2024

AP4 GESTION DE STOCK

APPLICATION WEB DE GESTION DE STOCK POUR UN LABORATOIRE DANS LE CONTEXTE GSB

Richer Marine
BTS SIO

SOMMAIRE

01 - INTRODUCTION

Le laboratoire Galaxy Swiss Bourdin (GSB) est une entreprise pharmaceutique résultant de la fusion entre le géant américain Galaxy et le conglomérat européen Swiss Bourdin. Cette fusion a créé un leader dans le secteur pharmaceutique, spécialisé notamment dans les maladies virales telles que le SIDA et les hépatites. Le siège administratif de l'entité Galaxy Swiss Bourdin Europe est situé à Paris, en France, tandis que le siège social de la multinationale se trouve à Philadelphie, en Pennsylvanie, aux États-Unis.

Suite à la fusion, l'entreprise a entrepris une réorganisation interne, visant à optimiser son activité en réalisant des économies d'échelle dans la production et la distribution des médicaments. Cela a entraîné une restructuration et une vague de licenciements. GSB compte actuellement 480 visiteurs médicaux en France métropolitaine et 60 dans les départements et territoires d'outre-mer, répartis en 6 secteurs géographiques.

Le système informatique de GSB est bien développé et prend en charge différentes fonctions de l'entreprise. Les fonctions administratives, telles que la gestion des ressources humaines et la comptabilité, sont présentes sur le site parisien. Il y a également un service de recherche en laboratoire, un service juridique et un service communication. La salle serveur, située au 6ème étage du bâtiment, est sécurisée et restreinte aux accès. Les serveurs assurent les fonctions de base du réseau ainsi que les communications internes.

Le réseau informatique est segmenté en VLAN pour fluidifier le trafic. Les données de l'entreprise sont considérées comme stratégiques et sont répliquées quotidiennement aux États-Unis pour assurer une sauvegarde et une redondance maximales. La Direction des Services Informatiques (DSI) joue un rôle important dans la gestion informatique de GSB et participe aux choix stratégiques de l'entreprise.

01 - INTRODUCTION

Sur le site, chaque employé dispose d'un poste fixe relié au système central, et il y a également des stations de travail plus puissantes dans la partie labo-recherche. De plus, de nombreux ordinateurs portables sont utilisés par le personnel de direction, le service informatique et les services commerciaux. Les visiteurs médicaux reçoivent une indemnité ou une dotation en équipement informatique, selon les politiques de Swiss-Bourdin et Galaxy.

Chaque employé de l'entreprise a une adresse de messagerie de la forme "nomUtilisateur@swiss-galaxy.com". Les anciennes adresses des laboratoires fusionnés ont été définitivement fermées au 1er janvier 2011.

Responsabilité :

Le laboratoire désire mettre à disposition une application mobile leur permettant de suivre leur traitement médicamenteux. Les utilisateurs peuvent programmer des rappels pour prendre leur médicament à temps et consulter leur historique médical.

Les équipes du service développement auront notamment à produire puis à fournir les éléments applicatifs permettant :

Ici, nous allons créer une application web de gestion de stock pour les employés, on y retrouve également les commandes auprès des fournisseurs

Lien github du projet :

02. CAHIER DES CHARGES

GESTION DE STOCK

Un employé peut réserver des produits comme : substances actives, médicaments et le matériel. Il y a également l'affichage de ce stock et la quantité présente.

AUTHENTIFICATION

Un employé peut s'authentifier et créer un compte afin d'accéder aux différentes parties qui se différencient selon les rôles.

COMMANDES

Un employé ayant un rôle qui lui permet de le faire, peut commander des produits auprès des fournisseurs.

BASE DE DONNEE

La base de données contient une liste importante de produits qui l'on retrouve en libre accès.

03. ENVIRONNEMENT

Docker

Docker permet de créer des containers, dans lequel je retrouve apache et ma base SQL.

MySQL

Base de donnée SQL.

Apache

Apache permet d'accéder à mon site en localhost.

PhpStorm

Il s'agit d'un éditeur de texte PHP.

INSTALLATION

Docker permet de créer des containers et de ne pas avoir à faire de configuration, ce qui rend le projet plus flexible.
Afin d'installer docker, il faut se rendre sur docker avec le lien ci-dessous.

L'éditeur de texte utilise ici demande une licence, alors on peut choisir Visual studio code qui est gratuit.

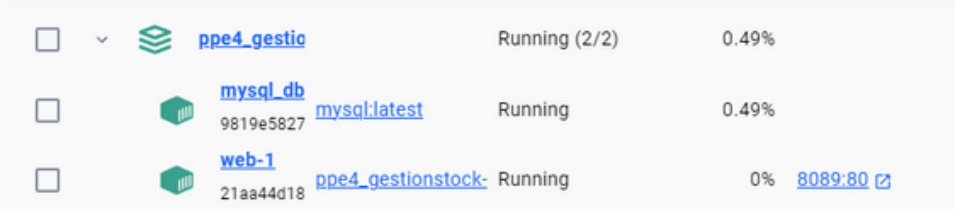
https://code.visualstudio.com/download

MISE EN ROUTE DU PROJET

Afin de lancer le projet, on a 2 fichiers docker : docker-compose.yaml et Dockerfile.
Le premier créé des container qui vont contenir apache et mysql.
L'autre créé un container pour avoir php et installer les dépendances requises.

```
FROM PHP:8.3.0-APACHE

RUN DOCKER-PHP-EXT-INSTALL PDO PDO_MYSQL
```

Si Apache ne fonctionne pas, dans docker, aller dans web-1, puis files>etc/apache2/apache2.conf



/etc/apache2/apache2.conf

```
162     Require all denied
163 </Directory>
164
165 <Directory /usr/share>
166     AllowOverride None
167     Require all granted
168 </Directory>
169
170 <Directory /var/www/>
171     Options Indexes FollowSymLinks
172     AllowOverride None
173     Require all granted
174 </Directory>
175
```

