avec une surcharge plus coûteuse sur les ressources du serveur. Cela rend Nginx plus efficace et moins exigeant pour les ressources système.

Cette image montre le nombre de demandes de contenu statique traitées par seconde.



Illustration qui monte l'utilisation en mémoire des deux serveurs



- Nginx surpassé clairement Apache ici! NGINX est d'autant plus efficace et moins exigeant sur le CPU et le RAM du serveur.

Comparaison d'efficacité en termes d'exigences système :

Apache permet des configurations additionnelles via `htaccess`

Nginx ne permet pas de faire des configurations additionnelles ce qui lui permet de servir les fichiers plus vite qu'Apache.

- Au contraire d'Apache, Nginx ne permet pas des configurations additionnelles ce qui augmente sa performance.

Comparaison en termes de sécurité :

Apache assure que tous les sites web du serveur sont protégés des menaces de piratages et autres menaces.

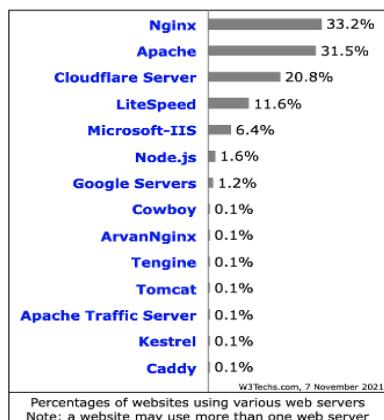
Ainsi, il offre des astuces de configuration pour la gestion des attaques par Dénie de service, tout comme le module *mod_evasive* pour répondre aux attaques http, DoS, DDoS ou autres types.

Cependant, Nginx apparaît plus puissant en offrant une meilleure sécurité avec une base-code plus fine. Le code base de NGINX est toutefois significativement plus petit, donc celui-ci est plus avantageux en termes de sécurité. NGINX a aussi une liste de balises sécuritaires.

- Nginx, encore une fois, gagne la bataille, cette fois-ci, en offrant une meilleure sécurité par rapport à Apache.

Comparaison en termes de nombre des utilisateurs :

Cette illustration montre le pourcentage des utilisateurs des serveurs les plus connus à travers le monde :



4-Test Postfix:

On vérifie le status d'abord :

```
abderrahimfizazi@abderrahimfizazi:~$ sudo postfix status
[sudo] password for abderrahimfizazi:
postfix: warning: /etc/postfix/main.cf, line 47: overriding earlier entry: relayhost=
/usr/sbin/postconf: warning: /etc/postfix/main.cf, line 47: overriding earlier entry: relayhost=
postfix/postlog: warning: /etc/postfix/main.cf, line 47: overriding earlier entry: relayhost=
postfix/postfix-script: the Postfix mail system is running: PID: 2647
abderrahimfizazi@abderrahimfizazi:~$ sudo postfix start
postfix: warning: /etc/postfix/main.cf, line 47: overriding earlier entry: relayhost=
/usr/sbin/postconf: warning: /etc/postfix/main.cf, line 47: overriding earlier entry: relayhost=
postfix/postlog: warning: /etc/postfix/main.cf, line 47: overriding earlier entry: relayhost=
postfix/postfix-script: fatal: the Postfix mail system is already running
abderrahimfizazi@abderrahimfizazi:~$
```

```
GNU nano 6.2                               /etc/postfix/sasl_passwd *
[smtp.gmail.com]:587      abderrahimfizazi2019@gmail.com:PASSWORD
```

```
abderrahimfizazi@abderrahimfizazi:~$ sudo nano /etc/postfix/main.cf\
abderrahimfizazi@abderrahimfizazi:~$
```

On utilise la commande :

```
cat ~/test_message | s-nail -s 'Test email subject line' -r
maysaehmamouchi@gmail.com abderrahimfizazi2019@gmail.com
```

suivie de : s-nail

et file +sent

On reçoit le mail :

23:23

88



Test postfix

Boîte de réception



abderrahim/maysaa 19 mai
à moi ▾



Test test

5-Comparaison entre Sendmail et Postfix:



Les caractéristiques de SendMail :

- **Sendmail est un serveur de messagerie électronique dont le code source est ouvert.**

- Il se charge de la livraison et de l'envoi de **courriers électroniques (courriels)**.
- Il est fiable et il peut gérer des environnements complexes.
- Il permet à chaque utilisateur de désigner un logiciel de messagerie personnalisé pour traiter le courrier entrant.

Caractéristiques de Postfix :

- Postfix est le MTA (Mail Transfer Agent) gratuit et open source le plus couramment utilisé pour décider des itinéraires et envoyer des e-mails.
- Il s'agit d'un serveur de messagerie multiplateforme qui peut être installé sur la plupart des systèmes d'exploitation de type UNIX.
- Postfix se compose de nombreux programmes serveur et client, dans lesquels les programmes serveur s'exécutent principalement dans le backend.
- les programmes clients sont invoqués par l'administrateur système ou les programmes utilisateur.

