

Contexte

Description du laboratoire GSB

Le secteur d'activité

L'industrie pharmaceutique est un secteur très lucratif dans lequel le mouvement de fusion acquisition est très forte. Les regroupements de laboratoires ces dernières années ont donné naissance à des entités gigantesques au seins desquelles le travail est longtemps resté organisé selon les anciennes structures.

De divers déboires récents autour de médicaments ou molécules ayant entraîné des complications médicales ont fait s'élever des voix contre une partie de l'activité des laboratoires : la visite médicale, réputée être le lieu d'arrangements entre l'industrie et les praticiens, et tout du moins un terrain d'influence opaque.

L'entreprise

Le laboratoire Galaxy Swiss Bourdin (GSB) est issu de la fusion entre le géant américain Galaxy (spécialisé dans le secteur des maladies virales dont le SIDA et les hépatites) et le conglomérat européen Swiss Bourdin (travaillant sur des médicaments plus conventionnels), lui-même déjà union de trois petits laboratoires. En 2009, les deux géants pharmaceutiques ont uni leurs forces pour créer un leader de ce secteur industriel. L'entité Galaxy Swiss Bourdin Europe a établi son siège administratif à Paris.

Le siège social de la multinationale est situé à Philadelphie, Pennsylvanie, aux Etats-Unis.

La France a été choisie comme témoin pour l'amélioration du suivi de l'activité de visite

L'application mis en place

L'application présentée dans ce compte rendu permet de gérer la chaine d'approvisionnement des différents laboratoires de l'entreprise GSB.

Elle permet aux différents laboratoires de commander des médicaments ou du matériel.

Mise en place

Prérequis

XAMPP

XAMPP permet de mettre en place facilement un environnement Apache et une base de données MySQL ou MariaDB.

Pour l'installation il faut se diriger sur [la page de téléchargement de XAMPP](#)

Une fois sur la page, téléchargé la version **PHP 8.2.12** :

Télécharger

XAMPP est une distribution Apache facile à installer. Elle contient MySQL, PHP et Perl. Il suffit de télécharger et de lancer l'installateur. C'est aussi simple que ça.

 **XAMPP pour Windows 8.0.30, 8.1.25 & 8.2.12**

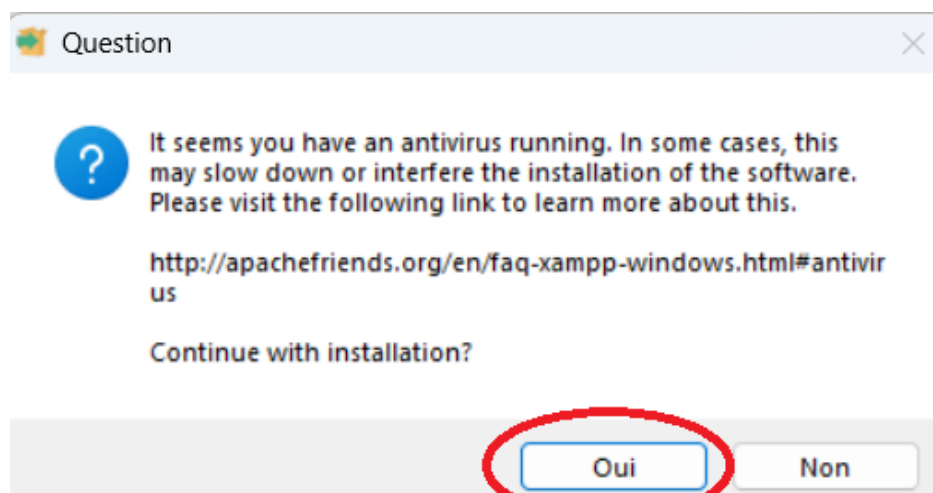
Version		Code de vérification		Taille
8.0.30 / PHP 8.0.30	Contenu	md5	sha1	Télécharger (64 bit) 144 Mb
8.1.25 / PHP 8.1.25	Contenu	md5	sha1	Télécharger (64 bit) 148 Mb
8.2.12 / PHP 8.2.12	Contenu	md5	sha1	Télécharger (64 bit) 49 Mb

[Pré-requis](#) [Plus de téléchargements »](#)

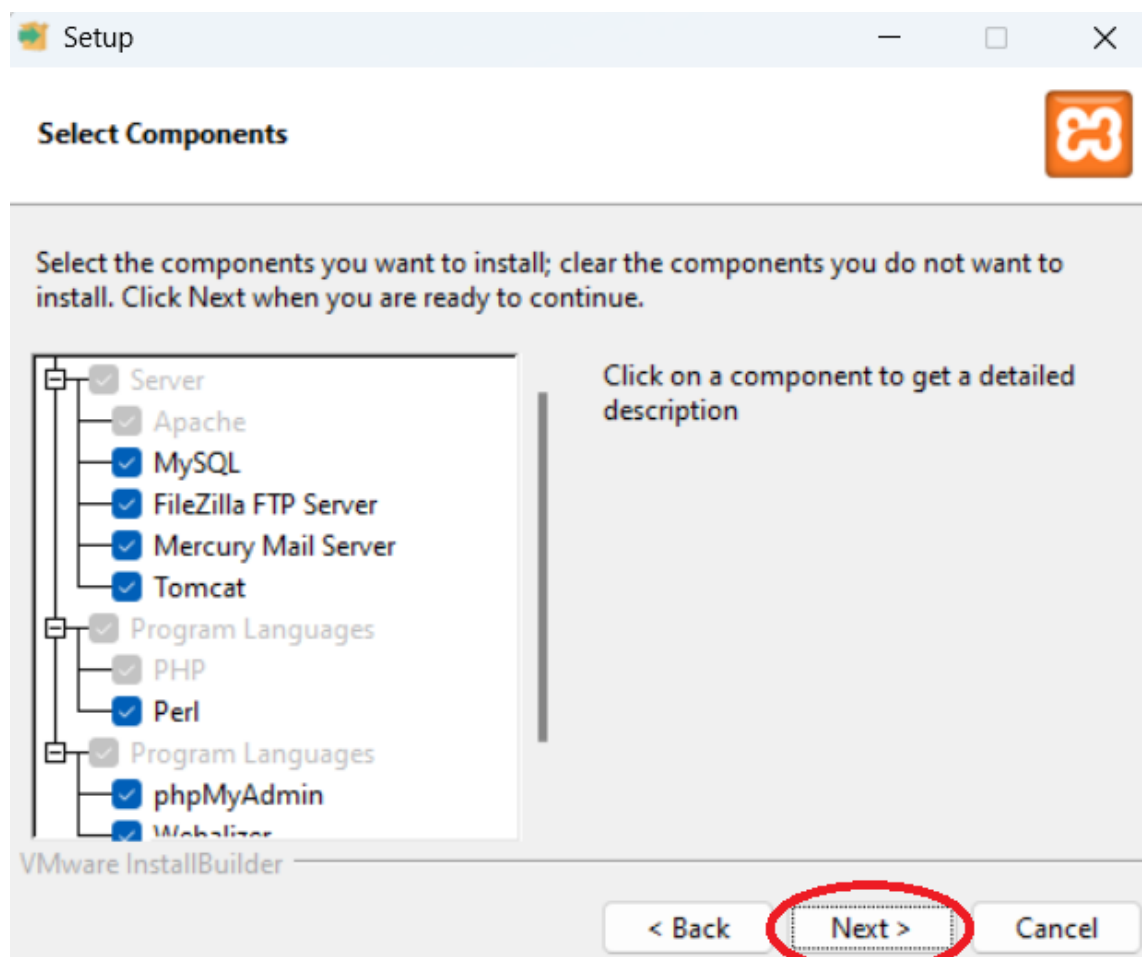
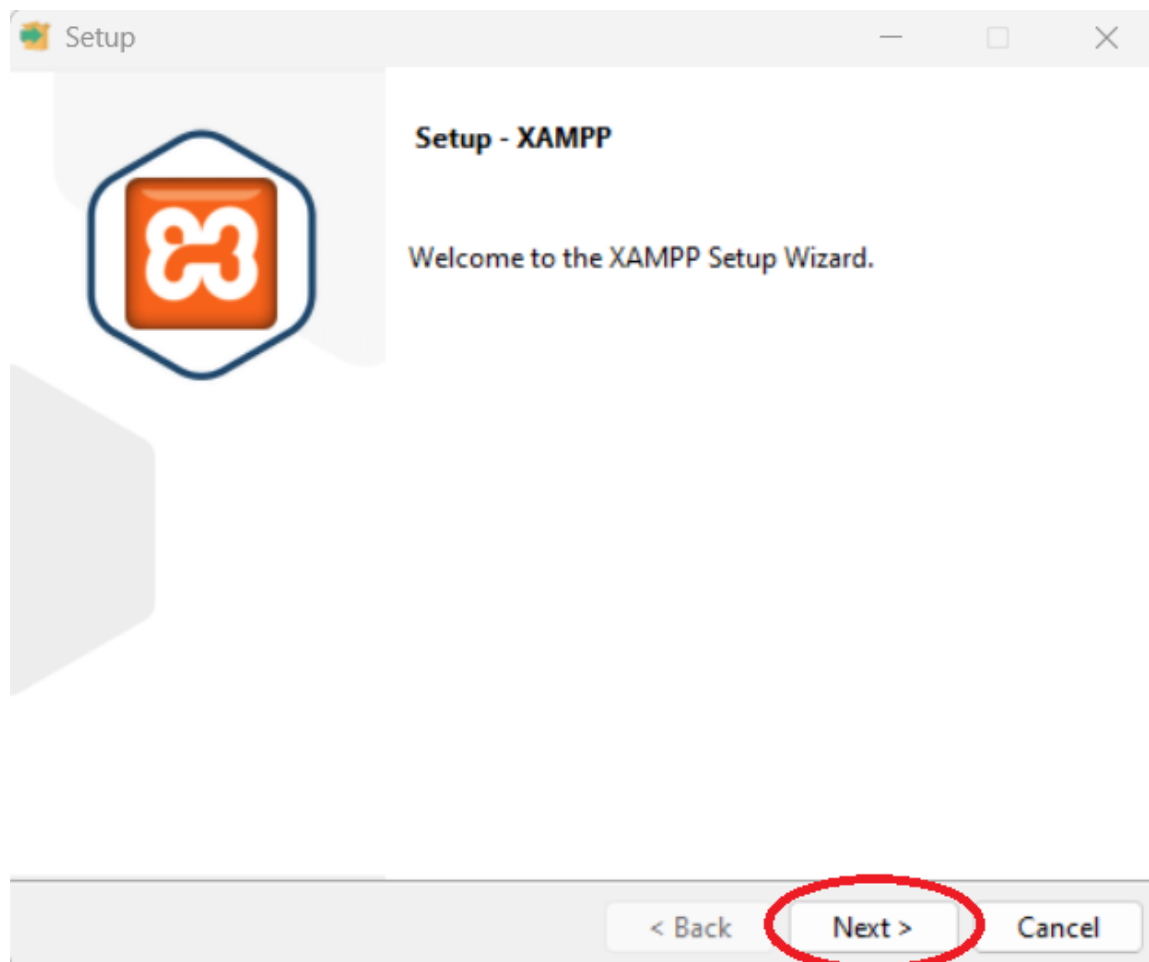
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Une fois téléchargé, lancer l'exécutable.