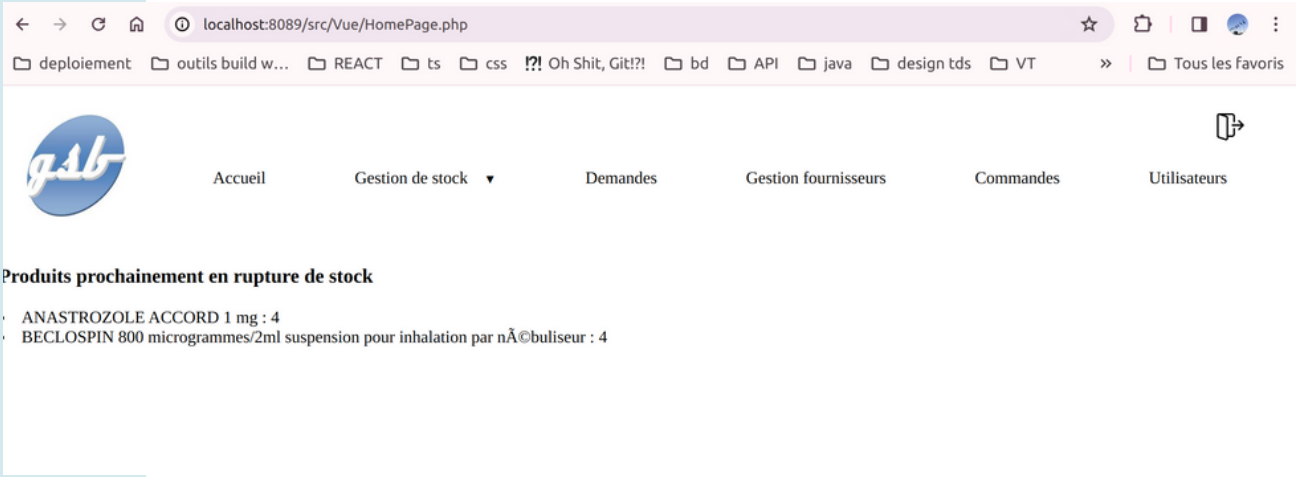
Changer AllowOverride None par AllowOveride All dans le partie surlignée.

MISE EN ROUTE DU PROJET

Pour lancer le projet, on a docker, vs code, et nos deux fichiers à la racine du projet. Maintenant, dans un terminal, il faut se déplacer dans le dossier PPE4_GestionStock dans lequel on lance les commandes suivantes :

```
FROM PHP:8.3.0-APACHE

RUN DOCKER-PHP-EXT-INSTALL PDO PDO_MYSQL
```

On peut donc accéder au projet à l'adresse **http://localhost:8089/src/controller/index?page=login** avec docker et **www.marine.isitech.fr/src/controller/index?page=login** si vous avez accès au site déployer sans avoir de configuration à faire.



COMPTES AVEC DIFFÉRENTS RÔLES

On retrouve 3 types de rôles : Admin, SuperUser et User.

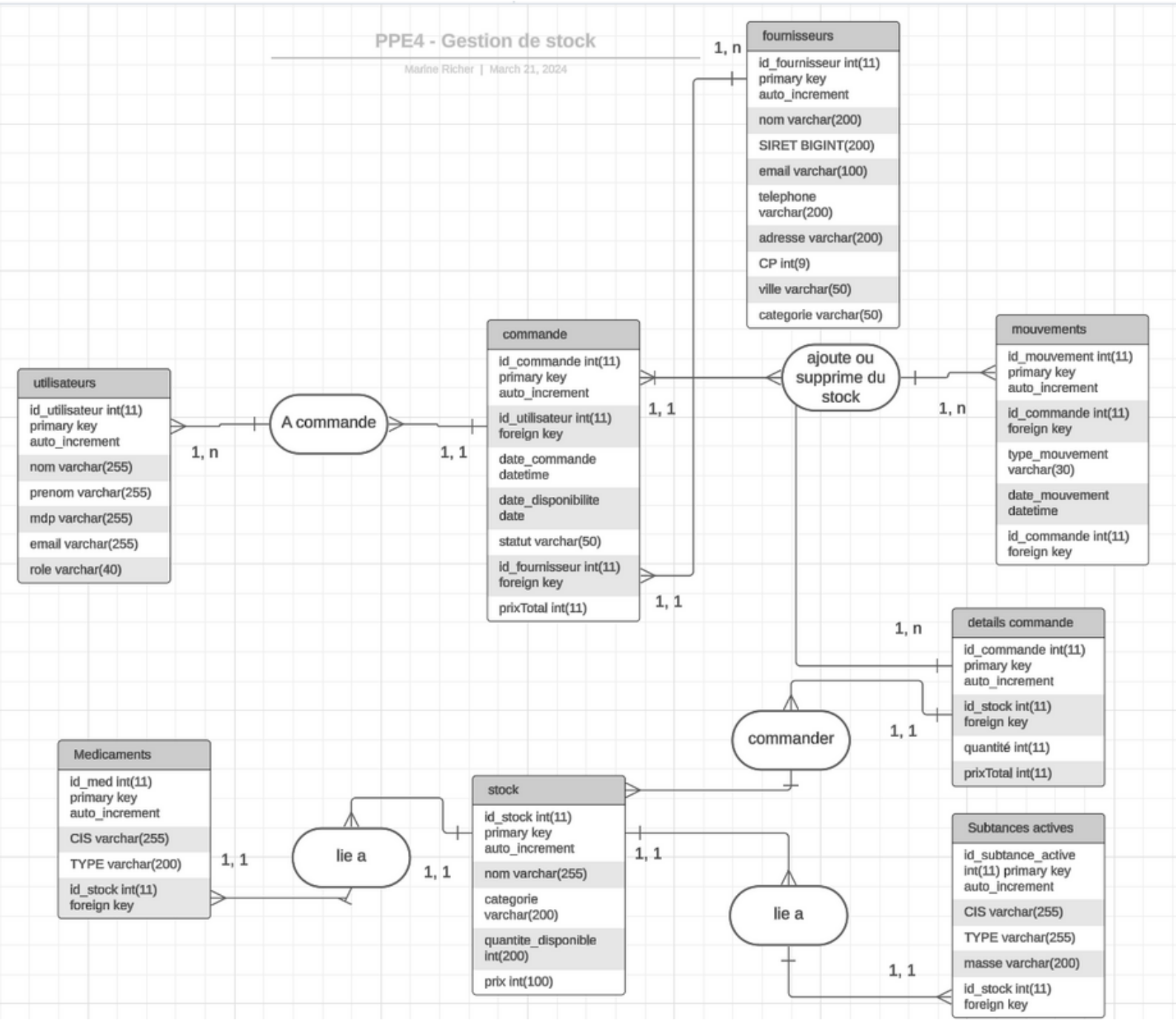
Admin
Email : admin@gmail.com
Mot de passe : P4~Y/3Rp2d$k