Comparaison au niveau de l'architecture:

Postfix a une architecture modulaire composée de nombreux petits exécutables indépendants. Il fournit plusieurs options, paramètres et fonctionnalités. En revanche, Sendmail a une conception monolithique qui utilise un seul processus toujours exécuté au niveau du backend. Sendmail est en train de disparaître par rapport à Postfix car beaucoup préfèrent l'architecture de Postfix.

Illustration qui montre l'architecture de SendMail :

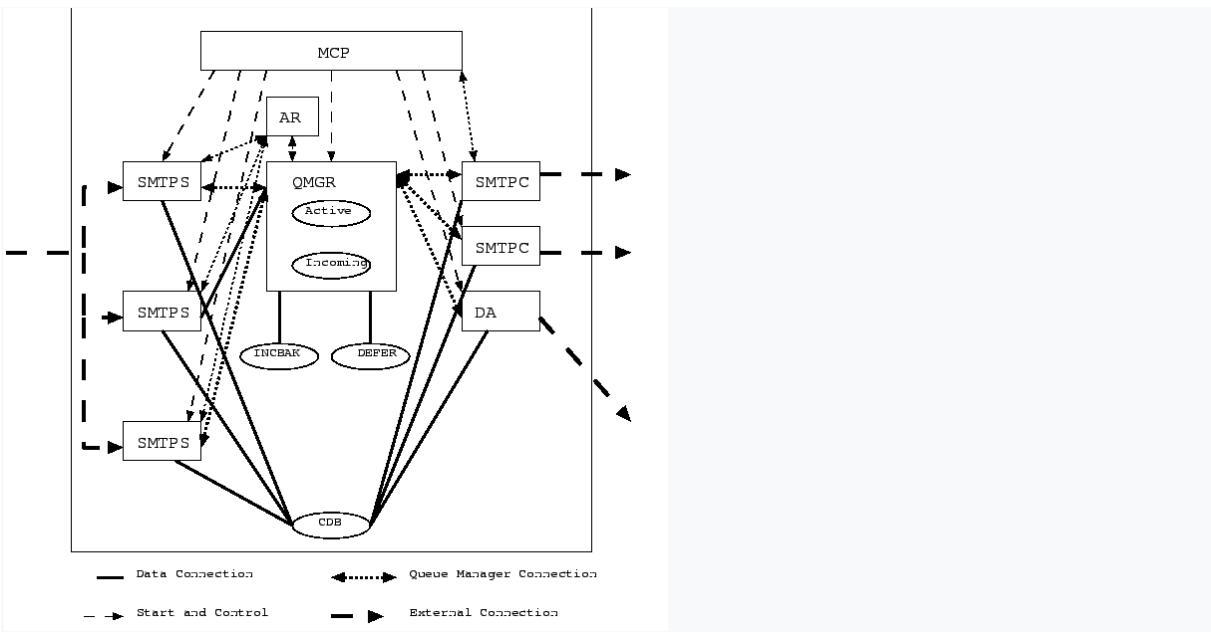
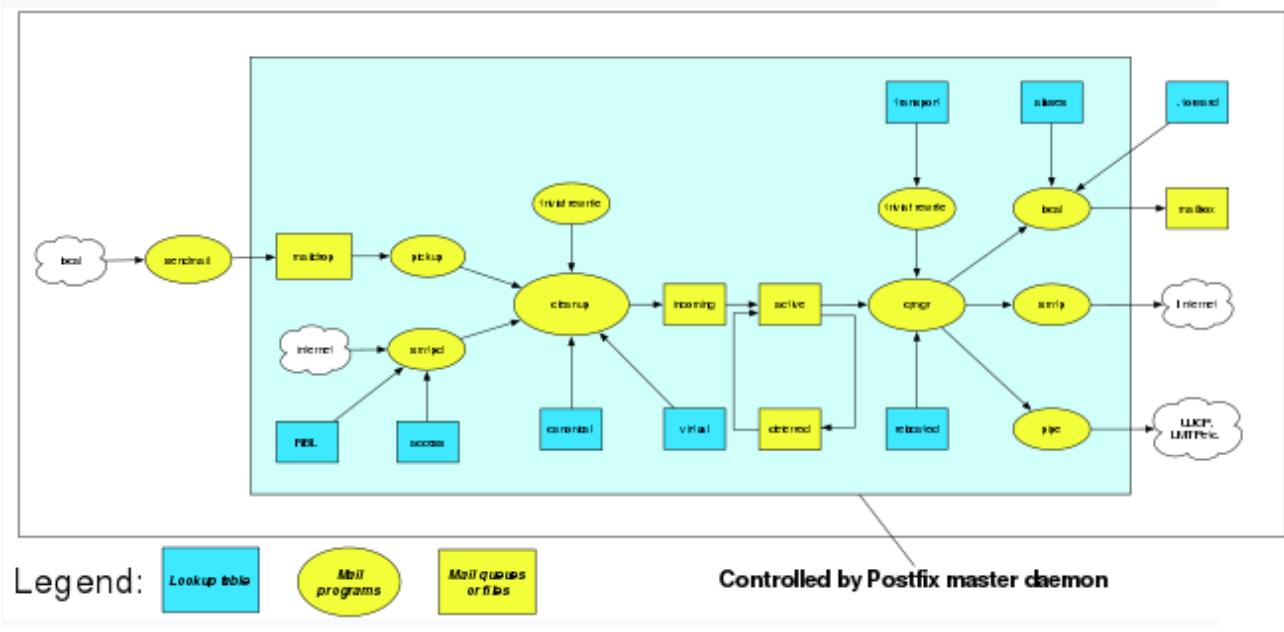