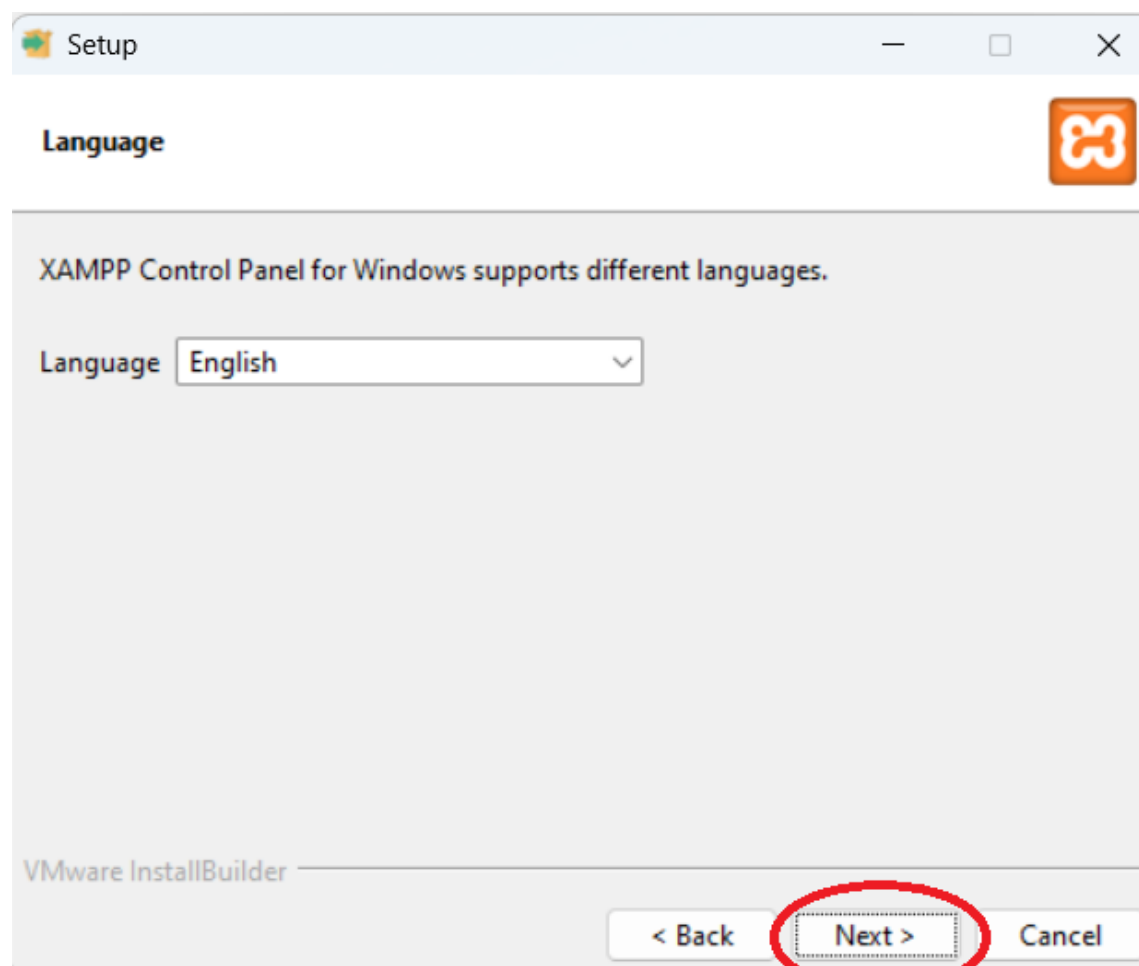
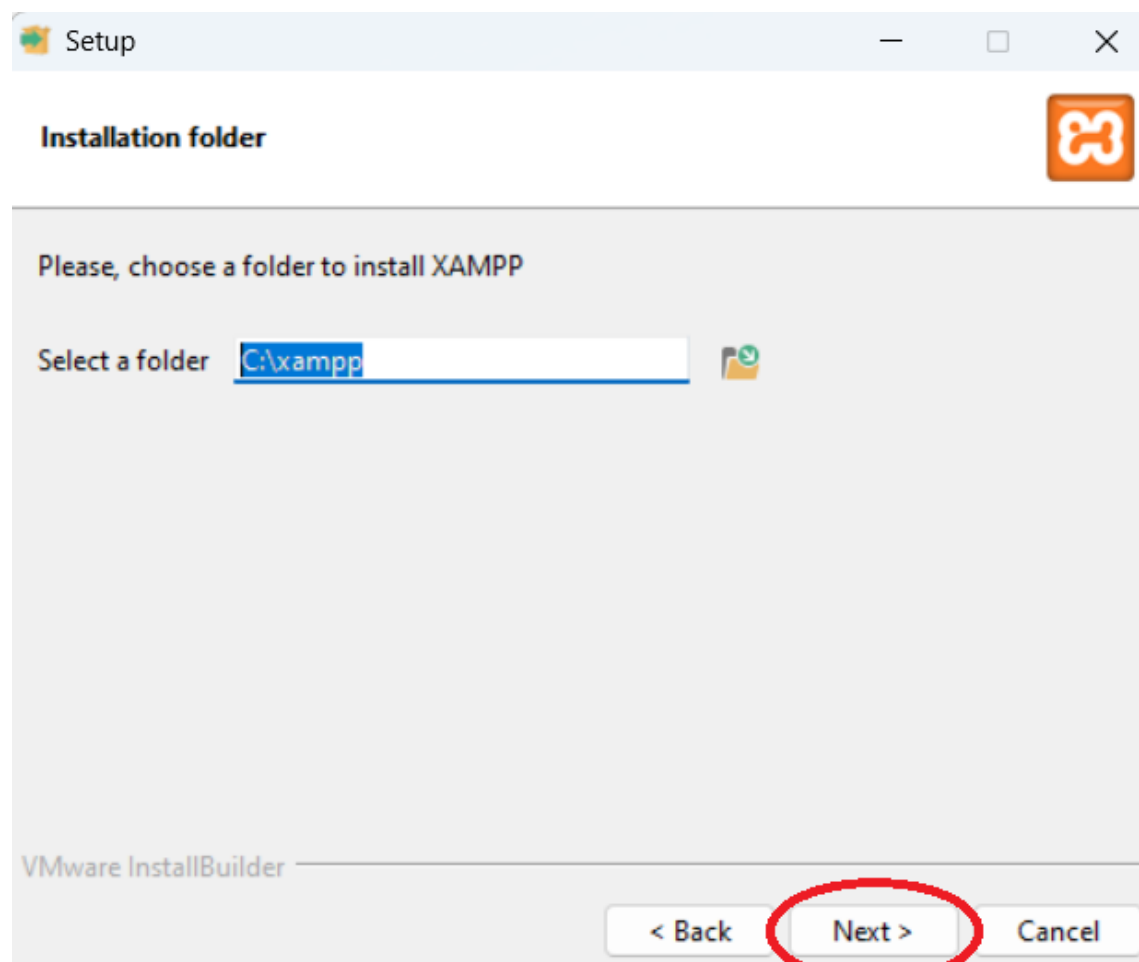
Il est possible qu'une fenêtre vous prévienne que votre antivirus est actif. N'y prêtez pas attention et cliquez sur oui :

