

Sommaire

Introduction.....	02
Schéma d'enchaînement des écrans.....	03-05
Structure BD mise à jour.....	06-08
Structure Applicative / Script de sauvegarde BD / Charte Graphique	09
Cahier de suivi agile.....	10-12
Étude Juridique	13
Code	14-41

Tester le projet : <https://flourence.fr/projet-3-php>



Introduction

Nous sommes ravis de vous présenter notre projet sur les médicaments et les activités, une plateforme web conçue pour fournir des informations détaillées sur les médicaments et offrir des opportunités d'inscription à diverses activités.

Fonctionnalités Principales :

Exploration des Médicaments : Notre plateforme propose une base de données complète de médicaments, présentant des détails tels que le nom, le laboratoire créateur, la dose maximale journalière et une liste exhaustive des effets secondaires associés.

Informations approfondies : Les utilisateurs peuvent accéder à des informations approfondies sur chaque médicament, notamment ses effets thérapeutiques et secondaires. Ces données sont présentées de manière structurée pour permettre une analyse efficace.

Inscription aux Activités : En plus de la section médicaments, notre plateforme offre une liste diversifiée d'activités auxquelles les utilisateurs peuvent s'inscrire. Chaque activité est accompagnée de détails tels que le nom, la description, la date et le nombre de places disponibles.

Gestion des Utilisateurs : Les utilisateurs ont la possibilité de créer un compte sur notre plateforme, facilitant ainsi l'inscription aux activités et la sauvegarde de leurs préférences en matière de médicaments.

Architecture MVC :

Notre projet est basé sur une architecture MVC (Modèle-Vue-Contrôleur) pour une organisation efficace du code et une séparation claire des préoccupations. Voici comment chaque composant est utilisé :


- **Modèle (Model)** : Responsable de la gestion des données, le modèle interagit avec la base de données MySQL pour stocker et récupérer les informations sur les médicaments, les activités et les utilisateurs.
- **Vue (View)** : La vue est chargée de présenter les données aux utilisateurs de manière claire et conviviale. Dans notre projet, les vues HTML sont utilisées pour afficher les détails des médicaments et des activités.
- **Contrôleur (Controller)** : Le contrôleur agit comme un intermédiaire entre le modèle et la vue, traitant les demandes des utilisateurs, récupérant les données nécessaires et les transmettant aux vues correspondantes pour affichage.

Conclusion :

Notre projet Médicaments & Activités offre une expérience interactive et informatique unique, permettant aux utilisateurs d'explorer les médicaments de manière approfondie et de participer à des activités enrichissantes. Avec une architecture solide et des fonctionnalités conviviales, nous sommes convaincus que notre plateforme répondra aux besoins variés de nos utilisateurs dans le domaine de la santé et des loisirs.

Schéma d'enchaînement des écrans :

Accueil : Permet d'accéder à nos pages



Accueil : Accéder à nos services

Nino FEDOU - Mathias HANY - Stefan FLOURENCE

Liste médicaments

Consultez aisément notre répertoire de médicaments.

Accédez à notre liste

Inscriptions aux événements

Découvrez les événements proposés par notre société. Profitez d'une inscription gratuite pour rester informé(e) sur notre actualité et sur les avancées dans le domaine médical.


Accéder aux événements

Mention juridique

Retrouvez l'ensemble de nos détails juridiques concernant notre site Internet.

Accéder à la Page

Liste Médicaments : Affiche notre liste de médicaments (provenant de notre base de données MySQL et récupérée via notre API) avec un bouton pour afficher leurs détails individuels.



Liste des Médicaments

Nom	Date de Création	Laboratoire Créateur	Détails
Paracétamol	2024-03-12	Sanofi	Détails
Ibuprofène	2024-03-11	Pfizer	Détails
Aspirine	2024-03-10	Novartis	Détails
Amoxicilline	2024-03-09	Roche	Détails
Loratadine	2024-03-08	GlaxoSmithKline	Détails
Omeprazole	2024-03-07	Merck	Détails
Simvastatine	2024-03-06	AstraZeneca	Détails
Levothyroxine	2024-03-05	Johnson & Johnson	Détails
Mélatonine	2024-03-04	Bayer	Détails
Diazépam	2024-03-03	Abbott	Détails
Ranitidine	2024-03-02	Boehringer Ingelheim	Détails

Détails Individuels

Détails du Médicament

Nom du médicament

Paracétamol

Date de Création

2024-03-12

Laboratoire Créateur

Sanofi

Dose journalière maximale

500mg

Effets Thérapeutiques

- Antalgique
- Anti-inflammatoire

Effets Secondaires

- Nausées
- Maux de tête

Médicaments Réactifs

- Ibuprofène
 - Peut causer des vertiges et une somnolence accrue
- Aspirine
 - Interaction déconseillée chez les personnes souffrant de troubles gastro-intestinaux

Retour à la liste

Liste des activités : Affiche les activités prévues dans notre BD, et permet de s'inscrire avec un bouton, avec pour chaque activité le nombre de place restante affichée.



Liste des activités

Conférence sur les avancées en oncologie

Présentation des dernières avancées en oncologie par des experts renommés. Cette conférence abordera les nouveaux traitements, les thérapies ciblées et les défis à venir dans le domaine de l'oncologie.

Places restantes : 50

15/04/2024 09:00

Inscription

Atelier sur la gestion de la douleur chronique

Formation sur les approches modernes de gestion de la douleur chronique. Cet atelier inclura des discussions sur les techniques pharmacologiques, non pharmacologiques et multidisciplinaires pour gérer la douleur chronique.

Places restantes : 50

20/04/2024 14:00

Inscription

Séminaire sur les thérapies géniques

Discussion sur les implications et les développements récents dans le domaine des thérapies géniques. Les sujets abordés incluront la technologie CRISPR, les vecteurs viraux et les applications cliniques des thérapies géniques.

Places restantes : 50

10/05/2024 10:30

Inscription

Formation sur les nouvelles normes de sécurité des médicaments

Formation sur les dernières normes et réglementations de sécurité des médicaments. Cette formation fournira des informations sur les essais cliniques, la surveillance post-commercialisation et les bonnes pratiques de fabrication.