Illustration qui montre l'architecture de Postfix :



Comparaison au niveau de la vitesse et la performance :

Sendmail n'est pas très performant en termes de performances, ce qui en fait un piètre concurrent. Postfix dispose d'un gestionnaire de file d'attente qui gère la file d'attente beaucoup plus rapidement. Certains administrateurs affirment que même avec une charge élevée, Postfix est plus efficace que Sendmail.

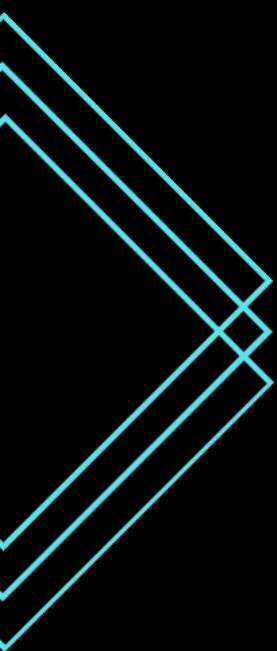
Comparaison au niveau de la sécurité :

Sendmail est né l'aube moderne de l'Internet, lorsque la sécurité était pas un problème aussi bien qu'entendu aujourd'hui. Les premières versions de Sendmail ont plusieurs vulnérabilités - par exemple, le ver Internet Morris exploité une **débordement de tampon** Sendmail. Aujourd'hui Sendmail, correctement configuré, est pas moins (ou plus) sûr que d'autres serveurs de messagerie modernes comme **Postfix**. Ce dernier, représente une alternative beaucoup plus puissante car il est conçu pour surmonter les vulnérabilités associées à Sendmail. De plus, une bonne configuration de Postfix sécurise les données sensibles contre le spam, les abus et les fuites.

Comparaison au niveau du nombre des utilisateurs :

Sendmail est toujours le plus populaire serveur de messagerie Internet, bien qu'il soit en train de perdre des positions. Sa popularité est probablement due au fait qu'il est le standard Mail Server La plupart des variantes Unix. Selon une étude de 2001, environ 42% du serveur de messagerie Internet accessible utilisé Sendmail. dernière enquête montre une baisse de l'utilisation de Sendmail, avec une part de 33% en Décembre 2006 . Sendmail est suivi comme un écart de Microsoft Exchange, Exim, et Postfix; ces quatre serveurs de messagerie sont les seuls à avoir un taux d'utilisation plus élevé de 10%.

Donc en terme de popularité, nous pouvons dire que Sendmail reste le premier.

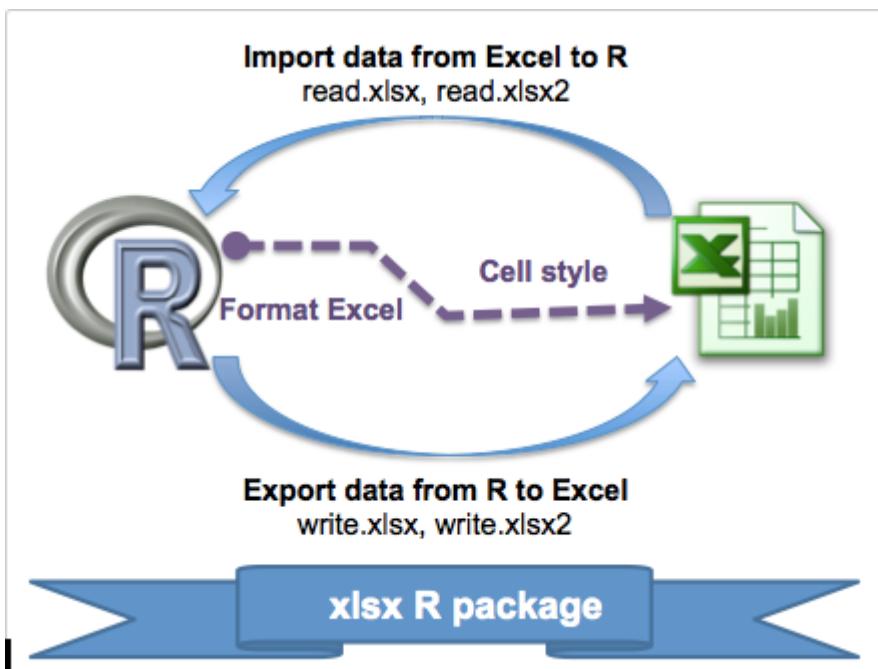


Chapitre 8

Fonctionnalités ajoutées à l'application

1-Exportation/Importation des notes via des fichiers Excel :

Dans le cadre de l'amélioration de la qualité de notre application et des services offerts, nous avons décidé de faire appel à un package très puissant et qui permet exportation et téléchargement des notes des étudiants via des fichiers Excel. Cette fonctionnalité va certainement optimisé le temps de la saisie des notes et elle va donner plus de fiabilité aux données saisies en évitant les erreurs de frappes.