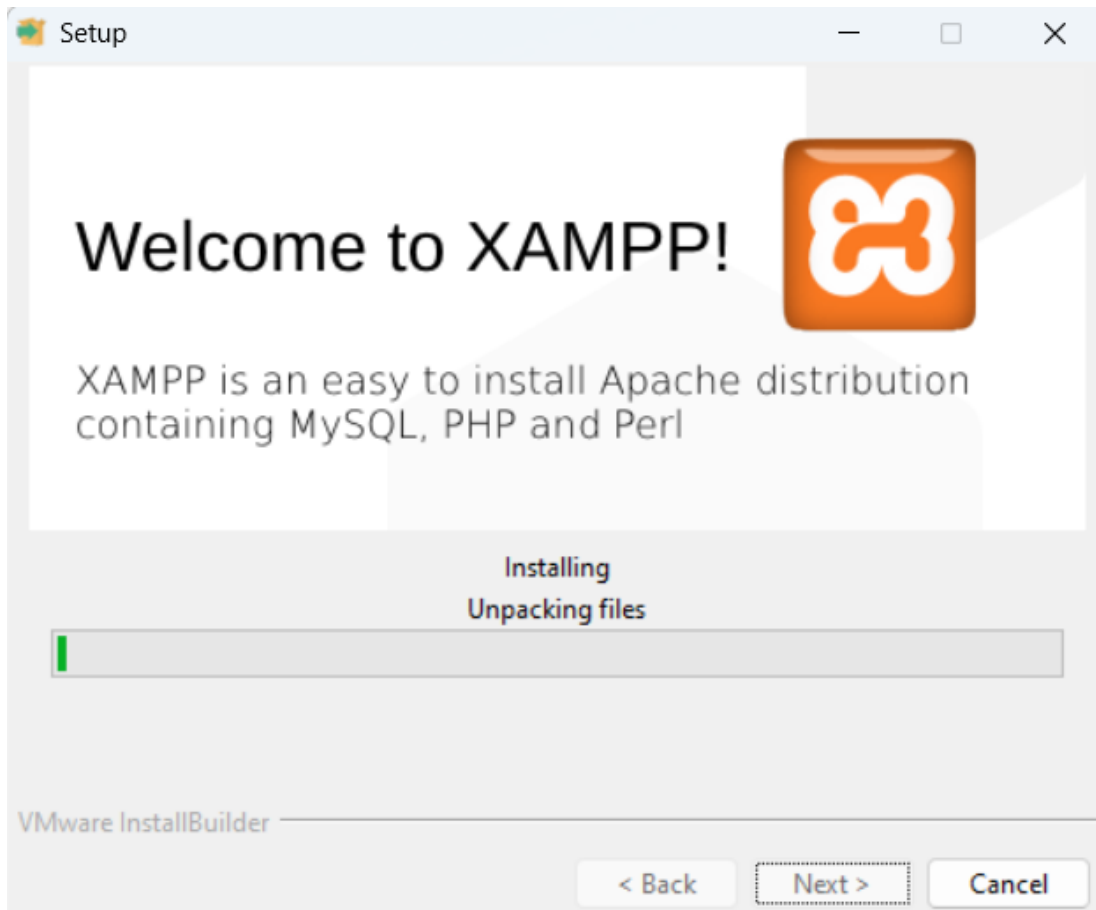


Ensuite, faite suivant plusieurs fois :



Vérifié bien que dans le champ **Select a folder**, le chemin soit C:\xampp



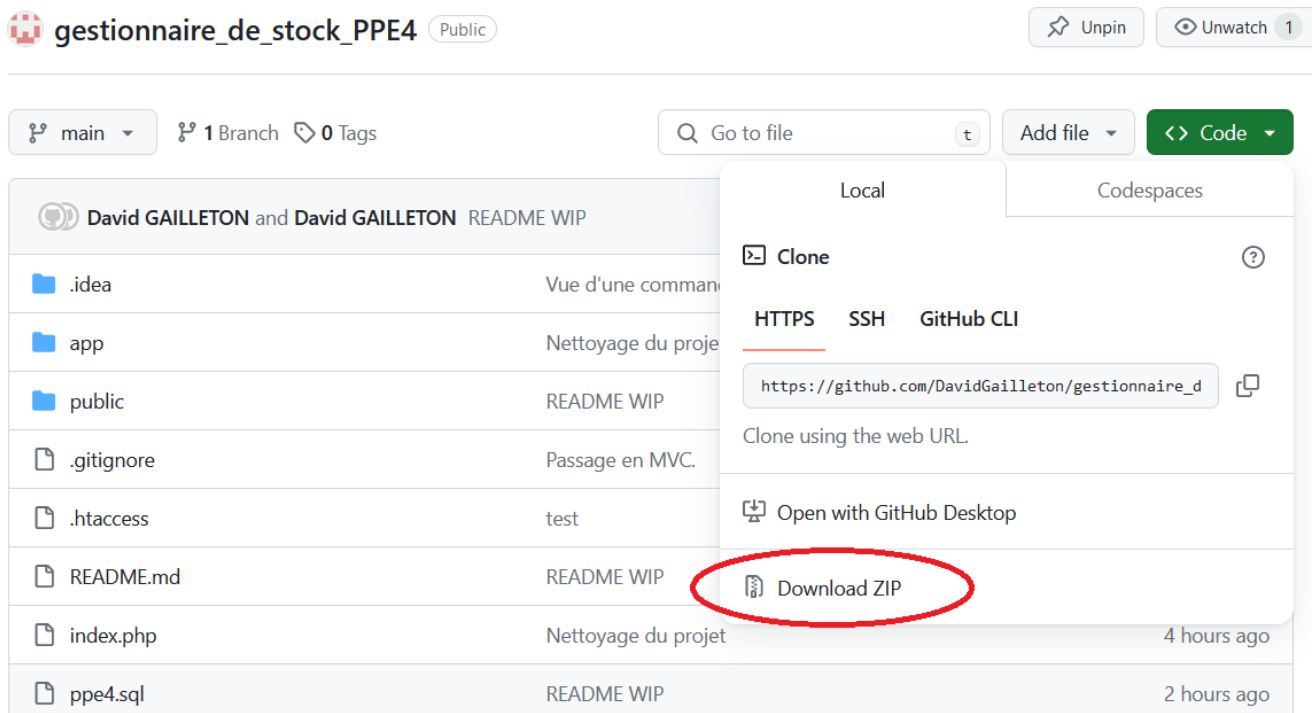


Si le programme d'installation vous propose de redémarrer votre poste, redémarrez le.

Installation du projet

Une fois XAMPP installé, nous pouvons ajouter le projet en local.