Places restantes : 50

15/05/2024 13:00

Inscription

Table ronde sur les maladies cardiovasculaires

Échange d'idées entre experts sur les traitements et les préventions des maladies cardiovasculaires. Cette table ronde discutera des avancées en matière de médicaments, de la modification du mode de vie et des défis dans la prise en charge des maladies cardiovasculaires.

Places restantes : 50

05/06/2024 11:00

Inscription

Retour à l'Accueil

[Formulaire d'Inscription](#) : Permet aux utilisateurs de s'inscrire à une activité.



Inscription

Conférence sur les avancées en oncologie

Nom :

Prénom :

Email :

[S'inscrire](#)

[Retour aux activités](#)

[Mention Juridique](#) : Accéder à tous nos fichiers juridique



Fichiers juridique - Téléchargements

[Base de donnée](#)

[Conditions D'utilisations](#)

[Charte d'Utilisation des Cookies](#)

[Charte de confidentialité](#)

[Droit a l'image](#)

[Sauvegarde de la base de donnée](#)

[Retour à l'Accueil](#)

Structure BD mise à jour :

MLD :

Medicaments = (id INT, nom VARCHAR(255), date_creation DATE, laboratoire_createur VARCHAR(255), dose_max_journaliere VARCHAR(50));

Liste_effets_therapeutiques = (id INT, effet TEXT);

Liste_effets_secondaires = (id INT, effet TEXT);

Activites = (id INT, nom VARCHAR(255), description TEXT, date_heure DATETIME, nombre_places INT);

Utilisateurs = (email VARCHAR(255), nom VARCHAR(255), prenom VARCHAR(255));

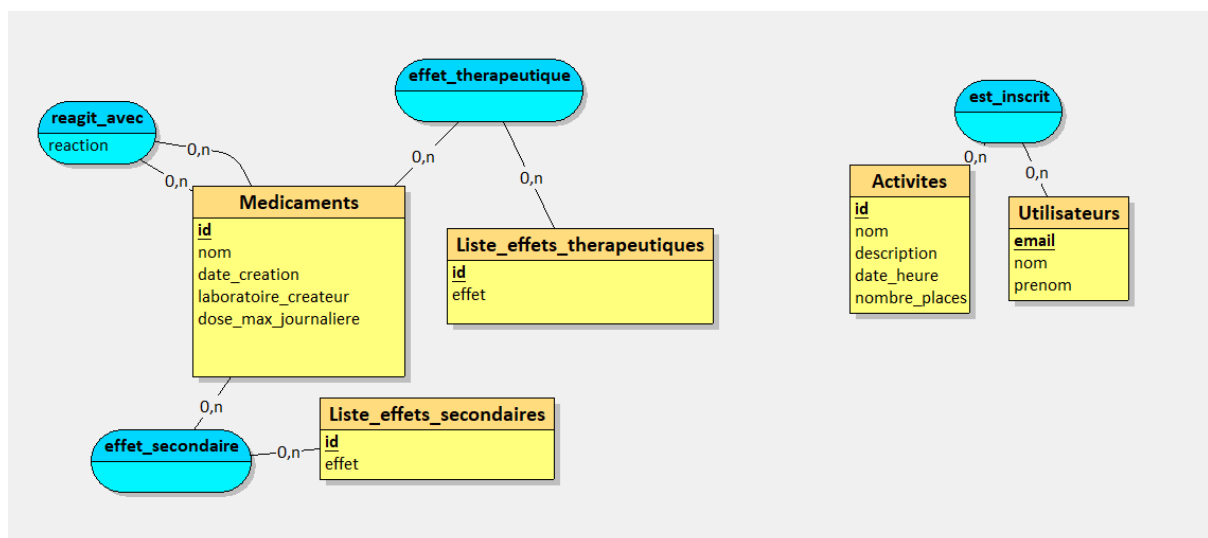
reagit_avec = (#id_medicament_1, #id_medicament_2, reaction TEXT);

effet_therapeutique = (#id_medicament, #id_effet);

effet_secondaire = (#id_medicament, #id_effet);

est_inscrit = (#id_activite, #id_utilisateur);

MCD



Script SQL :

```
CREATE TABLE Medicaments(  
  id INT,  
  nom VARCHAR(255),  
  date_creation DATE,  
  laboratoire_createur VARCHAR(255),  
  dose_max_journaliere VARCHAR(50),  
  PRIMARY KEY(id)  
);  
  
CREATE TABLE Liste_effets_therapeutiques(  
  id INT,  
  effet TEXT,  
  PRIMARY KEY(id)  
);  
  
CREATE TABLE Liste_effets_secondaires(  
  id INT,  
  effet TEXT,  
  PRIMARY KEY(id)  
);  
  
CREATE TABLE Activites(  
  id INT,  
  nom VARCHAR(255),  
  description TEXT,  
  date_heure DATETIME,  
  nombre_places INT,  
  PRIMARY KEY(id)  
);  
  
CREATE TABLE Utilisateurs(  
  email VARCHAR(255),  
  nom VARCHAR(255),  
  prenom VARCHAR(255),  
  PRIMARY KEY(email)  
);  
  
CREATE TABLE reagit_avec(  
  id INT,  
  id_1 INT,  
  reaction TEXT,  
  PRIMARY KEY(id, id_1),  
  FOREIGN KEY(id) REFERENCES Medicaments(id),  
  FOREIGN KEY(id_1) REFERENCES Medicaments(id)  
);  
  
CREATE TABLE effet_therapeutique(  
  id INT,
```

```

id_1 INT,
PRIMARY KEY(id, id_1),
FOREIGN KEY(id) REFERENCES Medicaments(id),
FOREIGN KEY(id_1) REFERENCES Liste_effets_therapeutiques(id)
);

```

```

CREATE TABLE effet_secondaire(
id INT,
id_1 INT,
PRIMARY KEY(id, id_1),
FOREIGN KEY(id) REFERENCES Medicaments(id),
FOREIGN KEY(id_1) REFERENCES Liste_effets_secondaires(id)
);