SuperUser
Email : SuperUser@gmail.com
Mot de passe : U(9qU7)F7gf~

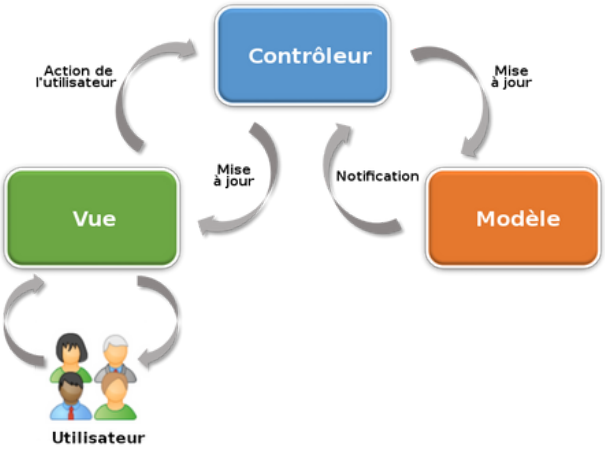
User
Email : User@gmail.com
Mot de passe : %[4_wPyWKt89

04. BASE DE DONNÉE

La base de données est en SQL.
Les données des médicaments et substances actives proviennent du site du gouvernement qui les met en libre-service :
https://base-donnees-publique.medicaments.gouv.fr/
Le schéma MCD est ci-dessous.



05. LES PAGES ET FONCTIONNALITÉS



L'architecture de ce projet est le model MVC, dans lequel on retrouve le visuel, le controller qui gère les fonctions et les models qui prennent en charge les requêtes.

INSCRIPTION

Dans la table utilisateur, j'ai besoin des données que l'on retrouve sur le formulaire d'inscription : prénom, nom, email, mot de passe.

La balise permet de récupérer les champs depuis la variable global $_POST[] dans controller.
Ensuite, après avoir récupéré les données on hash le mot de passe avec la fonction password_hash().
Afin de vérifier que l'utilisateur n'existe pas, on utilise une fonction verify_user() du model User pour vérifier que l'utilisateur n'existe pas en base puis on l'insère en base.

Le fichier RegisterController.php

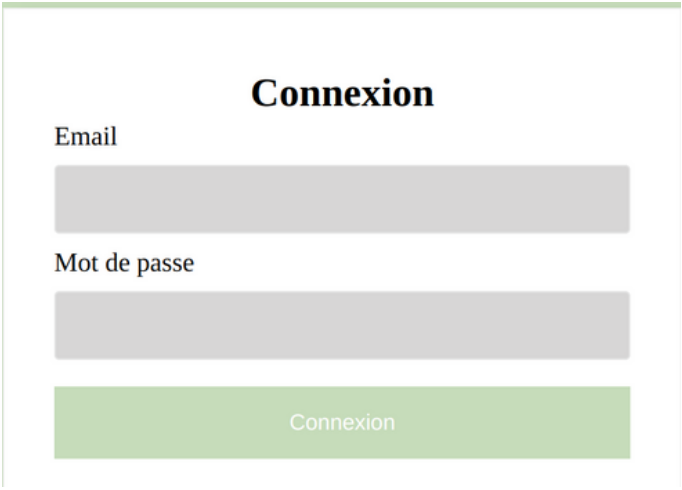
```
$user_data = $this->model('Users');

$user_data->registreUtilisateur($nom, $prenom, $email, $motDePasse);

$user_data->AddUser();
```

Le fichier Users.php

```
$SQL = "INSERT INTO UTILISATEURS (NOM, PRENOM, EMAIL, MOT_DE_PASSE) VALUES (:NOM,
:PRENOM, :EMAIL, :PASSWORD)";
      $STMT = $THIS->PDO->PREPARE($SQL);

      IF ($STMT->EXECUTE([
         ':NOM' => $THIS->NOM,
         ':PRENOM' => $THIS->PRENOM,
         ':EMAIL' => $THIS->EMAIL,
         ':PASSWORD' => $THIS->MOTDEPASSE
      ]))
```

CONNEXION

Afin de se connecter, il faut un email et un mot de passe. On récupère les données avec la balise également, on va également utiliser verify_user() qui nous renvoie ses données de connexion.
On y vérifie le mot de passe avec password_verify(), il va prendre en compte que le mot de passe en base est haché et le comparer avec celui renseigné.
Un nombre de tentatives de connexion est compté, à partir de la 5e, le compte ne peux plus se connecter et doit être débloqué par un administrateur.



Le fichier LoginController.php

```
$USER_DATA = $THIS->MODEL('USERS');
        $USER_DATA = $USER_DATA->VERIFY_USER($EMAIL);

         IF (!EMPTY($USER_DATA)) {
            IF (PASSWORD_VERIFY($PASSWORD, $USER_DATA['MOT_DE_PASSE'])) {

$USER_MODEL->MISEAJOURTENTATIVE($USER_DATA['ID_UTILISATEUR']);
$JWT = NEW \CONTROLLER\JWT();
$PAYLOAD = $JWT->GENERER_PAYLOAD($USER_DATA['ID_UTILISATEUR'], $USER_DATA['EMAIL'],
$USER_DATA['ROLE']);
SETCOOKIE("JWT", $JWT->GENERER_JWT($PAYLOAD), TIME() + 14400);
HEADER("LOCATION: ../VUE/HOMEPAGE.PHP");
                HEADER("LOCATION: ../VUE/HOMEPAGE.PHP");
```

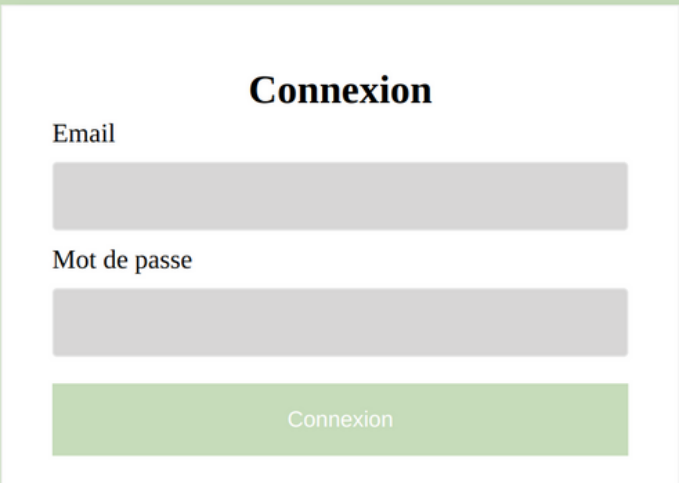
Le fichier Users.php

```
PUBLIC FUNCTION VERIFY_USER(STRING $EMAIL)
 {
    $SQL = "SELECT ID_UTILISATEUR, EMAIL, MOT_DE_PASSE, ROLE FROM UTILISATEURS WHERE EMAIL = :EMAIL"
    $STMT = $THIS->PDO->PREPARE($SQL); // UTILISE LA PROPRIÉTÉ $PDO
    $STMT->EXECUTE(['EMAIL' => $EMAIL]);
    $USER = $STMT->FETCH(PDO::FETCH_ASSOC); }
```

CONNEXION -CLÉ TOKEN

La clé token permet de garder les informations sur l'utilisateur connecté en sécurité.
On a un contrôler JWT, qui va générer la clé et permettre également de décrypter les informations.