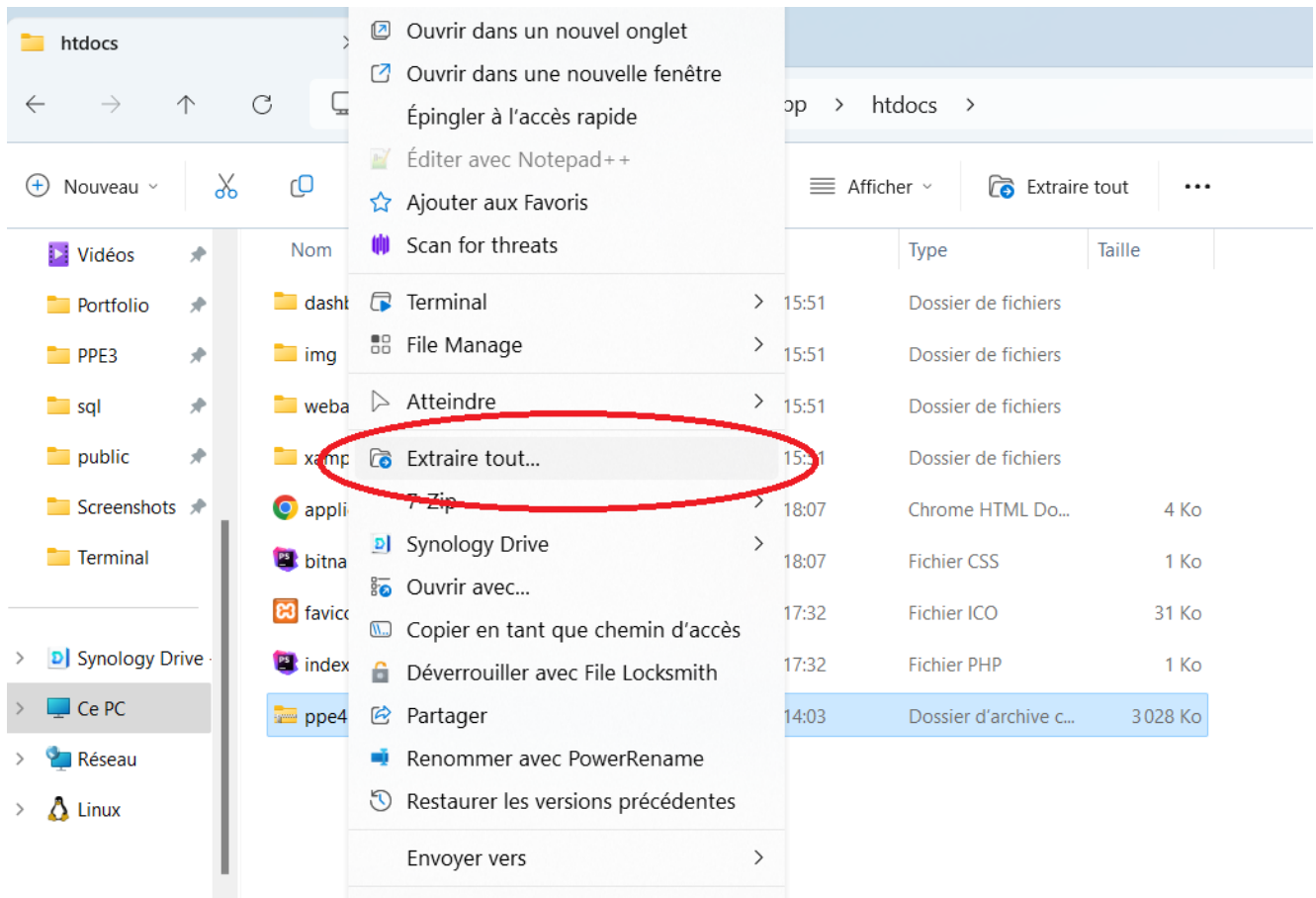
Pour commencer, il faut télécharger le projet depuis [le dépôt Github](#) :



Une fois téléchargé, extraire le dossier précédemment téléchargé dans le dossier `C:\xampp\htdocs`.

Attention, l'extracteur vous proposera surement de l'extraire dans le dossier

C:\xampp\htdocs\ppe4 . Bien penser à corriger.





Extraire l'archive

Sélectionner une destination et extraire les fichiers

Les fichiers seront extraits dans ce dossier :

C:\xampp\htdocs\

Parcourir...

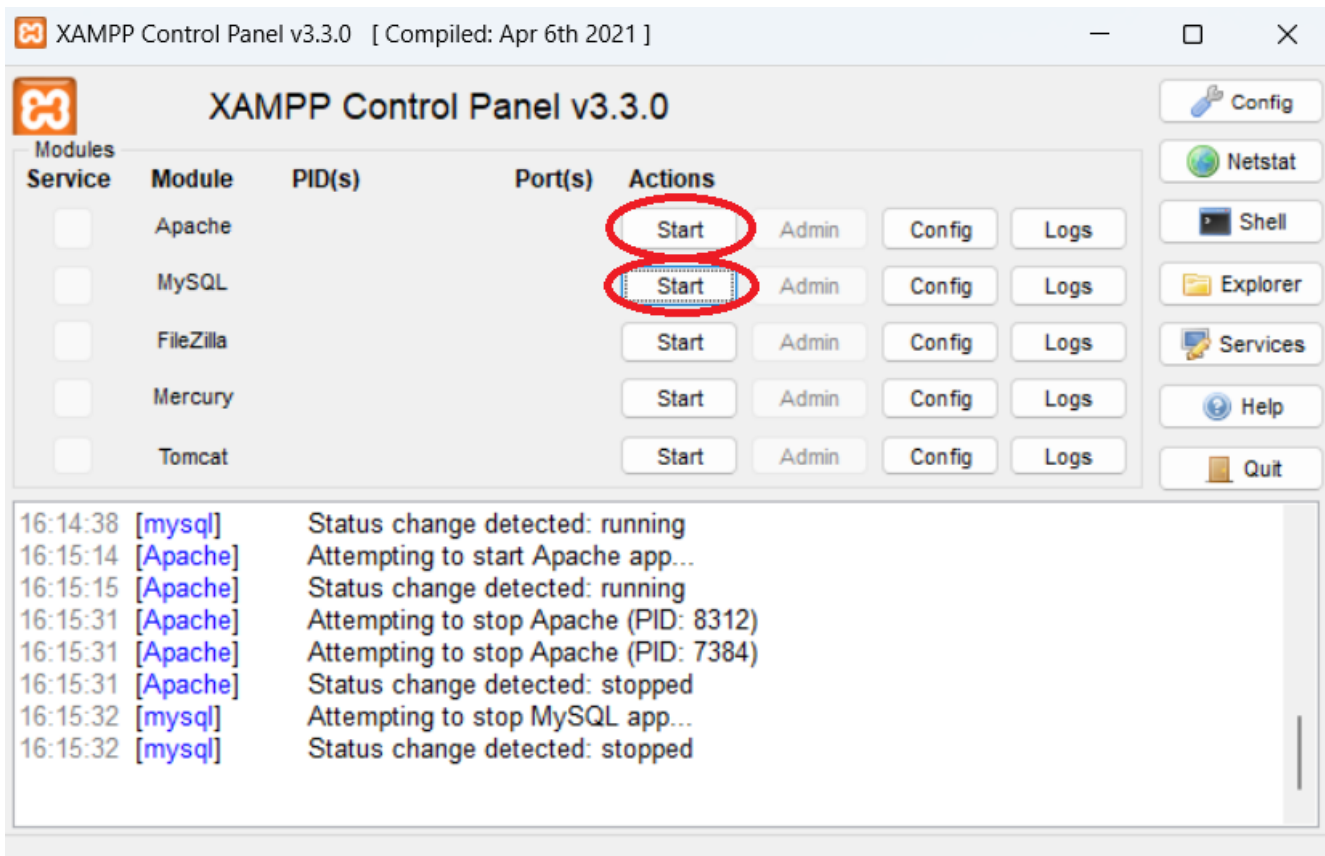
☐ Afficher les dossiers extraits une fois l

Extraire

Annuler

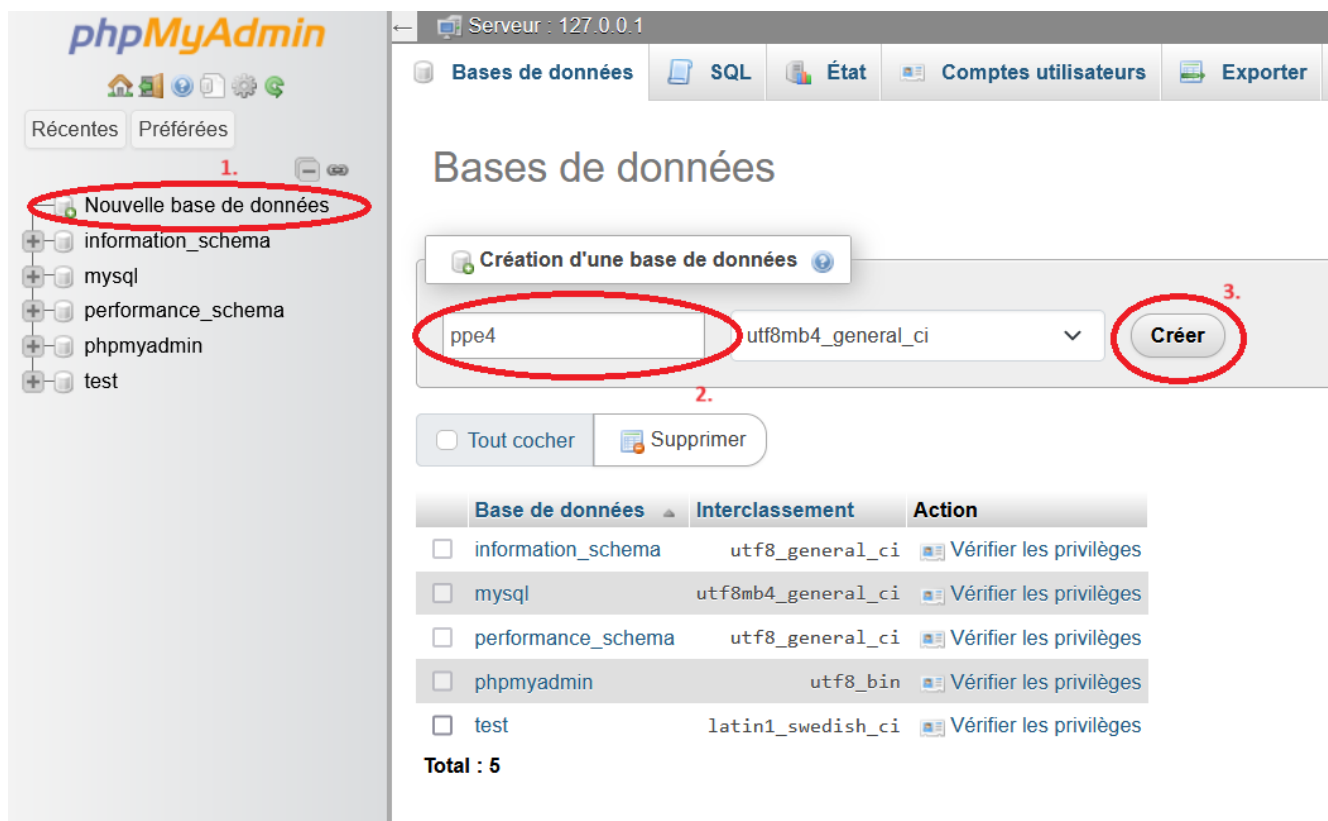
Base de données MariaDB

Pour mettre la base de données en place, il faut en premier temps démarrer **Apache** et **MySQL** depuis XAMPP :