```

```

CREATE TABLE est_inscrit(
id INT,
email VARCHAR(255),
PRIMARY KEY(id, email),
FOREIGN KEY(id) REFERENCES Activites(id),
FOREIGN KEY(email) REFERENCES Utilisateurs(email)
);

```

Triggers (3) :

Ajouter une place lorsqu'un utilisateur de désinscrit :

```

CREATE TRIGGER `desinscription_activite` AFTER DELETE ON `est_inscrit` FOR EACH ROW BEGIN
UPDATE Activites
SET nombre_places = nombre_places + 1
WHERE id = OLD.id_activite;
END

```

Enlever une place lorsqu'un utilisateur s'inscrit

```

CREATE TRIGGER `inscription_activite` AFTER INSERT ON `est_inscrit` FOR EACH ROW BEGIN
UPDATE Activites
SET nombre_places = nombre_places - 1
WHERE id = NEW.id_activite;
END

```

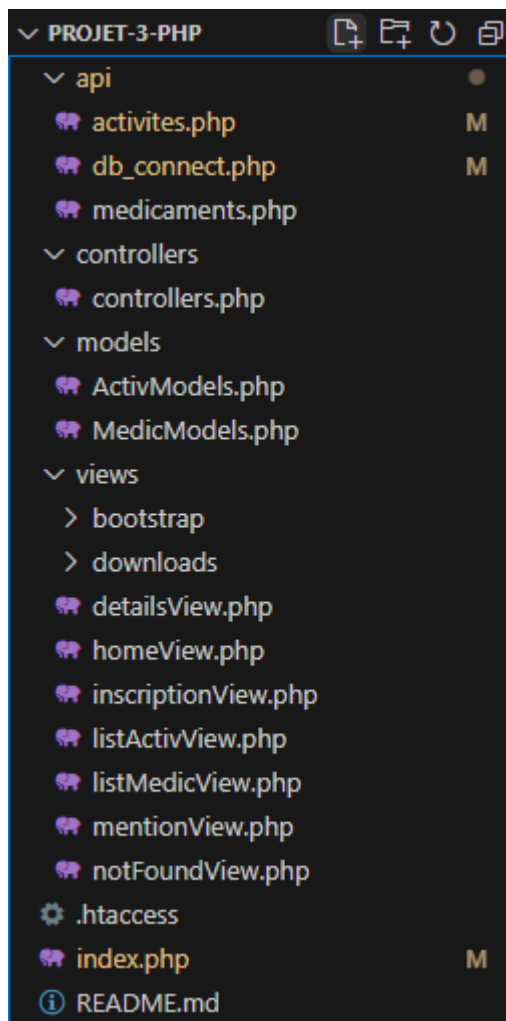
Supprimer toutes les inscriptions d'un utilisateur lorsqu'un est supprimé de la table utilisateurs

```

CREATE TRIGGER `suppression_utilisateur` AFTER DELETE ON `utilisateurs` FOR EACH ROW BEGIN
DELETE FROM est_inscrit WHERE email = OLD.email;
END

```


Structure applicative : Arborescence de notre projet



Script de sauvegarde de la BD : Retrouvez notre script de sauvegarde dans les fichiers joints car celui-ci est trop volumineux pour être retranscrit ici

Charte graphique : Retrouvez notre Charte Graphique dans les fichiers joints car celui-ci est trop volumineux pour être retranscrit ici

Cahier de suivie agile :

Tâches à faire :

créer BD :

- MCD
- MLD
- Implémentation SQL sur WAMP
- Insertion des données
- Exportation script de sauvegarde

créer MVC :

- **Modèle Médicament (API) : Requête SQL**
 - . Requête noms médicaments
 - . Requête détails médicaments
 - . Requête effets secondaires médicaments
 - . Requête Interactions autres médicaments
- **Modèle Activités (API) : Requête SQL**
 - . Requête liste évènements
 - . Requête insérer inscription
 - . Requête détails évènements
- **Vues :**
 - . Accueil
 - . Liste medicament,
 - . click infos médicament,
 - . page activité s'inscrire,
 - . mention juridique
- **Contrôleur :**

Charte graphique : (Code couleur, police d'écriture, logo utilisable . . .)

Mention juridiques :

- [droit à l'image](#)
- [Charte de confidentialité](#)
- [identité de l'entreprise](#)
- [adresse du siège social](#)
- [Charte d'utilisation des cookies](#)
- [condition d'utilisations du site](#)

Tableau de progression

	A faire	En cours	Finis
Séance 1	Sprints Finale Sprints Vérification	Sprints Dev	Sprints Init
Séance 2	Sprints Finale	Sprints Vérification	Sprints Init Sprints Dév
Séance 3		Sprints Vérification Sprints Finale	Sprints Init Sprints Dév
Séance 4			Sprints Init Sprints Dév Sprints Vérification Sprints Finale

Sprints Init : 1 Séances

- BackLog (Descriptif des tâches à accomplir)
- Graphique chart Down (Estimation de la durée nécessaires)
- Estimation du MVC
- Création de la structure MVC
- Estimation des besoins juridique
- Estimation des sprints

Sprints Développement : 2 séances

- créer BD
- Vue (MVC)
- Modèle Médicament (MVC)
- Modèle Activités (MVC)
- Contrôleur (MVC)