Le package que nous avons décidé d'utiliser s'appelle xlsx et il s'agit xlsx package est l'un des puissants packages R pour lire, écrire et formater des fichiers Excel. C'est une solution basée sur Java et elle est disponible pour Windows, Mac et Linux. Cela fonctionne pour toutes les version d'Excel

Installation du package :

```
abderrahimfizazi@abderrahimfizazi:~/sadproject$ npm install xlsx
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: '@emailjs/browser@3.6.2',
npm WARN EBADENGINE   required: { node: '>=14.0.0' },
npm WARN EBADENGINE   current: { node: 'v12.22.9', npm: '8.5.1' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'eslint-config-react-app@7.0.1',
npm WARN EBADENGINE   required: { node: '>=14.0.0' },
npm WARN EBADENGINE   current: { node: 'v12.22.9', npm: '8.5.1' }
npm WARN EBADENGINE }
```

Nous avons créer un bouton sur les interfaces du cadre administratif ainsi que les chefs de filière pour qu'ils puissent exporter les notes :

The screenshot shows a web page with a light gray background. At the top right is a green hexagonal button with a white 'X' icon. Below it, the text "Les Notes de:" is displayed in green. Underneath, several course details are listed in black text: "Filiere: GINF", "Niveau: 1", "Module: php", "Element: php poo", "Pois: 0.25", and "VH: 10h". At the bottom right of the page is a green rectangular button with the text "Download as XLS" in white.

Une fois le click est fait, le fichier Excel est téléchargé et peut être facilement manipulé :

Le code utilisé pour cette fin :

```
<ReactHTMLTableToExcel  
    id="walo"  
    className="btn btn-success float-end my-5"  
    table="my_table"  
    filename="tablexls"  
    sheet="tablexls"  
    buttonText="Download as XLS" />
```

2-Renforcement de sécurité des mots de passe des étudiants et responsables de filières:

Pour des raisons de sécurité, nous avons décidé de générer des mots de passe automatiques dédiés aux étudiants et responsables de filières et qui leur sont envoyé via email institutionnel.

Nous avons utilisé pour cela, la fonction Random() :

```
const Random = () => {
  var result      = [];
  var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  var charactersLength = characters.length;
  for ( var i = 0; i < 8; i++ ) {
    result.push(characters.charAt(Math.floor(Math.random() * charactersLength)));
  }
  return result.join('');
}
export default Random
```

Lorsque l'admin ajoute un nouveau étudiant ou un responsable de filière, ce dernier va recevoir automatiquement un email contenant ses informations de connexion qui doivent être modifiée au moment d'une première connexion pour plus de sécurité.

Cet email sera générer par le package de ReactJS : EmailJS



Le code utiliser est :

```
if(res.data.status === 200) {
    emailjs.sendForm('service_wusyzdo', 'template.sexuqy9', e.target, 'hFVxbWiMX6RCjK4zS')
    .then((result) => {
        console.log(result)
        navigate("/admin/view_respo")
        swal("Success", res.data.message, "success")
        setError_list('/')
    }), (error) => {
        console.log(error.text);
        swal("Success", "responsable added succesfully but email did'nt", "success")
        setError_list('/')
        navigate("/admin/view_respo")
    });
}
```

Exemple d'un mail envoyé après l'ajout d'un responsable de filière:

11:16 14%

✉ ⋮

Creation de votre compte
professionnelle responsable
filiere External Bin

 abderrahim/maysaa 19 May ✉ ⋮
to me ▾

Merci de trouver ci-joint les données permettant
l'accès à notre plateforme Gin

Login: maysaebcxb
Mot de passe: kRSNUUVU7

NB: Merci de réinitialiser votre mot de passe

Email sent via [EmailJS.com](#)

3-Génération d'un code QR pour entrer dans le site :