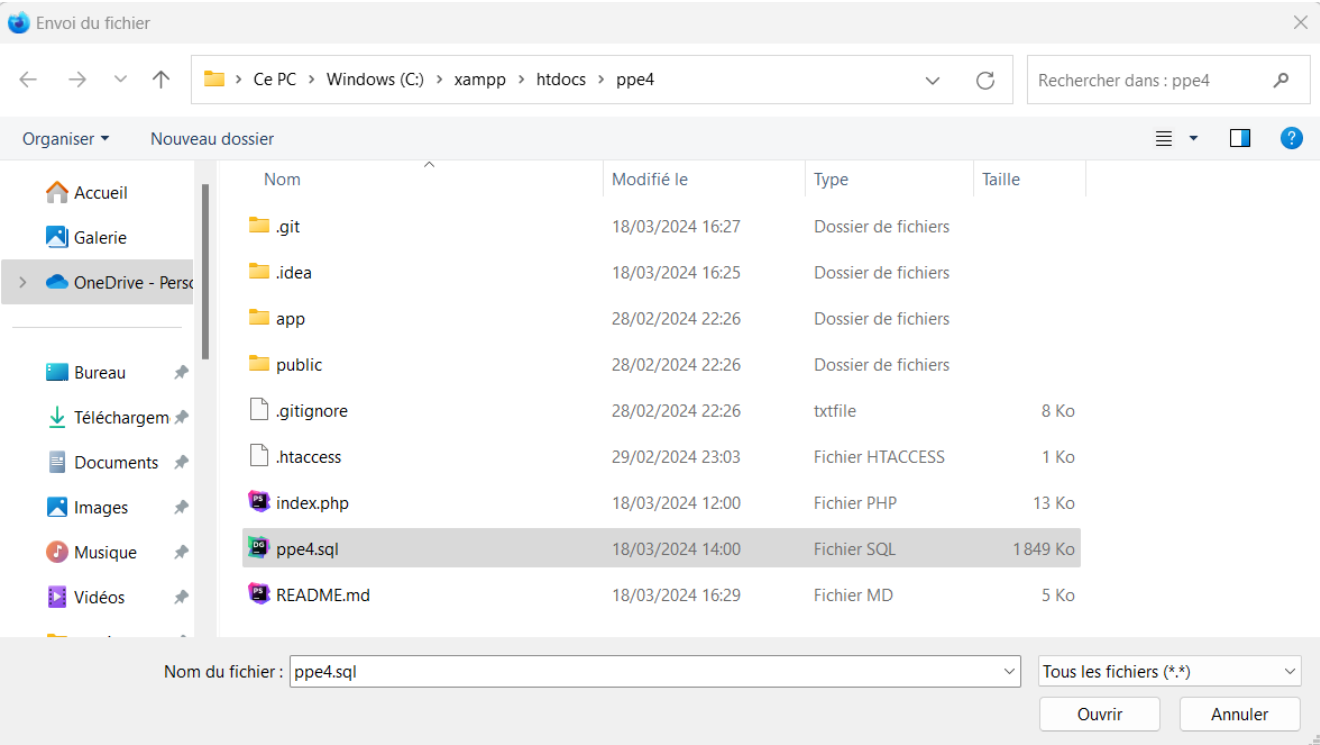
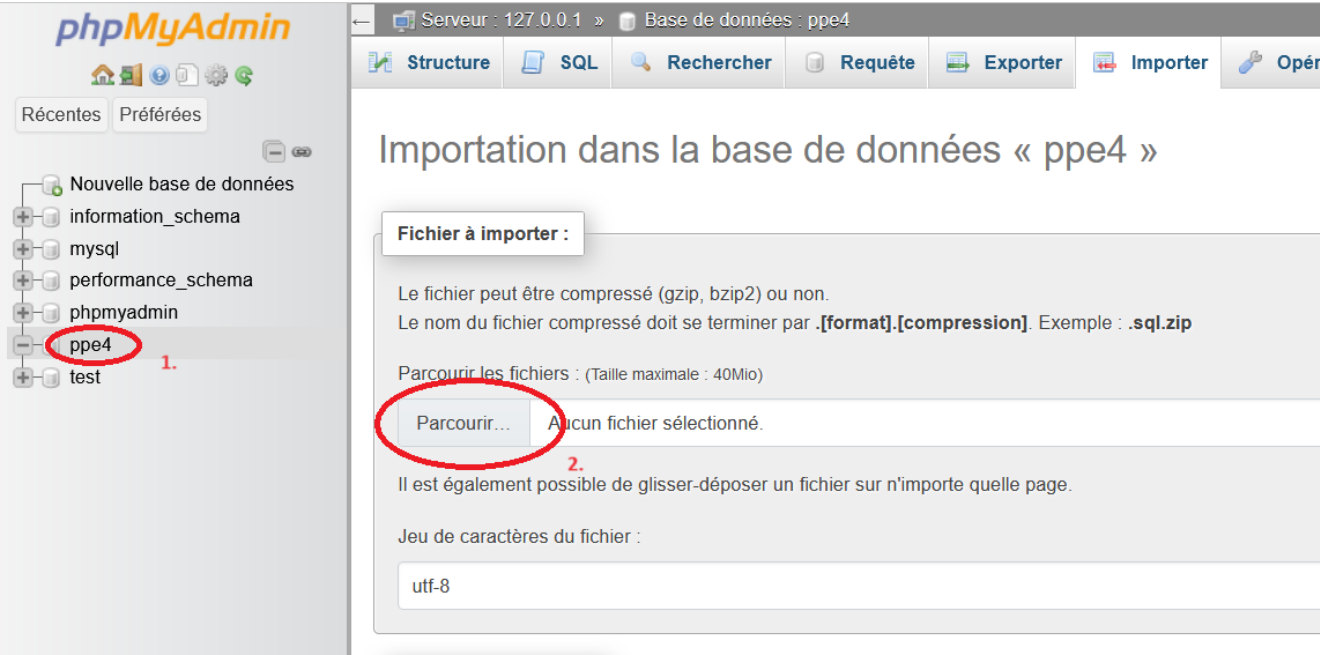
Il est maintenant possible d'accéder à PHP My Admin pour accéder aux bases de données depuis le navigateur web avec l'adresse <http://localhost/phpmyadmin/>

Pour créer la base de données, il faut en premier temps cliquer sur *Nouvelle base de données* (1.), puis donner un nom à la base de données, dans notre cas ppe4 (2.), puis cliquer sur *Créer* (3.) :



Nous allons ensuite importer la base de données du projet.

Cette base des données est disponible à la racine du projet. L'importation est simplifiée par l'interface **phpmyadmin** :



Format

SQL

Options spécifiques au format :

Mode de compatibilité SQL :

NONE

☒ Ne pas utiliser AUTO_INCREMENT pour la valeur zéro

Importer

Si tout s'est déroulé comme prévu, la base de données devrait se présenter comme ceci :



Pour finir, il va falloir créer un compte utilisateur ayant un accès limité à la base de données.

Pour ce faire, dirigez-vous sur l'accueil (1.), puis dans la catégorie SQL (2.) :



Il faut en suite rentrer la requete sql suivante puis l'exécuter :

```
GRANT ALL PRIVILEGES ON ppe4.* TO 'ppe4'@'localhost' IDENTIFIED BY  
'gAueR8DJ2J_DPU7Zz@c@'; FLUSH PRIVILEGES;
```



Base de données

Tous les médicaments présents dans cette table ont été ajouté depuis la [Base de données publique des médicaments](#).

Matériels

La table *Materiel* est vide, car les informations nécessaires sont déjà présents dans la table [Produits](#).

Panier

La table *Panier*, comme son nom l'indique, contient les différents produits ajoutés au panier par les utilisateurs avec une valeur quantité pour chaque produit ajoutés.

Commande

La table *Commande* contient toutes les informations des commandes.

La valeur booléenne `mouvement_com`, correspond au mouvement de la commande, `false` correspond à une commande en direction des stocks, `true` à une sortie de stock en direction d'un laboratoire.