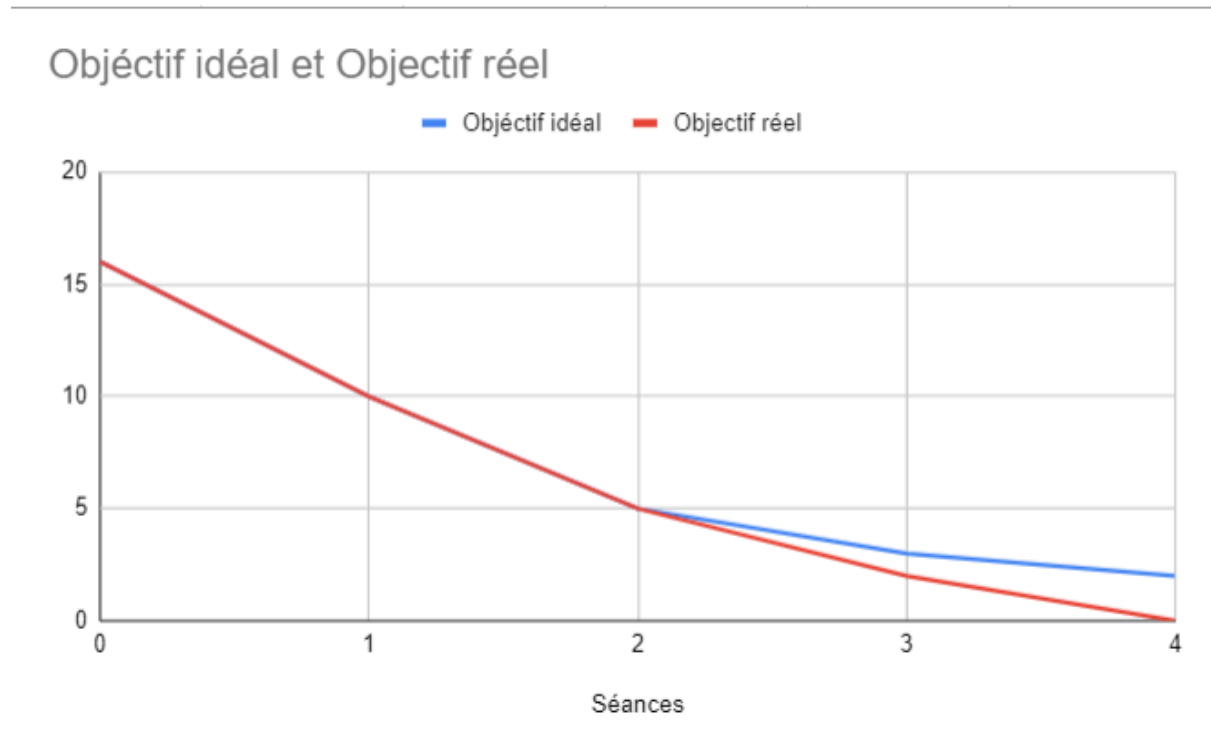
Sprints Vérification : 0.5 séances

- Test du programme
- Vérification de la BD
- Annotations du code

Sprints Finale : 0.5 séances

- Mention juridiques
- Charte Graphique

Chart Down :



Etude juridique : mentions obligatoires à mentionner sur le site :

<https://flourence.fr/projet-3-php/legal>

Code :

detailsViews.php

```
<!DOCTYPE html>

<html lang="fr">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <link rel="icon" href="views\downloads\Logo_Colore.png"
type="image/png">

    <title>Détails du Médicament</title>

    <link href="views/bootstrap/css/bootstrap.min.css"
rel="stylesheet">

</head>

<body>

    <div class="container mt-5">

        <div class="card">

            <div class="card-header">

                <h3 class="mb-0">Détails du Médicament</h3>

            </div>

            <div class="card-body">

                <h5>Nom du médicament</h5>

                <!-- Afficher les données du médicament -->

                <p class="lead"><?= $medicament->medicament[0]->nom;
?></p>

                <h5>Date de Création</h5>

                <p class="lead"><?=
$medicament->medicament[0]->date_creation; ?></p>
```

```

        <h5>Laboratoire Créateur</h5>

        <p class="lead"><?=
$medicament->medicament[0]->laboratoire_createur; ?></p>

        <h5>Dose journalière maximale</h5>

        <p class="lead"><?=
$medicament->medicament[0]->dose_max_journaliere; ?></p>

        <h5>Effets Thérapeutiques</h5>

        <ul>

            <!-- Boucle qui parcourt tous les effets
thérapeutiques -->

            <?php foreach($medicament->effets_therapeutiques as
$effet_therapeutique): ?>

                <li class="lead"><?=
$effet_therapeutique->effets_therapeutiques; ?></li>

            <?php endforeach; ?>

        </ul>

        <h5>Effets Secondaires</h5>

        <ul>

            <!-- Boucle qui parcourt tous les effets
secondaires -->

            <?php foreach($medicament->effets_secondaires as
$effet_secondaire): ?>

                <li class="lead"><?=
$effet_secondaire->effets_secondaires; ?></li>

            <?php endforeach; ?>

        </ul>

```

```

        <h5>Médicaments Réactifs</h5>

        <ul>

            <!-- Boucle qui parcourt tous les médicaments
réactifs -->

            <?php foreach($medicament->reactions as $reaction):
?>

                <li class="lead"><?= $reaction->nom; ?></li>

                <ul>

                    <li class="lead"><?= $reaction->reaction;
?></li>

                </ul>

            <?php endforeach; ?>

        </ul>

    </div>

</div>

<div class="mt-4">

    <a href="medicaments" class="btn btn-primary">Retour à la
liste</a>

</div>

</div>

</body>

</html>

```



```
<!-- Page d'accueil du site permettant d'accéder à tous les services
-->

<!DOCTYPE html>

<html lang="fr">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Accueil GSB : Projet 3 php</title>

    <link rel="stylesheet"
href="views/bootstrap/css/bootstrap.min.css">

    <link rel="icon" href="views\downloads\Logo_Colore.png"
type="image/png">

</head>

<body>

<div class="container mt-5">

    <h1 class="text-center">

        Accueil : Accéder à nos services

    </h1>

    <h3 class="text-center">Nino FEDOU - Mathias HANY - Stefan
FLOURENCE </h3>

    <br> <br>

    <div class="d-flex justify-content-between flex-wrap">
```

```
<div class="card mb-4" style="width: 18rem;">

  <div class="card-body">

    <h5 class="card-title">Liste médicaments</h5>

    <p class="card-text">Consultez aisément notre
répertoire de médicaments.</p>

    <a href="medicaments" class="btn btn-primary">Accédez à
notre liste</a>

  </div>

</div>

<div class="card mb-4" style="width: 18rem;">

  <div class="card-body">

    <h5 class="card-title">Inscriptions aux événements</h5>

    <p class="card-text">Découvrez les événements proposés
par notre société. Profitez d'une inscription gratuite pour rester
informé(e) sur notre actualité et sur les avancées dans le domaine
médical.</p>

    <a href="activites" class="btn btn-primary">Accéder aux
événements</a>

  </div>

</div>

<div class="card mb-4" style="width: 18rem;">

  <div class="card-body">

    <h5 class="card-title">Mention juridique</h5>

    <p class="card-text">Retrouvez l'ensemble de nos
détails juridiques concernant notre site Internet.</p>

    <a href="legal" class="btn btn-primary">Accéder à la
Page</a>

  </div>
```

```

        </div>

    </div>

</div>

</body>

</html>