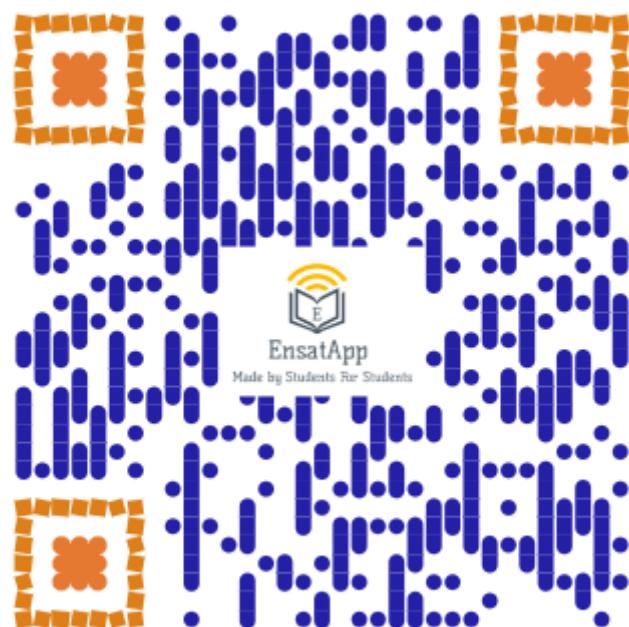
A l'aide du site : <https://www.logomaker.com>

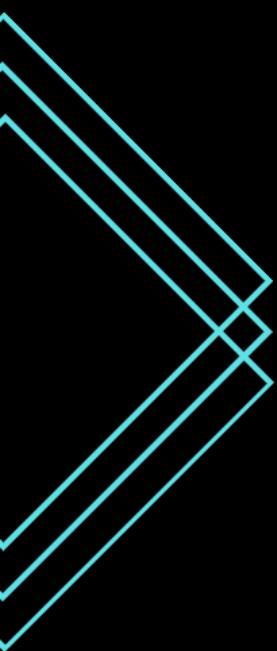
Nous avons pu générer un Logo de l'application.

Et en utilisant le site : [QRCode Monkey - The free QR Code Generator to create custom QR Codes with Logo \(qr-code-monkey.com\)](http://qr-code-monkey.com)

Nous avons générer un code QR avec un design unique pour permettre aux utilisateurs d'avoir l'URL de notre site en scannant le code.

Le résultat final :





Chapitre 9

Déploiement de l'application
sur un serveur réel et
configuration d'un nom de
domaine

1-Creation d'un sever Ubuntu via DigitalOceans



DigitalOcean est un fournisseur d'hébergement cloud qui propose des services de cloud computing et d'infrastructure en tant que service (IaaS). Reconnues pour leur tarification et leur évolutivité, les équipes peuvent se déployer sur DigitalOcean en quelques secondes avec des forfaits à partir de seulement 5 \$ par mois. Cette structure peut aider n'importe qui à se lancer rapidement dans le cloud, il permet de:

- Hébergez des sites Web avec la plate-forme d'hébergement cloud la plus simple.
- Créer des applications Web ou des backends d'API sur une infrastructure robuste.
- Déployer des applications basées sur des conteneurs avec Kubernetes géré
- Accélérer le développement avec une API intuitive, des outils de développement et des modules complémentaires CI/CD .
- Apprendre les bases du cloud computing.

Nous avons commencer par créer un compte payant sur DigitalOceon pour obtenir notre propre serveur en payant 10\$.

Search by resource name or public IP (Ctrl+B)