Afin de créer les cookie, lors de la connexion, il se créait avec JWT.

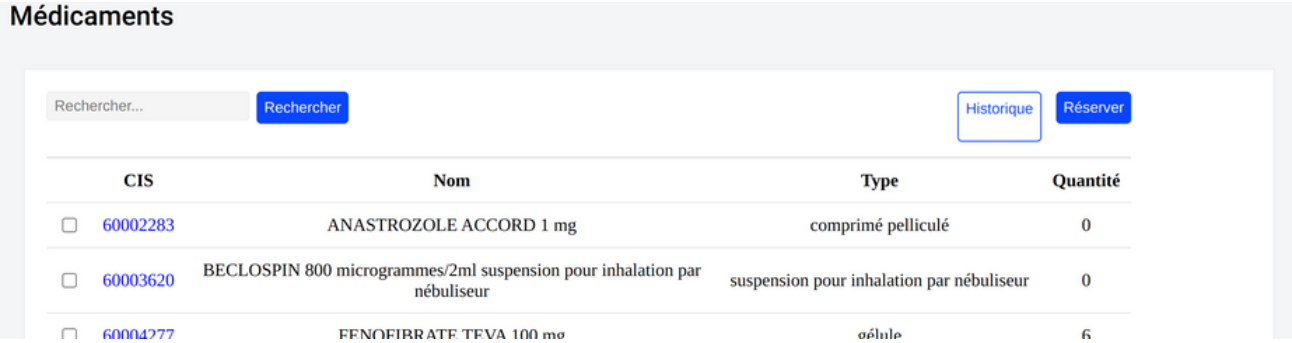



```
$PAYLOAD = $JWT->GENERER_PAYLOAD($USER_DATA['ID_UTILISATEUR'],
$USER_DATA['EMAIL'], $USER_DATA['ROLE']);
 SETCOOKIE("JWT", $JWT->GENERER_JWT($PAYLOAD), TIME() + 14400);
```

Generer_payload() permet de réunir les informations que l'on va récupérer.
Ensuite, Générer_JWT() va encoder les données à partir d'une clé JWT secret.

```
PUBLIC FUNCTION GENERER_PAYLOAD(
        STRING $ID,
        STRING $EMAIL,
        STRING $ROLE
    ): ARRAY {
        RETURN [
            "USER_ID" => $ID,
            "USER_EMAIL" => $EMAIL,
            "USER_ROLE" => $ROLE,
        ];
    }
```

AFFICHAGE DE STOCK MÉDICAMENT/ SUBSTANCES ACTIVES /MATÉRIEL



Les différents type de stock que l'on retrouve sont regroupes, car il s'agit du même fonctionnement pour l'affichage de ses différentes données.

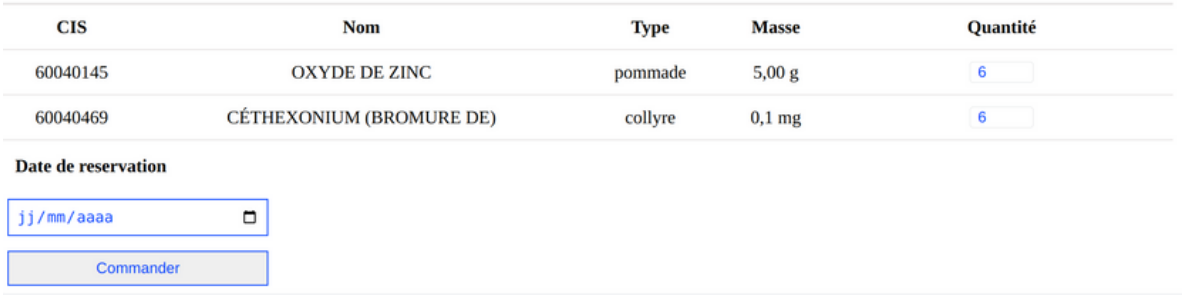
La requête pour afficher les médicaments et substances actives contiennent une jointure entre 2 tables, car elles contiennent des données en plus dans une autre table : le code CIS, le type et la masse en plus pour les substances actives.

```
$SQL= "SELECT CIS, S.NOM, TYPE, MASSE, S.QUANTITE_DISPONIBLE FROM GSB.STOCK AS S JOIN GSB.SUBTANCE_ACTIVE AS A ON A. ID_STOCK = S.ID_STOCK WHERE CATEGORIE='SUBTANCE ACTIVE' ";
```

Dans la page d'affichage, j'appelle la fonction du controller qui va permettre d'appeler la requête et de récupérer les données. Ensuite une condition foreach() permet de traiter chaque donnée indépendamment dans la table qui s'affiche.

```
REQUIRE_ONCE '../CONTROLLER/SUBTANCEACTIVECONTROLLER.PHP';
INCLUDE_ONCE '../MODEL/MATERIEL.PHP';

$SUBTANCEACTIVE = NEW \CONTROLLER\SUBTANCEACTIVECONTROLLER();

$DATASUBTANCEACTIVE = $SUBTANCEACTIVE->SELECTSUBTANCEACTIVE();

FOREACH ($DATASUBTANCEACTIVE AS $ITEM) {

            ECHO "<TR>
            <TD><INPUT TYPE='CHECKBOX' NAME='SUBTANCEACTIVESELECTIONNE[]' VALUE='" . $ITEM['CIS'] . "'></TD>
            <TD><A HREF='HTTPS://BASE-DONNEES-PUBLIQUE.MEDICAMENTS.GOUV.FR/EXTRAIT.PHP?SPECID=".$ITEM['CIS'] . "'>" .$ITEM['CIS'] . "</A></TD>
            <TD>" . $ITEM['NOM']. "</TD>
            <TD>" .$ITEM['TYPE'] . "</TD>
             <TD>" .$ITEM['MASSE'] . "</TD>
            <TD>" .$ITEM['QUANTITE_DISPONIBLE'] . "</TD>
        </TR>";
        }
```

RESERVATION DE STOCK MÉDICAMENT/ SUBSTANCES ACTIVES /MATÉRIEL



CIS	Nom	Type	Masse	Quantité
60040145	OXYDE DE ZINC	pommade	5,00 g	6
60040469	CÉTHEXONIUM (BROMURE DE)	collyre	0,1 mg	6

Date de reservation

jj/mm/aaaa

Commander

Après avoir sélectionné les produits concerne dans les 'checkbox', on clique sur réserver.
On y retrouve les données sélectionnées avec la requête à partir des code CIS sélectionnés, pour information les code CIS sont des données uniques dans notre cas.
La quantité peut être modifiée afin de choisir la quantité souhaitée, puis une date de réservation est à indiquer afin que le valideur le prenne en compte en fonction de ses validations.