```

inscriptionView.php

```

<!DOCTYPE html>

<html lang="fr">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Inscription aux Événements</title>

    <link rel="icon" href="views\downloads\Logo_Colore.png"
type="image/png">

    <link rel="stylesheet"
href="views/bootstrap/css/bootstrap.min.css">

</head>

<body>

<div class="container mt-5">

    <h1 class="text-center">

        Inscription

    </h1>

```

```

    <!-- Afficher le nom de l'activité à laquelle l'utilisateur tente
de s'inscrire -->

    <h3><?= $activite[0]->nom; ?></h3>

    <form method="post">

        <input type="hidden" name="idActivite" value=<?=
$activite[0]->id; ?>>

        <div class="form-group">

            <label for="nom">Nom :</label>

            <input type="text" class="form-control" id="nom" name="nom"
placeholder="Entrez votre nom" required>

        </div>

        <div class="form-group">

            <label for="prenom">Prénom :</label>

            <input type="text" class="form-control" id="prenom"
name="prenom" placeholder="Entrez votre prénom" required>

        </div>

        <div class="form-group">

            <label for="email">Email :</label>

            <input type="email" class="form-control" id="email"
name="mail" placeholder="Entrez votre adresse email" required>

        </div>

        <button type="submit" name="action" value="inscription"
class="btn btn-primary">S'inscrire</button>

        <div class="mt-4">

            <a href="activites" class="btn btn-primary">Retour aux
activités</a>

        </div>

```

```

        </form>

</div>

</body>

</html>

```

listActivView.php

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Liste des activités - Projet 3 PHP</title>
    <link rel="icon" href="views/downloads/Logo_Colore.png"
type="image/png">
    <link rel="stylesheet"
href="views/bootstrap/css/bootstrap.min.css">
    <style>
        .card {
            flex: 0 0 calc(33.33% - 1rem);
            margin-right: 1rem;
            margin-bottom: 1rem;
        }
        .card-body {
            min-height: 200px;
        }
        @media (max-width: 992px) {
            .card {
                flex: 0 0 calc(50% - 1rem);
            }
        }
        @media (max-width: 576px) {
            .card {
                flex: 0 0 100%;
            }
        }
    </style>

```

```

</head>
<body>

<div class="container mt-5">

    <!-- Récupérer le paramètre GET "status" pour déterminer si
    l'utilisateur vient d'essayer de s'inscrire à une activité -->
    <!-- Si le paramètre est "error", afficher un message d'erreur -->
    <!-- Si le paramètre est "success", afficher un message de succès -->
    <?php if (isset($_GET['status'])): ?>
        <?php if ($_GET['status'] === 'error'): ?>
            <div class="text-center mt-3">
                <div class="alert alert-danger d-inline-block"
role="alert">
                    <strong>Erreur :</strong> Vous êtes déjà inscrits à
cette activité
                </div>
            </div>
        <?php endif; ?>
        <?php if ($_GET['status'] === 'success'): ?>
            <div class="text-center mt-3">
                <div class="alert alert-success d-inline-block"
role="alert">
                    <strong>Succès :</strong> Vous vous êtes inscrits à
cette activité
                </div>
            </div>
        <?php endif; ?>
    <?php endif; ?>

    <h1 class="text-center">
        
        Liste des activités
    </h1>
    <br> <br>

    <div class="d-flex flex-wrap justify-content-between">
        <!-- Boucle qui parcourt toutes les activités récupérés dans le
        modèle et les affiche -->
        <?php foreach ($activites as $activite): ?>
            <div class="card">
                <div class="card-body">

```

```

        <h5 class="card-title"><?= $activite->nom; ?></h5>
        <p class="card-text"><?= $activite->description; ?></p>
        <p class="card-text">Places restantes : <?=
$activite->nombre_places; ?></p>
        <p class="card-text"><?php echo date('d/m/Y H:i',
strtotime($activite->date_heure)); ?></p>
        <a href="inscription?activite=<?= $activite->id; ?>"
class="btn btn-primary">Inscription</a>
    </div>
</div>
<?php endforeach; ?>
</div>

<div class="mt-4">
    <a href="home" class="btn btn-primary">Retour à l'Accueil</a>
</div>

</div>
</body>
</html>

```

listMedicView.php

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Liste des Médicaments</title>
    <link rel="icon" href="views\downloads\Logo_Colore.png"
type="image/png">
    <link href="views/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>

<div class="container mt-5">
    <h1 class="text-center">
        
        Liste des Médicaments
    </h1>

```

```

        <!-- Création d'un tableau affichant les informations principales
des médicaments -->
        <table class="table mt-4">
            <thead>
                <tr>
                    <th scope="col">Nom</th>
                    <th scope="col">Date de Création</th>
                    <th scope="col">Laboratoire Créateur</th>
                    <th scope="col">Détails</th>
                </tr>
            </thead>
            <tbody>
                <!-- Boucle qui parcourt tous les médicaments récupérés
dans le modèle et les ajoute au tableau -->
                <?php foreach($medicaments as $medicament): ?>
                    <tr>
                        <td><?= $medicament->nom ?></td>
                        <td><?= $medicament->date_creation ?></td>
                        <td><?= $medicament->laboratoire_createur ?></td>
                        <td><a href="details?id=<?= $medicament->id ?>"
class="btn btn-primary">Détails</a></td>
                    </tr>
                <?php endforeach; ?>
            </tbody>
        </table>
        <div class="mt-4">
            <a href="home" class="btn btn-primary">Retour à l'Accueil</a>
        </div>
    </div>

</body>
</html>