Create ? My Team Estimated costs: \$0.00

ProjectAbdo
Class project / Educational purposes

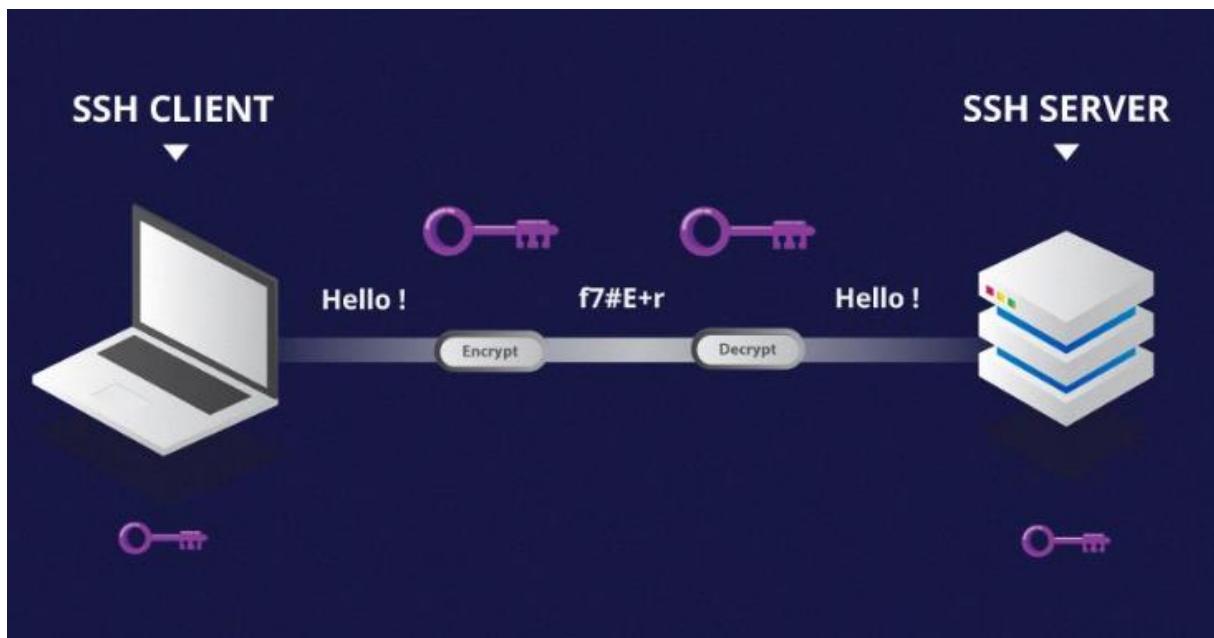
Resources Activity Settings

DROPLETS (1)

	abdo	+	+	...
Image	Ubuntu 20.04 (LTS) x64	Region	AMS3	
Size	1 vCPU 2GB / 25GB Disk (\$10/mo) Resize	IPv4	128.199.51.117	
		IPv6	Enable	
		Private IP	10.110.0.3	
		VPC	default-ams3	

2-Creation d'une clé ssh dans la machine locale pour accéder au serveur via le protocole ssh

SSH est un protocole qui facilite les connexions sécurisées entre deux systèmes à l'aide d'une architecture client/serveur et permet aux utilisateurs de se connecter à distance à des systèmes hôte de serveurs.



Nous avons générée une clé SSH privée pour assurer la connexion entre le serveur et la machine locale de la manière suivante :

```
abderrahimfizazi@abderrahimfizazi:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/abderrahimfizazi/.ssh/id_rsa): app
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in app
Your public key has been saved in app.pub
The key fingerprint is:
SHA256:VPogGI+9kGgZzAaEupAbQdBDjL9uNfYfYVbDGCWJqxs abderrahimfizazi@abderrahimfizazi
The key's randomart image is:
+---[RSA 3072]---+
|B@o ..oB+ . |
|+ B+.B*..+ |
|.=+=+++.+ |
|=... o..+ o |
|.+ + ...S . |
|o E . |
|+ + |
|+ |
|+ |
+---[SHA256]---+
abderrahimfizazi@abderrahimfizazi:~$
```

3-Connexion du serveur via la machine locale :

Après le partage de la clé public dans notre compte sur DigitalOceans, nous avons pu se connecter à notre serveur, de la manière suivante :

```
abderrahimfizazi@abderrahimfizazi:~$ ssh root@128.199.51.117
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-117-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Jun 13 10:40:19 UTC 2022

System load:  0.0                  Users logged in:      0
Usage of /:   24.1% of 24.06GB   IPv4 address for eth0: 128.199.51.117
Memory usage: 37%                IPv4 address for eth0: 10.18.0.6
Swap usage:   0%                 IPv4 address for eth1: 10.110.0.3
Processes:    119

0 updates can be applied immediately.

Last login: Sun Jun 12 21:33:56 2022 from 162.243.190.66
root@abdo:~#
```

Et nous avons envoyé le projet vers le serveur pour l'héberger par la commande :

```
Scp -i .ssh/key -r (chemin d'accès au projet sur la machine locale)  
(domain ) : (chemin d'accès sur le serveur)
```

NB :Nous devons refaire toutes les installations et configurations citées dans le premier chapitre au niveau du serveur pour commencer faire fonctionner l'application depuis le serveur.(Nous avons profiter des connaissances que nous avons acquis au début dans le cadre des installations effectuées au début du projet pour optimiser le temps et mieux gérer notre travail).

4-Configuration DNS

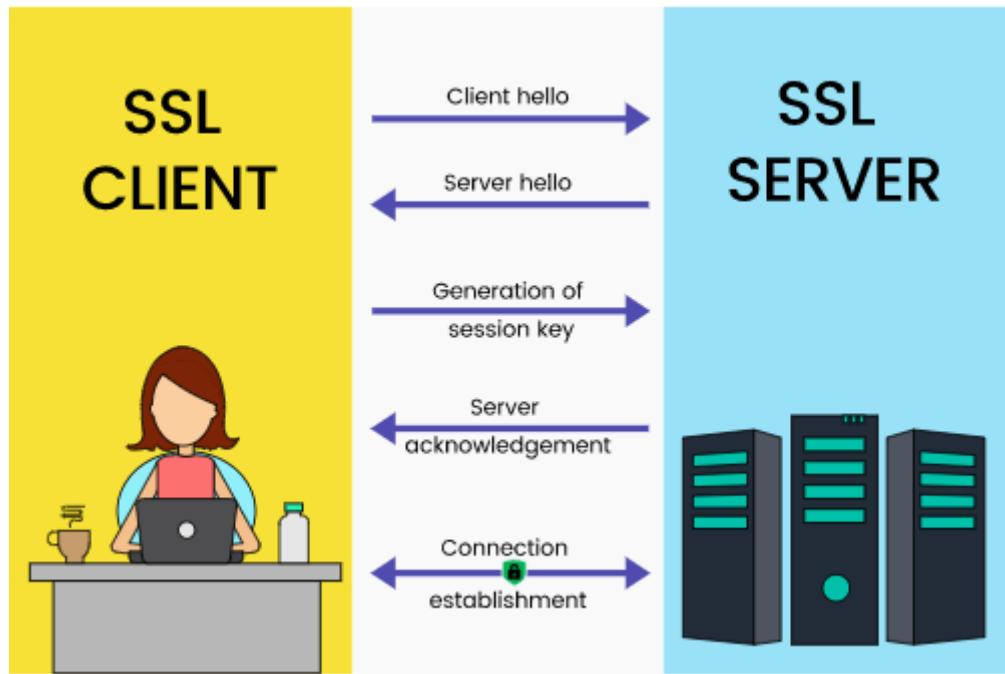
Nous avons créer un compte sur dot.tech afin de générer un nom de sous-domaine, plus précisément le nom '.tech' qui est un domaine générique de premier niveau du système de noms de domaine utilisé sur Internet. Le nom est tronqué de la technologie.