Les produits contenus dans une autre table nommée *Ligne_commande*

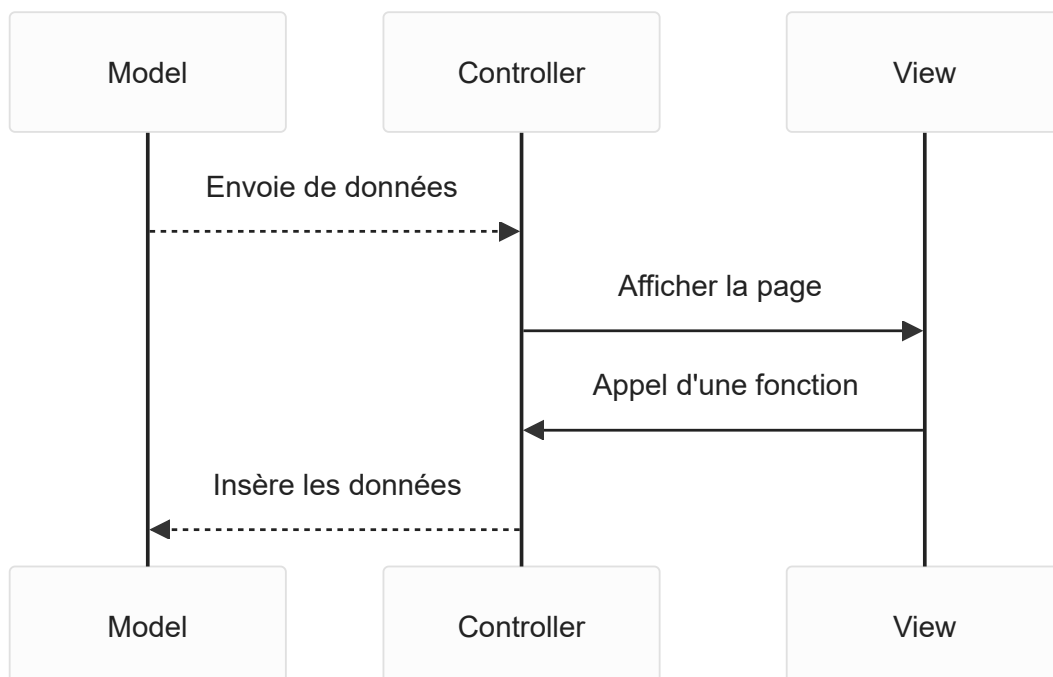
Ligne commande

Les données contenues dans la table *ligne_commande* sont similaires à la table [Panier](#). Elle associe un produit à une commande, en ajoutant une quantité aux produits.

Modèle de conception

Ce projet utilise une architecture **MVC** (Model View Controller).

Cette structure permet de faciliter la maintenance et l'évolutivité des logiciel en séparant les vue des requête SQL.



Le projet

4 comptes de test sont mis à disposition pour tester l'application :

- utilisateur@gsb.fr
- validateur@gsb.fr
- gestionnaire-de-stock@gsb.fr
- admin@gsb.fr

Les 4 comptes possèdent le même mot de passe : **Password@123**

Sécurité

La sécurité est une partie extrêmement importante dans n'importe quelles applications, mais encore plus dans le secteur médical.

BCrypt

Pour encrypter les mots de passe dans la base de données, j'ai choisi d'utiliser l'algorithme de BCrypt. BCrypt permet d'encrypter un mot de passe dans une chaîne de caractère unique et indéchiffrable.

Unique, car même avec le même mot de passe, la clé sera différente.

En PHP, on utilise une méthode nommée `password_hash()` :

```
/**
 * Crypt le mot de passe mis en paramètre puis retourne son hash
 *
 * @param string $mot_de_passe
 * @return string
 */
public function crypter_mot_de_passe(string $mot_de_passe): string
{
    return password_hash($mot_de_passe, PASSWORD_BCRYPT, ["cost" => 13]);
}
```

La seule manière de vérifier si un mot de passe est correct est de le comparer avec la version cryptée. En PHP, on utilise la méthode `password_verify()` :

```
/**
 * Vérifie si le mot de passe mis en paramètre est le bon mot de passe, retourne
 true si c'est le cas, false sinon
 *
 * @param string $email
 * @param string $mot_de_passe
 * @return bool
 */
public function verifier_mot_de_passe(
    string $email,
    string $mot_de_passe
): bool {
```

```

require_once ROOT . "app/models/Utilisateur.php";
$utilisateur = new Utilisateur();

$mot_de_passe_importe = $utilisateur->selectionner_mot_de_passe($email);

return password_verify($mot_de_passe, $mot_de_passe_importe);
}

```

L'algorithme permet donc de protéger les utilisateurs en cas de fuite de la base de données ou du code source.

JWT (JSON Web Token)

JSON Web Token permet d'inscrire un token JSON sur la session de l'utilisateur, dans le stockage local, ou mieux, dans le cookie du navigateur.

Le token JWT est composé de 3 parties, le **header** (en tête), le **payload** et la **signature**.

Header

Le header contient les informations du token permettant de l'identifier en tant que token JWT.

```

{
  "alg": "HS256",
  "typ": "JWT"
}

```

Payload

Le payload permet de contenir toutes les informations que l'on souhaite enregistrer sur le navigateur de l'utilisateur.

Dans notre cas, nous allons enregistrer son **id**, son **email**, son **role**, sa date d'émission (**iat**) et sa date d'expiration (**exp**).

```

{
  "user_id": "9",
  "user_email": "utilisateur@gmail.com",
  "user_role": "utilisateur",
  "iat": 1710779138,
  "exp": 1710793538
}

```

Signature

La partie la plus importante du JWT est la signature. Elle permet d'assurer l'authenticité du token via une clé privée qui ne doit absolument pas être diffusée.

La signature est un hash généré via un algorithme de hashage choisi préalablement. Pour créer cet hash nous allons concaténer l'header et le payload puis l'hasher avec la clé secrète que nous aurons au préalable encodé en [base 64](#).

Dans notre cas, l'algorithme de hashage utilisé est le [SHA256](#) :

```
// Encodage en base 64 des éléments de la signature
$base_64_header = base64_encode(json_encode(JWT_HEADER));
$base_64_payload = base64_encode(json_encode($payload));
$secret = base64_encode(JWT_SECRET);

//generation de la signature
$signature = hash_hmac(
    "sha256",
    $base_64_header . "." . $base_64_payload, $secret
    true,
);
```

Une fois le tout généré, nous allons concaténer l'header, le payload et la signature en les séparant par un point ce qui donnera :

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

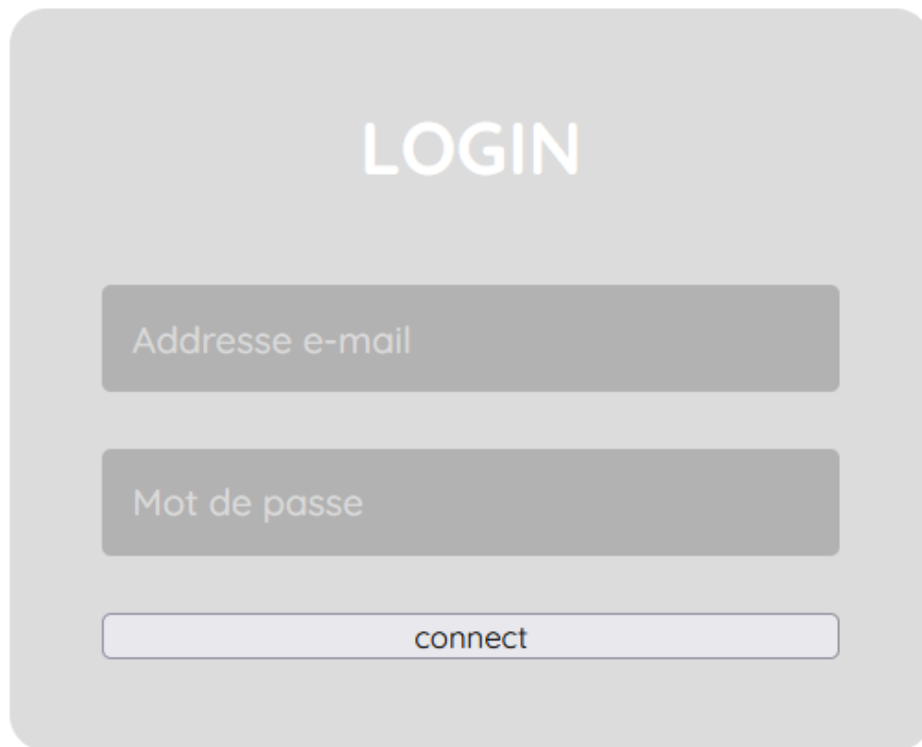
Injection SQL

Pour éviter le risque d'injection SQL, PDO a introduit les [Requête préparée](#) via la fonction `prepare()` .

```
public function get_one()
{
    $sql = "GET * FROM :table WHERE :id_table = :id";
    // préparation de la requête
    $query = $this->pdo->prepare($sql);
    $query->execute(
        ["table" => $this->table],
        ["id_table" => "id_" . $this->table[0 - 2]],
        ["id" => $this->id],
    );
}
```

Page de connexion

La page de connexion présente un formulaire où rentrer son email et son mot de passe :

A login form with a light gray background. At the top, the word "LOGIN" is written in large, white, bold, sans-serif capital letters. Below it, there are two dark gray rectangular input fields. The first field contains the text "Adresse e-mail" in a light gray font. The second field contains the text "Mot de passe" in a light gray font. Below these fields is a light gray rectangular button with the text "connect" in a dark gray font.