Lors de la commande, on l'insère dans la table commande qui place son statut 'en_attente'.
Afin d'ajouter cette demande, on doit insérer des données dans plusieurs tables : commandes et détails_commande.
Pour insérer dans commande, on a besoin de la date de résolution souhaitée et de l'id de l'utilisateur.

```
 $SQL = "INSERT INTO COMMANDES (ID_UTILISATEUR, DATE_DISPONIBILITE) VALUES
(:ID_UTILISATEUR, :DATE_DISPONIBILITE)";
      $STMT = $THIS->PDO->PREPARE($SQL);

      IF ($STMT->EXECUTE([
         ':ID_UTILISATEUR' => $_SESSION['ID_UTILISATEUR'],
         ':DATE_DISPONIBILITE' => $THIS->DATERESERVATION,

      ])) {
         $_SESSION['IDCOMMANDE'] = $THIS->PDO->LASTINSERTID();
```

On récupère l'id de la donnée insérer avec lstInsertId() afin de l'ajouter par la suite dans la table details_commande qui va ajouter pour chaque produit concerne son id, sa quantité demandée et l'id de la commande.

```
$SQL = "INSERT INTO DETAILS_COMMANDE (ID_COMMANDE, ID_STOCK, QUANTITE) VALUES
(:LASTINSERTID, :IDSTOCK, :QUANTITE)";
      $STMT = $THIS->PDO->PREPARE($SQL);

      IF ($STMT->EXECUTE([
         ':LASTINSERTID' => $_SESSION['IDCOMMANDE'],
         ':IDSTOCK' => $IDSTOCK,
         ':QUANTITE' => $THIS->QUANTITE,

      ]))
```

DEMANDES DE RESERVATION

	Date de de la demande	Numéro de la demande	Statut de la demande	Date de disponibilité souhaitée	Catégorie
☐	2024-03-19 12:48:17	1	validee	2024-03-22	medicament
☐	2024-03-19 12:49:28	2	validee	2024-03-23	Materiel

Les demandes faites sont affichées dans Demandes, en fonction du rôle, on a un bouton ‘changer de statut’ et le nom du demandeur apparaît.

Changer le statut

	Date de de la demande	Numéro de la demande	Statut de la demande	Date de disponibilité souhaitée	Catégorie	Nom du demandeur
☐	2024-03-19 12:48:17	1	validee	2024-03-22	medicament	RICHER
☐	2024-03-19 12:49:28	2	validee	2024-03-23	Materiel	RICHER

Deux requêtes différentes sont passées en fonction du rôle, dans le controller commandesController.
Pour le superUser, on regroupe toutes les demandes, mais pour les utilisateurs, on entre leur id en paramètre.

```
$SQL = "SELECT DISTINCT C.ID_COMMANDE, C.DATE_COMMANDE, C.STATUT,
C.DATE_DISPONIBILITE, CATEGORIE FROM GSB.COMMANDES AS C JOIN GSB.DETAILS_COMMANDE
AS D ON D.ID_COMMANDE = C.ID_COMMANDE JOIN GSB.STOCK AS S ON S.ID_STOCK = D.ID_STOCK
WHERE ID_UTILISATEUR = :ID AND ID_FOURNISSEUR IS NULL";
$STMT = $THIS->PDO->PREPARE($SQL);
$STMT->EXECUTE([':ID' => $_SESSION['ID_UTILISATEUR']]);
```