```

Page d'erreur : **notFoundView.php**

```

<!-- Page d'erreur 404 -->

<!DOCTYPE html>
<html lang="fr">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

```



```

<title>Page introuvable</title>

<link rel="stylesheet"
href="views/bootstrap/css/bootstrap.min.css">
</head>

<body class="bg-light">

    <div class="container text-center mt-5">

        <h1 class="display-1 text-primary">404</h1>

        <p class="lead">Désolé, la page que vous recherchez est
introuvable.</p>

        <a class="btn btn-primary" href="home">Retour à la page
d'accueil</a>

    </div>

</body>
</html>

```

mentionView.php

```

<!-- Page mentions légales permettant de télécharger les documents
légaux -->
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Téléchargements</title>
    <link rel="icon" href="views\downloads\Logo_Colore.png"
type="image/png">
    <link href="views/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>

<div class="container mt-5">
    <h1 class="text-center">
        

```

```

        Fichiers juridique - Téléchargements
    </h1>

    <ul class="list-group mt-4">
        <li class="list-group-item"><a href="views/downloads/MLD_SCRIPT_SQL.pdf" download>Base de donnée</a></li>
        <li class="list-group-item"><a href="views/downloads/Conditions_d_utilisation.pdf" download>Conditions D'utilisations</a></li>
        <li class="list-group-item"><a href="views/downloads/Charte_d_Utilisation_des_Cookies.pdf" download>Charte d'Utilisation des Cookies</a></li>
        <li class="list-group-item"><a href="views/downloads/Charte_de_confidentialité.pdf" download>Charte de confidentialité</a></li>
        <li class="list-group-item"><a href="views/downloads/Droit_a_l_image.pdf" download>Droit a l'image</a></li>
        <li class="list-group-item"><a href="views/downloads/sauvegardeBD.sql" download>Sauvegarde de la base de donnée</a></li>
        <li class="list-group-item"><a href="views/downloads/Charte_Graphique.pdf" download>Charte graphique</a></li>
    </ul>

    <div class="mt-4">
        <a href="home" class="btn btn-primary">Retour à l'Accueil</a>
    </div>

</div>

</body>
</html>

```

index.php

```

<?php

//Importation des modeles et des controleurs
require_once "models/ActivModels.php";

```

```
require_once "models/MedicModels.php";
require_once "controllers/controllers.php";

//Détecter une tentative d'inscription à une activité
if(isset($_POST['action'])){
    if($_POST['action'] === "inscription"){
        inscriptionActivite();
    }

    exit();
}

//Récupérer la page que l'utilisateur souhaite afficher
if(isset($_GET["action"])) {

    $action = $_GET["action"];

    switch($action) {
        case "home":
            home();
            break;
        default:
            notFound();
            break;

        case "medicaments":
            afficherListeMedicaments();
            break;

        case "details":
            afficherDetailsMedicament();
            break;

        case "activites":
            afficherListeActivites();
            break;

        case "inscription":
            afficherInscription();
            break;

        case "legal":
            afficherMentions();
    }
}
```

```

        break;
    }

    //Si rien n'a été spécifié, afficher la page d'accueil
} else {
    home();
}

?>

```

ActivModels.php

```

<?php

//Importation des modeles et des controleurs
require_once "models/ActivModels.php";
require_once "models/MedicModels.php";
require_once "controllers/controllers.php";

//Déteçter une tanttve d'inscription à une activité
if(isset($_POST['action'])){
    if($_POST['action'] === "inscription"){
        inscriptionActivite();
    }

    exit();
}

//Récupérer la page que l'utilisateur soiuhaite afficher
if(isset($_GET["action"])) {

    $action = $_GET["action"];

    switch($action) {
        case "home":
            home();
            break;
        default:
            notFound();
            break;

        case "medicaments":

```

```

        afficherListeMedicaments();
        break;

    case "details":
        afficherDetailsMedicament();
        break;

    case "activites":
        afficherListeActivites();
        break;

    case "inscription":
        afficherInscription();
        break;

    case "legal":
        afficherMentions();
        break;
    }

//Si rien n'a été spécifié, afficher la page d'accueil
} else {
    home();
}

?>

```

MedicModels.php

```

<?php

//Fonction pour récupérer les médicaments
function getMedics() {

    //Endpoint de l'API
    $url = "http://localhost/projet-3-php/api/medicaments";

    //Options de la requête
    $option = array(
        "http" => array(
            "method" => "GET",
            "header" => "Content-Type: application/x-www-form-urlencoded\r\n"
        )
    )
}

```

```

    );
    $context = stream_context_create($option);
    $result = file_get_contents($url, false, $context);

    //Retourner le résultat de la requête
    return json_decode($result);
}

//Fonction pour récupérer un médicament spécifique
function getMedic($id){

    //Endpoint de l'API
    $url = "http://localhost/projet-3-php/api/medicaments?id=$id";

    //Options de la requête
    $option = array(
        "http" => array(
            "method" => "GET",
            "header" => "Content-Type: application/x-www-form-urlencoded\r\n"
        )
    );
    $context = stream_context_create($option);
    $result = file_get_contents($url, false, $context);

    //Si le médicament n'existe pas, retourner null
    if (json_decode($result)->message == "invalid_id" || json_decode($result)->message == "not_found") {
        return null;
    }

    //Retourner le résultat de la requête
    return json_decode($result);
}

?>