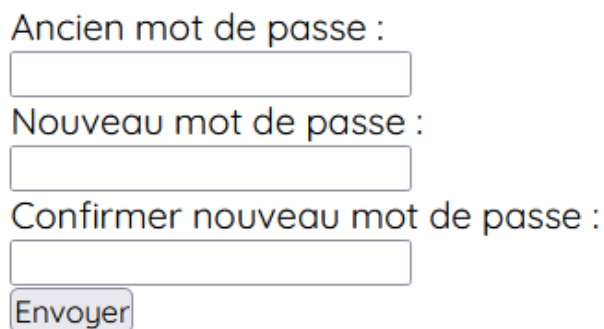
Lors de la soumission du formulaire, le programme PHP va vérifier plusieurs choses :

- En premier temps, il va simplement vérifier si l'utilisateur existe.
- Ensuite, il va vérifier si le compte n'est pas bloqué (valeurs booléenne *compte_desactive_utilisateur* dans la table **Utilisateur**).
- Pour finir, il va vérifier si le mot de passe entré est valide avec [BCrypt](#).

Si toutes ses conditions sont valides, le programme vérifiera si le mot de passe doit être modifié (ce qui arrive lors de la première connexion ou si un administrateur a réinitialisé le mot de passe).

Nouveau mot de passe

Si un nouveau mot de passe doit être mis en place, l'utilisateur sera redirigé sur le formulaire suivant :

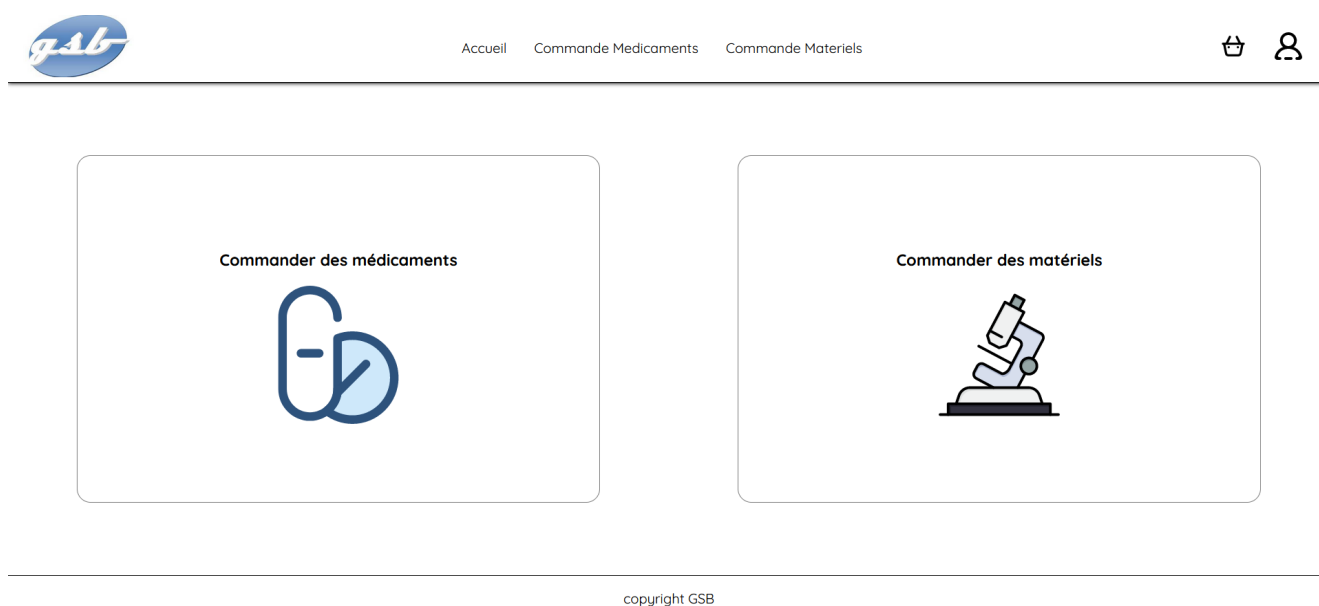
A form for changing a password. It consists of three text input fields stacked vertically. The first field is labeled "Ancien mot de passe :". The second field is labeled "Nouveau mot de passe :". The third field is labeled "Confirmer nouveau mot de passe :". Below the third field is a button labeled "Envoyer".

Pour valider le changement de mot de passe, plusieurs conditions doivent être remplies :

- Minimum 8 caractères
- Minimum 1 majuscule
- Minimum 1 minuscule
- Minimum 1 caractère spécial
- Minimum 1 chiffre

Si le code JS présent dans la page n'a pas été modifié, l'utilisateur devrait être prévenu si l'une de ces conditions n'est pas respectée (Le JavaScript permet seulement de prévenir l'utilisateur, si la requête est quand même envoyée, le code PHP vérifiera et refusera la modification du mot de passe).

Espace Utilisateur



L'espace utilisateur du site permet aux laboratoires d'effectuer des commandes qui devront être, par la suite, validé.

Choix des produits

L'espace de commandes de médicaments et de matériels fonctionne exactement de la même manière, la seule différence est la façon dont sont affichés les produits.

Cette page est composée de 3 éléments :

- La liste des produits
- La barre de recherche
- Le choix de page

Liste des produits

La liste des produits est affichée par des composants :

TAVANIC 500 mg, comprimé pellicule sécable
- CIS : 69486152
orale

En stock

6



Ajouter au panier

En ajoutant un produit au panier, l'application va ajouter l'ajout à la table [panier](#) via la requête suivante :

```
INSERT INTO panier (id_utilisateur, id_produit, quantite);
VALUES (:id_utilisateur, :id_produit, :quantite);
```

Barre de recherche

Lors de la soumission d'une recherche, la requête exécutée ajoutera une variable `LIKE` :

```
SELECT produits.id_pro AS id, libelle_pro AS libelle, description_pro AS description,
qte_stock_pro AS quantite_stock, forme_med AS forme, cis_med AS cis
FROM medicaments
INNER JOIN produits ON medicaments.id_pro = produits.id_pro
WHERE produits.libelle_pro
LIKE :recherche
LIMIT :offset , 25;
```

Choix de page

Pour éviter de faire planter la page (la base de données des médicaments contient 15767 lignes), la liste des produits est limitée à 25 par page.

Pour choisir une page, une barre disponible en bas de page est disponible :

<	1	...	2	3	4	5	6	...	631	>
---	---	-----	---	---	---	---	---	-----	-----	---

Panier

La page du panier présente la liste des produits ajoutés, ainsi qu'un bouton permettant de confirmer la commande :

AMOXICILLINE/ACIDE CLAVULANIQUE VIATRIS 1 g/125 mg En stock

- CIS : 60998361
orale

7



Supprimer

Spectrophotometre

En stock

Appareil de mesure de l'absorbance des
liquides permettant de determiner la
concentration de certaines substances

1



Supprimer

Confirmer la commande

Il est possible de supprimer un produit de panier, ainsi que de modifier sa quantité.

Lors du changement de quantité un script JS est exécuté permettant d'effectuer une requête à chaque changement :

```
document.getElementById("formulaire_'. $i .'").addEventListener("input", function(){  
    let form = document.getElementById("formulaire_'. $i .'");  
    let formData = new FormData(form);  
    fetch("index.php?action=modifier_qte_produit_panier", {  
        method: "POST",  
        body: formData  
    })  
});
```

```
UPDATE panier  
SET qte = :qte  
WHERE id_uti = :id_utilisateur AND id_pro = :id_produit;
```

A la confirmation de la commande, L'application va en premier temps créer une commande via la requête suivante :

```
INSERT INTO commande (commande.date_com, commande.mouvement_com,  
commande.id_uti_Utilisateur, commande.statut_com)  
VALUES (NOW(), :mouvement, :id_utilisateur, :statut);
```

Puis dans un second temps, créer pour chaque élément du panier une ligne dans la table *ligne_commande* :

```
foreach ($produits as $produit) {  
    $ligne_commande->inserer_ligne_commande(  
        $id_commande,
```

```
        $produit["id"],  
        $produit["qte"],  
    );  
}
```

```
INSERT INTO ligne_commande (id_com, id_pro, qte)  
VALUES (:id_commande, :id_produit, :qte);
```

Enfin, pour finir, vider le panier de l'utilisateur :

```
DELETE  
FROM panier  
WHERE id_uti = :id_utilisateur
```

Vos commandes

Il est possible d'accéder aux commandes faites au préalable via le menu déroulant disponible en cliquant sur l'icône profile :



Vos commandes
Deconnexion

Cette page présente l'ensemble des commandes faites par l'utilisateur :

Commande n°26

Date de commande : 19/03/2024

Statut : en_cours_de_preparation

Commande n°25

Date de commande : 18/03/2024

Statut : refuse

En sélectionnant une, vous accédez aux informations de la commande :

Centrifugeuse

7

Appareil permettant de separer les melanges homogenes ou heterogenes par la force centrifuge

Espace Valideur

L'espace valideur permet de valider les commandes faites par les utilisateurs.