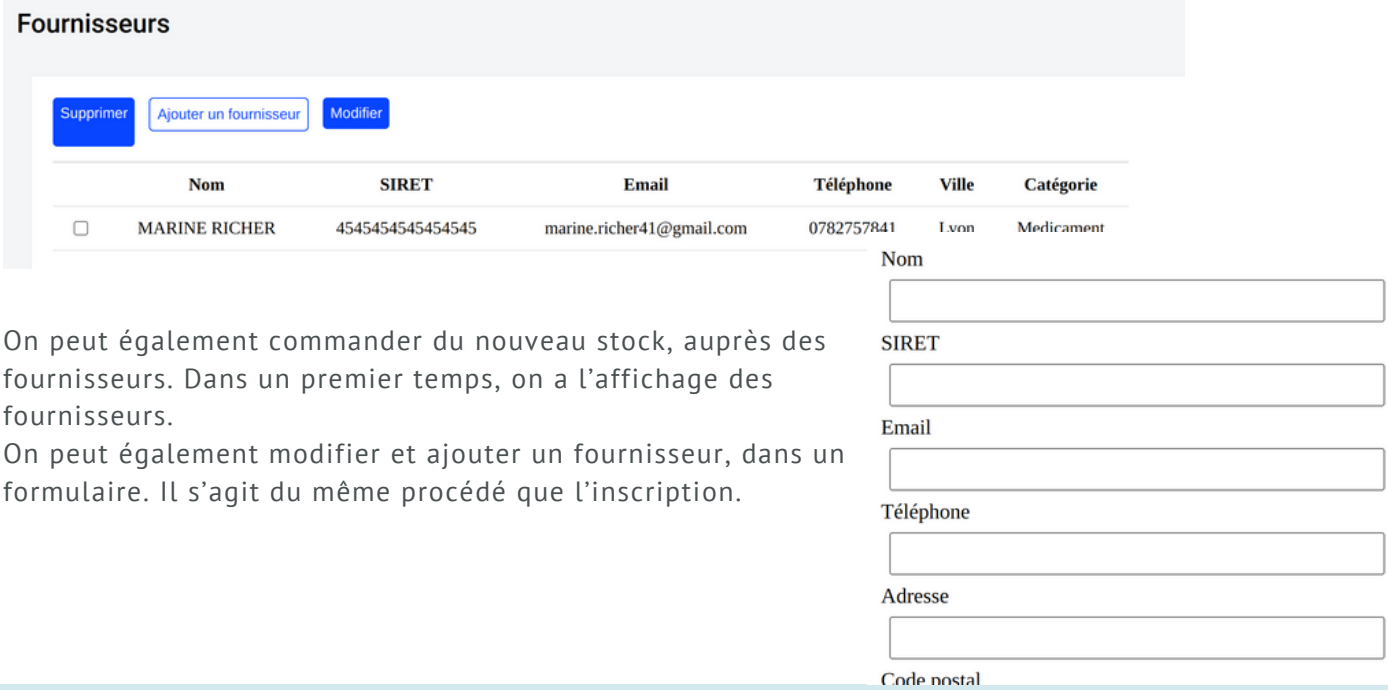
CHANGEMENT DE STATUT

Le statut peut changer, si le rôle superUser, indique ‘validee’, le nombre de quantité réservé associé au numéro de commandes et au produit concerné.
Le statut se change avec l’id de commande, puis on insère dans la table mouvements, l’id de commande, et les id stock associés, le mouvement de sortie et la quantité associée.
Puis la quantité est retirée dans la table stock.
Dans sortieStock, je sélectionne les quantités à retirer du stock dans details_commande.

```
IF ($STATE == 'VALIDEE') {

                $STOCKS = $MODELMOUVEMENT->SELECTIDSTOCK($IDCOM);

                FOREACH ($STOCKS AS $STOCK) {

                   $IDSTOCK = $STOCK['ID_STOCK'];
                   $MODELMOUVEMENT->INSERTMOUVEMENT($IDCOM, $IDSTOCK);
                   $MODELMOUVEMENT->MOUVEMENTSORTIE();
                   $MODELMOUVEMENT->SORTIESTOCK(); } }
```

```
 PUBLIC FUNCTION MOUVEMENTSORTIE(){

       $SQL = "INSERT INTO GSB.MOUVEMENTS (ID_STOCK, TYPE_MOUVEMENT,
ID_COMMANDE) VALUES (:ID_STOCK, 'SORTIE', :ID_COMMANDE)";
      .....
    }
PUBLIC FUNCTION SORTIESTOCK()
   {
      $SQLQUANTITE = 'SELECT QUANTITE FROM DETAILS_COMMANDE WHERE ID_COMMANDE =
:ID_COMMANDE AND ID_STOCK = :ID_STOCK';
      $STMTQUANTITE = $THIS->PDO->PREPARE($SQLQUANTITE);
      IF ($STMTQUANTITE->EXECUTE([':ID_COMMANDE' => $THIS->IDCOMMANDE, ':ID_STOCK' =>
$THIS->IDSTOCK])) {
         $DETAILS = $STMTQUANTITE->FETCHALL(PDO::FETCH_ASSOC);
         $DETAIL['QUANTITE']=[];
         FOREACH ($DETAILS AS $DETAIL) {
            $QUANTITE = $DETAIL['QUANTITE'];

            $SQLUPDATE = 'UPDATE GSB.STOCK SET QUANTITE_DISPONIBLE =
QUANTITE_DISPONIBLE  - :QUANTITE WHERE ID_STOCK = :ID_STOCK';
            $STMTUPDATE = $THIS->PDO->PREPARE($SQLUPDATE);
      ..... }
```

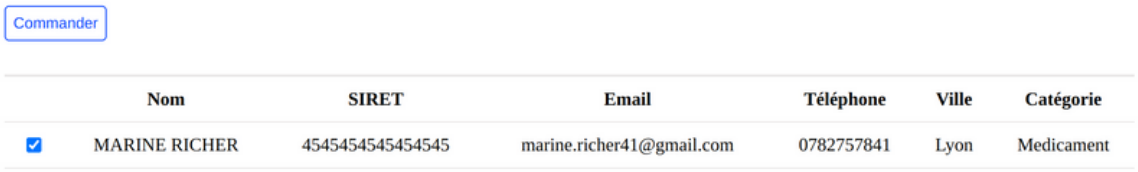
FOURNISSEURS



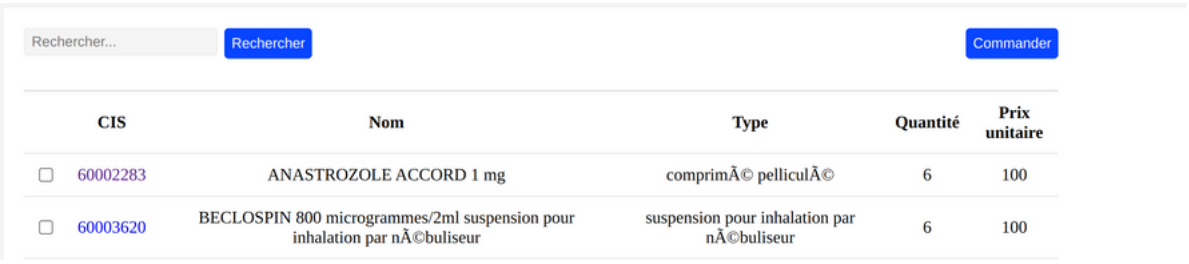
On peut également commander du nouveau stock, auprès des fournisseurs. Dans un premier temps, on a l'affichage des fournisseurs.
On peut également modifier et ajouter un fournisseur, dans un formulaire. Il s'agit du même procédé que l'inscription.

```
PUBLIC FUNCTION AFFICHAGEFOURNISSEURS()
   {
      $SQL = "SELECT ID_FOURNISSEUR, NOM, SIRET, EMAIL, TELEPHONE, VILLE,
CATEGORIE FROM GSB.FOURNISSEURS";
      $STMT = $THIS->PDO->PREPARE($SQL); // UTILISE LA PROPRIÉTÉ $PDO
      $STMT->EXECUTE();
      $FOURNISSEURS = $STMT->FETCHALL(PDO::FETCH_ASSOC);

      RETURN $FOURNISSEURS;
   }
```

COMMANDES DE STOCK



Avant de commander, il faut sélectionner le fournisseur, puis en fonction de la catégorie du fournisseur. Ici, on a des médicaments alors ce sera des médicaments qui seront affichés.



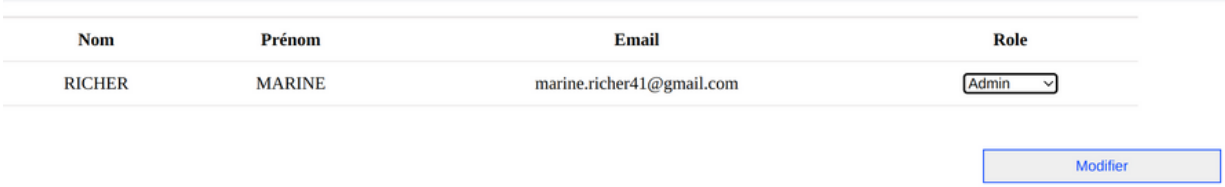
Après avoir sélectionné les produits, on va être redirigé vers le formulaire de commande.