```

controllers.php

```

<?php

//Affichage de la page d'accueil
function home() {

```

```
require "views/homeView.php";
}

//Affichage de la page d'erreur 404
function notFound() {
    require "views/notFoundView.php";
}

//Affichage de la liste des médicaments
function afficherListeMedicaments() {

    //Récupérer les médicaments depuis le modèle
    $medicaments = getMedics();

    //Afficher la vue en lui envoyant les médicaments
    require_once "views/listMedicView.php";
}

//Affichage des détails d'un médicament
function afficherDetailsMedicament() {

    //Récupérer l'id du médicament en parametre GET
    $id = $_GET["id"];

    //Récupérer le médicament depuis le modèle
    $medicament = getMedic($id);

    //Si le médicament n'existe pas, afficher une erreur 404
    if($medicament == null) {
        notFound();
    } else {
        //Sinon, afficher la vue en lui envoyant les détails du médicament
        require_once "views/detailsView.php";
    }
}

//Affichage de la liste des activités
function afficherListeActivites() {

    //Récupérer les activités depuis le modèle
    $activites = getActivs();
}
```

```

        //Afficher la vue en lui envoyant les activités
        require_once "views/listActivView.php";
    }

//Affichage de la page d'inscription à une activité
function afficherInscription(){

    //Récupérer l'id de l'activité en parametre GET
    $id = $_GET["activite"];

    //Récupérer l'activité depuis le modèle
    $activite = getActiv($id);

    //Si l'activité n'existe pas, afficher une erreur 404
    if($activite == null){
        notFound();
        //Sinon, afficher la vue Inscription en lui envoyant les détails de
l'activité
    } else {
        require_once "views/inscriptionView.php";
    }
}

//Affichage de la page des mentions légales
function afficherMentions(){
    require_once "views/mentionView.php";
}

//Inscrire l'utilisateur a une activité
function inscriptionActivite(){

    //Récupérer les données du formulaire
    $nom = $_POST['nom'];
    $prenom = $_POST['prenom'];
    $mail = $_POST['mail'];
    $idActivite = $_POST['idActivite'];

    //Insérer les données grâce au modèle
    $resultat = inscrireActivite($nom, $prenom, $mail, $idActivite);

    //Rediriger vers la liste des activités avec un message de succès
ou d'erreur

```



```

        if (json_decode($resultat)->status_message ===
"user_successfully_inscribed"){
    header('Location: activites?status=success');
} else {
    header('Location: activites?status=error');
}
}
?>

```

[medicaments.php](#)

```

<?php

//Connexion à la base de données
include("db_connect.php");

$request_method = $_SERVER["REQUEST_METHOD"];

//Déterminer l'action de l'utilisateur
switch($request_method){

    //En cas de requête GET, récupérer les médicaments
    case "GET":
        //Si un ID est spécifié, récupérer le médicament correspondant
        if(isset($_GET["id"]) && $_GET["id"] !== ""){
            $id=intval($_GET["id"]);
            getMedic($id);
        }
        //Sinon, récupérer tous les médicaments
        else{
            getMedics();
        }
        break;
}

//Fonction pour récupérer les médicaments
function getMedics(){
    global $conn;

    $query = "SELECT * FROM medicaments";
    $response = array();

    $conn->query("SET NAMES 'utf8'");
}

```

```

$result = $conn->query($query);
while($row = $result->fetch()){
    $response[] = $row;
}
$result->closeCursor();
header('Content-Type: application/json');
echo json_encode($response);
}

//Fonction pour récupérer un médicament spécifique
function getMedic($id){
    global $conn;

    //Vérifier que l'id est un nombre
    if (!is_numeric($id)) {
        echo json_encode(array("message" => "invalid_id"));
        return;
    }

    //Vérifier que l'id est supérieur à 0
    if($id === 0){
        echo json_encode(array("message" => "not_found"));
        return;
    }

    $queryMedic = "SELECT * FROM medicaments WHERE medicaments.id = :id";

    $responseMedic = array();

    $conn->query("SET NAMES 'utf8'");
    $result = $conn->prepare($queryMedic);

    if ($id != 0) {
        $result->bindParam(':id', $id);
    }

    $result->execute();

    //Vérifier que le médicament existe
    if ($result->rowCount() == 0) {
        echo json_encode(array("message" => "not_found"));
        return;
    }
}

```

```

    }

    while($row = $result->fetch()){
        $responseMedic[] = $row;
    }

    $result->closeCursor();

    //Récupérer les effets thérapeutiques du médicament
    $queryEffetsTherapeutiques = "SELECT
liste_effets_therapeutiques.effet AS effets_therapeutiques FROM
medicaments
    JOIN effet_therapeutique ON effet_therapeutique.id_medicament =
medicaments.id
    JOIN liste_effets_therapeutiques ON
liste_effets_therapeutiques.id = effet_therapeutique.id_effet WHERE
medicaments.id = :id";

    $responseEffetsTherapeutiques = array();

    $result = $conn->prepare($queryEffetsTherapeutiques);

    if ($id != 0) {
        $result->bindParam(':id', $id);
    }

    $result->execute();

    while($row = $result->fetch()){
        $responseEffetsTherapeutiques[] = $row;
    }

    $result->closeCursor();

    //Récupérer les effets secondaires du médicament
    $queryEffetsSecondaires = "SELECT liste_effets_secondaires.effet AS
effets_secondaires FROM medicaments
    JOIN effet_secondaire ON effet_secondaire.id_medicament =
medicaments.id
    JOIN liste_effets_secondaires ON liste_effets_secondaires.id =
effet_secondaire.id_effet WHERE medicaments.id = :id";

    $responseEffetsSecondaires = array();

```

```

$result = $conn->prepare($queryEffetsSecondaires);

if ($id != 0) {
    $result->bindParam(':id', $id);
}

$result->execute();

while($row = $result->fetch()){
    $responseEffetsSecondaires[] = $row;
}

$result->closeCursor();

//Récupérer les réactions du médicament
//On récupère toutes les lignes ou est contenu l'id du médicament
que ce soit dans id_medicament_1 ou id_medicament_2
    $queryReactions = "SELECT DISTINCT Medicaments.nom,
reagit_avec.reaction
                        FROM Medicaments
                        JOIN reagit_avec ON ((Medicaments.id =
reagit_avec.id_medicament_1 OR Medicaments.id =
reagit_avec.id_medicament_2)
                        AND Medicaments.id !=
:id_medicament)
                        WHERE reagit_avec.id_medicament_1 =
:id_medicament OR reagit_avec.id_medicament_2 = :id_medicament";

$responseReactions = array();

$resultReactions = $conn->prepare($queryReactions);
$resultReactions->bindParam(':id_medicament', $id);

if ($id != 0) {
    $resultReactions->bindParam(':id', $id);
}

$resultReactions->execute();

while($row = $resultReactions->fetch()){
    $responseReactions[] = $row;
}