On accède aux commandes en attente de validation en cliquant sur l'un des raccourcis proposés :



Accueil

Valider une commande



copyright GSB

Vous pouvez ensuite sélectionner une commande à valider :

Commande n°27

Date de commande : 20/03/2024

Statut : en_attente

test test

Sur cette commande, vous pouvez seulement valider ou refuser la commande :

en_attente

AMOXICILLINE/ACIDE CLAVULANIQUE VIATRIS 1 g/125 mg
- CIS : 60998361
orale

CLIMASTON 1 mg/5 mg, comprimé pellicule
- CIS : 66257038
orale

Spectrophotometre
Appareil de mesure de l'absorbance des liquides permettant de déterminer la concentration de certaines substances

Accepter la commande

Refuser la commande

Valider commande

Si la commande est validée, 2 requêtes sont exécutées.

La première permet de modifier le statut de la commande :

```
UPDATE commande
SET commande.date_val_com = NOW(), commande.id_uti_valisateur = :id_valisateur,
commande.statut_com = :statut
WHERE commande.id_com = :id_commande;
```

La seconde permet de réduire le nombre de produits disponibles en stock :

```
UPDATE produits
SET qte_stock_pro = produits.qte_stock_pro - :quantite
WHERE id_pro = :id;
```

refuser commande

Si la commande est refusée, seulement le statut sera modifié :

```
UPDATE commande
SET commande.date_val_com = NOW(), commande.id_uti_valisateur = :id_valisateur,
commande.statut_com = :statut
WHERE commande.id_com = :id_commande;
```



Gestion des utilisateurs



Créer un utilisateur



https://localhost/ooe4/index.php?page=liste_utilisateur

copyright GSB

Le compte administrateur permet seulement de gérer les comptes utilisateur.

Il est possible de :

- Créer un compte
- Modifier un compte
- Désactiver un compte
- Supprimer un compte
- Modifier le mot de passe

Création d'un utilisateur

Email

Prenom

Nom de famille

Mot de passe temporaire

Role

Créer utilisateur

L'espace de création d'un compte présente plusieurs champs.

Email

L'email de l'utilisateur est l'identifiant de connexion de ce dernier. L'email doit être unique parmi tous les comptes non archivés.

Prenom et Nom de famille

Le prénom et le nom de famille sont seulement présents à titre indicatif.

Mot de passe temporaire

Le mot de passe mis en place par l'administrateur devra obligatoirement être modifié à la première connexion de l'utilisateur.

Role

Les rôles affichés dans la liste sont importés depuis [la table Role](#) de la base de données.

Soumission du formulaire

Lors de la soumission du formulaire la fonction suivante sera exécutés :

```
public function creer_utilisateur(  
    string $mot_de_passe,  
    string $email,  
    string $prenom,  
    string $nom,  
    string $libelle_role  
) : bool {  
    require_once ROOT . "app/models/Role.php";  
    $role_model = new Role();  
    $role = $role_model->selectionner_role_par_libelle($libelle_role);  
    $id_role = $role_model->selectionner_id_role($role);  
  
    require_once ROOT . "app/controllers/Bcrypt.php";  
    $bcrypt = new Bcrypt();  
    $mot_de_passe_crypte = $bcrypt->crypter_mot_de_passe($mot_de_passe);  
  
    require_once ROOT . "app/models/Utilisateur.php";  
    $utilisateur_model = new Utilisateur();  
    $result = $utilisateur_model->creer_utilisateur(  
        $mot_de_passe_crypte,  
        $email,  
        $prenom,  
        $nom,  
        $id_role  
    );  
  
    if (!$result){  
        echo '<script>alert("Une erreur s\'est produit")</script>';  
    }  
    return $result;  
}
```

```
# Requete exécuté avec la fonction creer_utilisateur()  
INSERT INTO utilisateur (email_uti, password_uti, nom_uti, prenom_uti, id_rol)  
VALUES (:email, :mot_de_passe, :nom, :prenom, :id_role);
```

Gestion des utilisateurs

Sur la page de gestion des utilisateurs, en sélectionnant un compte, vous arriverez sur cette page :

Email

Prenom

Nom de famille

Role

Modification

Pour modifier les informations du compte, il faut simplement modifier les différents champs que l'on souhaite modifier pour cliquer sur le bouton `Modifier l'utilisateur`.

Lors de la soumission du formulaire, la requête SQL suivante est exécutée :

```
UPDATE utilisateur  
SET email_uti = :email, prenom_uti = :prenom, nom_uti = :nom, id_rol = :id_role  
WHERE id_uti = :id_utilisateur  
AND est_archive_uti = false;
```

Reinitialiser mot de passe

En cliquant sur le bouton *Reinitialiser mot de passe*, une pop-up va s'afficher vous demandant d'entrer un mot de passe temporaire :

The screenshot shows a web interface with a form containing an 'Email' field with the value 'validateur@gmail.com'. Below the form, there are two buttons: 'Modifier l'utilisateur' and 'Supprimer l'utilisateur'. A modal dialog is open in the center, titled 'localhost' with a globe icon. The dialog contains the text 'Veuillez entrer le nouveau mot de passe pour cet utilisateur' and a text input field. At the bottom of the dialog are 'OK' and 'Annuler' buttons.

Une fois soumis, le mot de passe sera encrypté, puis modifier sur [la table Utilisateur](#).
A la prochaine connexion, l'utilisateur sera forcé à modifier le mot de passe temporaire mis en place.

Désactiver l'utilisateur

Il est également possible de désactiver l'utilisateur.

La désactivation va simplement changer la valeur booléenne `compte_desactivé_utili` de la table *Utilisateur*, ce qui l'empêchera de se connecter.

Suppression de l'utilisateur

En cliquant sur le bouton *Supprimer utilisateur*, un pop-up demandera une seconde confirmation de suppression afin d'éviter toutes erreurs :

The screenshot shows a web interface with a form containing an 'Email' field with the value 'test123@gsb.fr' and a 'Prenom' field. Below the form, there are two buttons: 'Modifier l'utilisateur' and 'Supprimer l'utilisateur'. A modal dialog is open in the center, titled 'localhost' with a globe icon. The dialog contains the text 'Êtes-vous sûr de vouloir supprimer cet utilisateur ?' and 'OK' and 'Annuler' buttons.

Etant donnée que la clé primaire de l'utilisateur peut être liée à d'autres tables, ce dernier n'est pas réellement supprimé de la table utilisateur, mais archivé via la valeur booléenne `est_archive_utili`. L'intégralité de requête `SELECT` faite vers la table **utilisateur** filtre les utilisateurs archivés.

Gestionnaire de stock

La vue du gestionnaire de stockage est exactement la même que l'utilisateur.

La différence vient du fait que lors de la [confirmation d'une commande](#), les paramètres de la fonction appelé sont différents :

```
public function confirmer_la_commande_gestionnaire(
    array $produits,
    int $id_utilisateur,
): void {
    require_once ROOT . "app/models/Commande.php";
    require_once ROOT . "app/models/Ligne_commande.php";
    require_once ROOT . "app/models/Produit.php";
    $commande = new Commande();
    $produit_model = new Produit();
    $id_commande = $commande->insérer_commande(
        $id_utilisateur,
        // false pour une commande entrante
        false,
        // statut de la commande
        "en_cours_de_preparation",
    );

    $ligne_commande = new \ppe4\models\Ligne_commande();
    foreach ($produits as $produit) {
        $ligne_commande->insérer_ligne_commande(
            $id_commande,
            $produit["id"],
            $produit["qte"],
        );
        $produit_model->augmenter_quantite($produit["id"], $produit["qte"]);
    }

    require_once ROOT . "app/models/Panier.php";
    $panier = new \ppe4\models\Panier();
    $panier->vider_le_panier($id_utilisateur);
}
```

Dans ce cas-là, la commande n'a pas besoin d'être validé par un validateur.

Conclusion

La mise en place de l'application de gestion de commandes pour le laboratoire Galaxy Swiss Bourdin (GSB) représente un pas important vers l'optimisation de la chaîne d'approvisionnement et la gestion des stocks au sein de l'entreprise.

Son modèle de conception lui permet une évolutivité ainsi qu'une maintenance facilitée.

La structure de la base de données permet également cette modularité primordiale pour toutes applications.