Il s'agit de la même chose que le formulaire de réservation.

En changeant le statut, s'il est reçu, en cliquant sur modifier les médicaments, sont ajoutés à la base.

CHANGEMENTS DE RÔLES

On accède à une liste d'utilisateurs inscrits dans 'GestionUtilisateurs.php', dans lequel on sélectionne le ou les utilisateurs auxquels on veut changer le rôle.



On accède au changement de rôle, dans une liste de déroulante avec 3 rôles possibles : Admin, SuperUser, User.
Les rôles peuvent être modifié avec l'id_utilisateur.

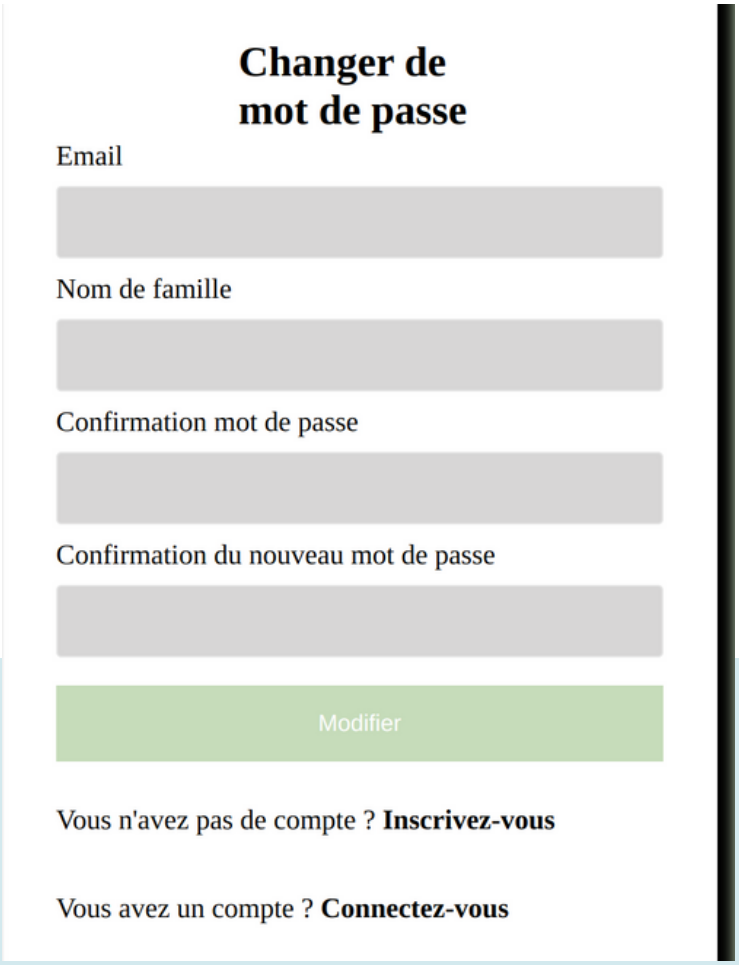
```
$COMBINEDARRAY = ARRAY_COMBINE($ID, $ROLE);
        FOREACH ($COMBINEDARRAY AS $ID_UTILISATEUR => $ROLE_UTILISATEUR) {
            $MODELCOMMANDE->UPDATEROLE($ROLE_UTILISATEUR);
            $ERRORDETAIL = $MODELCOMMANDE->MISEAJOUTROLE($ID_UTILISATEUR);
        }
```

array_combine() permet de combiner deux tableaux et d'associer leur valeur a chacun.

CHANGEMENTS DE MOT DE PASSE

Pour changer de mot de passe, il faut avoir l'email et renseigner son nom puis le nouveau mot de passe.
Grace a ses données, je le modifie en prenant en compte qu'il faut le hacher.

Afin d'instaurer au minimum, 12 caractères, 1 majuscule, 1 chiffre et 1 caractère spécial, on utilise pregmatch(), afin de vérifier s'il répond aux règles.




```
PREG_MATCH("/^(?=.*[A-Z])
  (?=.*\D)(?=.*\W).{12,}$/",
    $_POST['PASSWORD']
```

```
PUBLIC FUNCTION MISEAJOURMDP()
   {
      $SQL = "UPDATE GSB.UTILISATEURS SET MOT_DE_PASSE=:MOT_DE_PASSE WHERE
EMAIL=:EMAIL AND NOM=:NOM";
      $STMT = $THIS->PDO->PREPARE($SQL); // UTILISE LA PROPRIÉTÉ $PDO

      IF($STMT->EXECUTE( [':MOT_DE_PASSE' => $THIS->MOTDEPASSE,
         ':EMAIL' => $THIS->EMAIL,
         ':NOM' => $THIS->NOM
      ])){
         RETURN "MOT DE PASSE MODIFIE AVEC SUCCES";
      }ELSE {
         RETURN 'ERREUR LORS DE LA MODIFICATION DE MOT DE PASSE';
      }

   }
```

ROUTES / ROLES

Pour les routes, on met en place 2 fichiers : "index.php" et ".htaccess",
Dans ".htaccess", il faut montrer de quelle manière créer l'url a apache2.
L'URL est sous la forme : index?page=NomPage

```
REWRITEENGINE ON
REWRITEBASE /
REWRITECOND %{REQUEST_FILENAME} !-F
REWRITECOND %{REQUEST_FILENAME} !-D
REWRITERULE ^(.*)$ /PUBLIC/INDEX.PHP?PAGE=$1 [QSA,L]
```

Ensuite, dans "index.php", on utilise switch pour choisir l'URL en fonction de l'action et ensuite, on ajuste l'affichage en fonction des rôles : User, SuperUser, Admin.

La différence d'action en fonction du rôle est :
Admin : il peut tout faire, y compris de la gestion des utilisateurs.
Le superUser : il peut commander du stock et valide les demandes de réservations.

- L'User peut réserver du stock.

```
SWITCH ($PAGE) {
    CASE 'LOGIN':
INCLUDE '../SRC/VUE/SIGNIN.PHP';
        BREAK;

    /...../

    CASE 'UTILISATEUR':
 IF($PAYLOAD['USER_ROLE'] == 'ADMIN') {
        INCLUDE
'../SRC/VUE/GESTIONUTILISATEURS.PHP';
        }ELSE{
INCLUDE '../SRC/VUE/SIGNIN.PHP';
        }
        BREAK;
```

06. DÉPLOIEMENT / INSTALLATION EN LOCAL

Afin de déployer nos projets, l'école a mis à disposition un serveur dans lequel on y a ajouté nos projets et créer un nom de domaine.