Nous avons générer le nom de domaine « <https://ensatapp.fazazi.tech> »

5-Installation de SSL (HTTPS) :

Afin de garantir plus de sécurité pour notre application nous avons exploiter le protocole https en installant SSL sur notre serveur :



```

root@abdo:/home/sadproject# sudo certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): Abderrahimfizazi2019@gmail.com

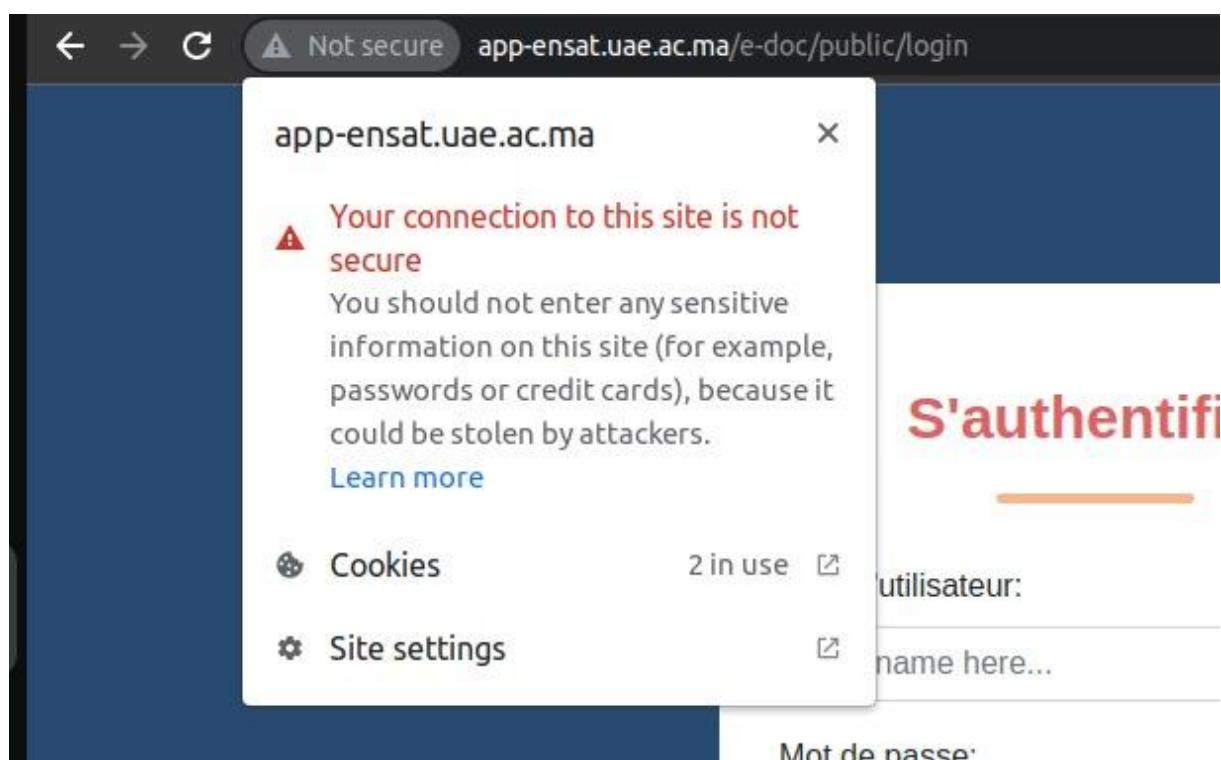
-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: A

-----
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: Y

Which names would you like to activate HTTPS for?
-----
1: ensatapp.fazazi.tech
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 1
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for ensatapp.fazazi.tech
Enabled Apache rewrite module
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at /etc/apache2/sites-available/ensatapp.fazazi.tech-le-ssl.conf
Enabled Apache socache_shmcb module
Enabled Apache ssl module
Deploying Certificate to VirtualHost /etc/apache2/sites-available/ensatapp.fazazi.tech-le-ssl.conf
Enabling available site: /etc/apache2/sites-available/ensatapp.fazazi.tech-le-ssl.conf