```

```

$resultReactions->closeCursor();

//On lie toutes les données en un seul résultat JSON
$responseData = array(
    'message' => 'success',
    'medicament' => $responseMedic,
    'effets_therapeutiques' => $responseEffetsTherapeutiques,
    'effets_secondaires' => $responseEffetsSecondaires,
    'reactions' => $responseReactions
);

header('Content-Type: application/json');

echo json_encode($responseData);
}

?>

```

[db_connect.php](#)

```

<?php

//Connexion à la base de données
try {
    $servername = "localhost";
    $username = "root";
    $password = "";

    $conn = new PDO("mysql:host=$servername;dbname=projet_php",
$username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Connexion échouée : " . $e->getMessage();
}

?>

```

[activites.php](#)

```

<?php

//Connexion à la base de donnée
include("db_connect.php");

$request_method = $_SERVER["REQUEST_METHOD"];

```

```

//Déterminer l'action de l'utilisateur
switch($request_method){

    //En cas de requête GET, récupérer les activités
    case "GET":
        //Si un ID est spécifié, récupérer l'activité correspondante
        if(isset($_GET["id"]) && $_GET["id"] != ""){
            $id=intval($_GET["id"]);
            getActiv($id);
        }
        //Sinon, récupérer toutes les activités
        else{
            getActivs();
        }
        break;

    //En cas de requête POST, inscrire l'utilisateur à une activité
    case "POST":
        inscrireActivite();
        break;
}

//Fonction pour récupérer les activités
function getActivs(){
    global $conn;

    $query = "SELECT * FROM activites";
    $response = array();

    $conn->query("SET NAMES 'utf8'");
    $result = $conn->query($query);
    while($row = $result->fetch()){
        $response[] = $row;
    }
    $result->closeCursor();
    header('Content-Type: application/json');
    echo json_encode($response);
}

//Fonction pour récupérer une activité spécifique
function getActiv($id){
    global $conn;

```

```

//Vérifier que l'id est un nombre
if (!is_numeric($id)) {
    echo json_encode(array("message" => "invalid_id"));
    return;
}

//Vérifier que l'id est supérieur à 0
if ($id === 0) {
    echo json_encode(array("message" => "not_found"));
    return;
}

$queryActivites = "SELECT * FROM activites WHERE activites.id =
:id";

$responseActivites = array();

$conn->query("SET NAMES 'utf8'");
$result = $conn->prepare($queryActivites);

if ($id != 0) {
    $result->bindParam(':id', $id);
}

$result->execute();

//Vérifier que l'activité existe
if ($result->rowCount() == 0) {
    echo json_encode(array("message" => "not_found"));
    return;
}

while($row = $result->fetch()){
    $responseActivites[] = $row;
}

$result->closeCursor();
header('Content-Type: application/json');

//Renvoyer l'activité
echo json_encode($responseActivites);
}

```

```

//Fonction pour inscrire l'utilisateur à une activité
function inscrireActivite(){
    global $conn;

    //Récupérer les données de l'utilisateur
    $nomU = $_POST["nom"];
    $prenomU = $_POST["prenom"];
    $mailU = $_POST["mail"];
    $idActivite = $_POST["idActivite"];

    $userCheckQuery = "SELECT * FROM utilisateurs WHERE email = '$mailU'";
    $userCheckResult = $conn->query($userCheckQuery);

    //Vérifier si l'utilisateur existe, sinon le créer
    if ($userCheckResult->rowCount() == 0) {
        $insertUserQuery = "INSERT INTO utilisateurs (email, nom, prenom) VALUES ('$mailU', '$nomU', '$prenomU')";
        $conn->query($insertUserQuery);
    }

    $activityCheckQuery = "SELECT * FROM est_inscrit WHERE id_activite = '$idActivite' AND email = '$mailU'";
    $activityCheckResult = $conn->query($activityCheckQuery);

    //Vérifier si l'utilisateur est déjà inscrit à l'activité
    if ($activityCheckResult->rowCount() > 0) {
        $response = array(
            "status" => 0,
            "status_message" => "user_already_inscribed"
        );
        //Sinon, l'inscrire
    } else {
        $insertActivityQuery = "INSERT INTO est_inscrit (id_activite, email) VALUES ('$idActivite', '$mailU')";
        $conn->query($insertActivityQuery);

        $response = array(
            "status" => 1,
            "status_message" => "user_successfully_inscribed"
        );
    }
}

```



```
header('Content-Type: application/json');

echo json_encode($response);
}

?>
```

README.md

Projet 3 : PHP/MVC/API/SQL

Site web de GSB

Règles de programmation imposées :

Application réalisée en php avec structure MVC.

La couche modèle utilisera une API REST pour l'accès à la BD.

Le framework bootstrap sera utilisé pour gérer la structure des pages et rendre le site responsive.

Le projet sera réalisé en binôme en appliquant la méthode agile et en utilisant github ou gitlab.

Fonctionnalités attendues :

Présentation des médicaments fabriqués par GSB avec pour chacun possibilité d'afficher les effets thérapeutiques et les effets secondaires ainsi que les interactions avec d'autres médicaments.

Présentation des activités complémentaires proposées par GSB avec possibilité de s'inscrire à l'une des activités.

Le site comportera les mentions juridiques nécessaires, le droit des images utilisées sera vérifié

Documents à fournir :

Schéma d'enchaînement des écrans,

Structure BD mise à jour,

Structure applicative,

Script de sauvegarde de la BD

Charte graphique

Cahier de suivi Agile

Etude juridique : mentions obligatoires à mentionner sur le site

Calendrier (2 semaines= 4 séances) :

L'application devra être réalisée en 2 semaines. Voici une proposition de planning :

séance 1 : détermination du backlog et des sprints (prévoir un sprint init projet et un sprint finish projet + ...)

séances suivantes : la séance débute par une mêlée, on tient à jour le cahier de suivi Agile