INSTALLATION DES PACKAGES

Avant de mettre à jour les packages, on met à jour les packages mis à disposition sur le serveur.

SUDO APT UPDATE && SUDO APT UPGRADE -Y

Installation de apache : le serveur web

SUDO APT INSTALL APACHE2 -Y

Ensuite on le redémarre et on vérifie qu'il fonctionne

SUDO SYSTEMCTL START APACHE2
SUDO SYSTEMCTL ENABLE APACHE2

```
slam@server:~$ sudo systemctl status apache2
[sudo] password for slam:
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
     Active: active (running) since Wed 2024-03-20 15:31:59 UTC; 20h ago
```

Installation de mysql : la base de donnée

SUDO APT INSTALL MYSQL-SERVER -Y
SUDO MYSQL_SECURE_INSTALLATION

Pour finir, on installe PHP et on redemarre Apache2:

SUDO APT INSTALL PHP LIBAPACHE2-MOD-PHP PHP-MYSQL -Y
SUDO SYSTEMCTL RESTART APACHE2

CRÉATION DE DOSSIER ET CONFIGURATION NOM DE DOMAINE

On crée un dossier dans '/var/www/', celui-ci va se nommer 'marine.isitech.fr

```
SUDO MKDIR -P /VAR/WWW/MARINE.ISITECH.FR/PUBLIC
```

Attribution des droits au dossier :

```
SUDO CHOWN -R $USER:$USER
/VAR/WWW/MARINE.ISITECH.FR/HTML/MARINE.ISITECH.FR/PUBLIC
```

Changement de droit sur les dossiers afin que apache puisse lire les fichiers

```
SUDO CHMOD -R 755 /VAR/WWW
```

Dans /var/www/marine.isitech.fr/public, on créé un fichier index.html avec la commande nano dans lequel on met un script comme <h2>Hello</h2> afin de tester sur le site web plus tard.
Puis on s'occupe de la configuration, on copie la configuration par défaut, dans un nouveau fichier pour y insérer la nouvelle configuration.

```
SUDO CP /ETC/APACHE2/SITES-AVAILABLE/000-DEFAULT.CONF /ETC/APACHE2/SITES-AVAILABLE/MARINE.ISITECH.CONF
```

```
<VIRTUALHOST *:80>
SERVERADMIN WEBMASTER@LOCALHOST
SERVERNAME MARINE.ISITECH.FR
SERVERALIAS WWW.MARINE.ISITECH.FR
DOCUMENTROOT /VAR/WWW/MARINE.ISITECH/PUBLIC
ERRORLOG ${APACHE_LOG_DIR}/ERROR.LOG
CUSTOMLOG ${APACHE_LOG_DIR}/ACCESS.LOG COMBINED
</VIRTUALHOST>
```

Ensuite, on change de fichier de configuration par défaut, pour que Apache se référence à celui que l'on vient de créer.

```
SUDO A2ENSITE MARINE.ISITECH.CONF
```

```
SUDO A2DISSITE 000-DEFAULT.CONF
```

ACTIVATION DE SSL

SSL permet d'effectuer un transfert sécurisé et de passer en https.
Afin de ne pas avoir d'erreur, il faut autoriser apache2 dans le pare feu

```
SUDO UFW ALLOW "APACHE FULL"
```

Installation SSl avec a2nmod

```
SUDO A2ENMOD SSL
```

On redémarre apache et on créer le certificat, pour lequel on nous demande différentes informations sur le domaine concerné.

```
SUDO SYSTEMCTL RESTART APACHE2

SUDO OPENSSL REQ -X509 -NODES -DAYS 365 -NEWKEY RSA:2048 -KEYOUT
/ETC/SSL/PRIVATE/APACHE-SELFSIGNED.KEY -OUT /ETC/SSL/CERTS/APACHE-
SELFSIGNED.CRT
```

Dans le fichier de configuration éditer précédemment on ajoute plusieurs configuration, comme la redirection des chemins :

```
REWRITEENGINE ON
 REWRITECOND %{SERVER_NAME} =MARINE.ISITECH.FR [OR]
 REWRITECOND %{SERVER_NAME} =WWW.MARINE.ISITECH.FR
 REWRITERULE ^ HTTPS://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=PERMANENT]
/....AUTRE CODE../
<VIRTUALHOST *:443>
   SERVERNAME YOUR_DOMAIN_OR_IP
   DOCUMENTROOT /VAR/WWW/YOUR_DOMAIN_OR_IP

   SSLENGINE ON
   SSLCERTIFICATEFILE /ETC/SSL/CERTS/APACHE-SELFSIGNED.CRT
   SSLCERTIFICATEKEYFILE /ETC/SSL/PRIVATE/APACHE-SELFSIGNED.KEY
</VIRTUALHOST>
```

Il faut installer le module de réécriture pour apache2 :

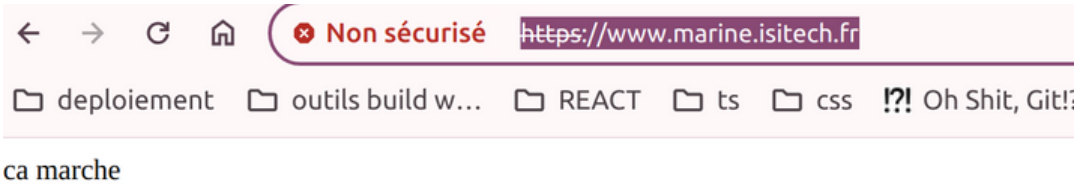
```
SUDO A2ENMOD REWRITE
```

ACTIVATION DE SSL

On peut tester notre fichier de configuration :

```
slam@server:~$ sudo apache2ctl configtest
[sudo] password for slam:
AH00558: apache2: Could not reliably determine the server's fully qualified doma
in name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress th
is message
Syntax OK
```

Si on a accès au réseau SLAM, en allant sur https://www.marine.isitech.fr/



07. CONCLUSION

L’application web est à destination des techniciens de laboratoires, les valideurs pour les mouvements en stock et des employés s’occupant des commandes auprès des fournisseurs.
L’application web peut avoir encore de multiples évolutions, telles que la sécurité, la rédaction des rapports.

Securite de l’application

- Sécurisation des transactions avec une vérification des données
- Vérification que l’utilisateur contient ses identifiants à chaque action
- Création de logs à chaque action d’un utilisateur

Rédaction des rapports

La rédaction des rapports après avoir fait des tests cliniques sur des médicaments, fabriquer des médicaments à partir de substance active.
Par conséquent, l’ajout des médicaments créé en stock.