```

Comme résultat, la connexion à l'application est bien sécurisée au contraire de celle du site de l'ENSAT qui contient nos informations personnelles et professionnelles et malgré ça, il n'est pas bien sécurisé :



Conclusion

A travers ce projet, nous avons pu exploiter les différentes phases de la constitution d'une application web sécurisée, interactive, responsive et capable de permettre au cadre administrative de l'ENSAT de gérer ses étudiants ainsi que l'échange des emails entre professeurs et étudiants. Nous avons conçu trois espaces selon les privilèges de chaque utilisateur, ce qui nous a permis de développer plusieurs compétences techniques à travers la manipulation et la configuration de différents serveurs, l'assurance d'une sécurité réseau, le travail sous une architecture MVC grâce au Framework Laravel, la programmation sous différents langages de programmation, suivant les finalités recherchées, telles que : PHP, PLSQL, JavaScript... Et également, nous avons pu développer notre esprit d'équipe, d'engagement et de responsabilité envers le cahier de charge qui nous a été communiqué, nous avons réalisé la majorité des tâches demandées ainsi que l'ajout de plusieurs fonctionnalités.

Sincèrement nous avons bien aimé l'idée de travail sur un tel projet : encadré et noté dans un cadre professionnel, ce qui nous a motivé à faire de notre mieux. Nous aimerions bien que cela reste maintenu pour les années futures et nous espérons que les autres responsables de matières suivent la même approche au lieu de l'approche classique, vu l'importance de la réalisation de tels projets dans notre parcours entant que futurs ingénieurs.

Planning des tâches

Tâche	Membre responsable	Date de finalisation
Installations Linux	Abderrahim Fazazi et Maysae Hmamouchi	12/05/2022
Configuration des serveurs	Maysae Hmamouchi	07/06/2022
Tests de connexion Oracle	Maysae Hmamouchi	15/05/2022
Réalisation des contrôleurs et procédures nécessaires	Abderrahim Fazazi et Maysae Hamamouchi	08/05/2022
Réalisation des Triggers	Abderrahim Fazazi et Maysae Hmamouchi	07/06/2022
Réalisation des observers et migrations	Abderrahim Fazazi	07/05/2022
Configuration DNS	Abderrahim Fazazi	11/06/2022
Echange des emails avec les fonctionnalités de sécurité	Abderrahim Fazazi	10/06/2022
Création d'un compte sur DigitalOcean et hébergement du site	Abderrahim Fazazi	12/06/2022
Comparaisons entre les serveurs	Maysae Hmamouchi	23/05/2022
Rédaction du rapport	Maysae Hmamouchi	13/06/2022
Réalisation d'une vidéo représentative	Abderrahim Fazazi et Maysae Hmamouchi	13/06/2022
Réalisation des ptts	Maysae Hmamouchi et Abderrahim Fazazi	14/06/2022
Design du Logo et Code QR	Maysae Hmamouchi	13/06/2022

Références

Api avec Laravel [Cours Laravel 8 – les données – les ressources d'API – Laravel \(sillo.org\)](#)

Architecture PostFix : [File:Postfix architecture.svg - Wikimedia Commons](#)

Architecture Postman : [Microservices Example: How Postman Engineering Does Microservices | Postman Blog](#)

Architecture SendMail : [Sendmail X: Architecture](#)

Architecture React : [React JS— Architecture tutorial + Features + Folder structure + Design Pattern + boilerplate code repo | Geek Culture \(medium.com\)](#)

Comparaison Postfix et SendMail : [Postfix vs. Sendmail \(linuxhint.com\)](#)

Comparaison Apache et Nginx: [Comparatif Apache vs Nginx 2020 Points Forts et Faibles. Lequel choisir \(tophebergeur.com\)](#)

Configuration Laravel avec Nginx : [Deployment - Laravel - The PHP Framework For Web Artisans](#)

Dot.tech :_.TECH domains | Find the perfect domain name for your tech company on .TECH (get.tech)

Installer et configurer Nginx :

<https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04-fr>

Installation et configuration de Postfix <https://guide.ubuntu-fr.org/server/postfix.html>

Modèle MVC :

<https://www.youtube.com/watch?v=9LqxOEI2YAQ&list=PLIeeyAQEJWw5ANlav0W01A60sV7xF2QjN>

NodeJs : Qu'est-ce que Node.js et pourquoi l'utiliser ? (kinsta.com)

Npm : npm (npmjs.com)

Postman : Postman API Platform

Sendmail: [Sendmail. Histoire, Sécurité, X Sendmail, Sendmail-8 Rejets \(boowiki.info\)](#)

SSh : -<https://www.hostinger.fr/tutoriels/ssh-linux>

Trouver l'adresse ip privée : <http://icanhazip.com/>