

Projet médicaments PHP

Sommaire:

I - Base de Données	3
II - Cahier de suivi Agile	4
III - Schéma d'enchaînement des écrans	17
IV - Schéma de structure applicative	17
V - Script de sauvegarde de la bd	18

I - Base de Données

MLD:

Medicament = (idMedicament INT, nom VARCHAR(50), effetsSecondaires VARCHAR(250), effetsTherapeutiques VARCHAR(250), description VARCHAR(250));

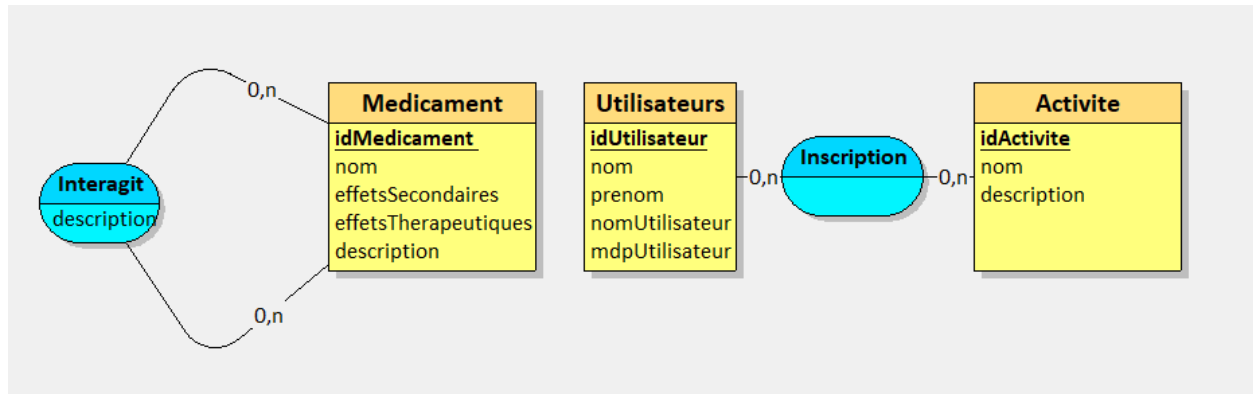
Activite = (idActivite INT, nom VARCHAR(50), description VARCHAR(250));

Utilisateurs = (idUtilisateur INT, nom VARCHAR(50), prenom VARCHAR(50), nomUtilisateur VARCHAR(50) UNIQUE, mdpUtilisateur VARCHAR(50));

Interagit = (#idMedicament, #idMedicament1, description VARCHAR(250));

Inscription = (#idActivite, #idUtilisateur);

MCD:



Commande SQL:

```
CREATE TABLE Medicament(
    idMedicament_ INT,
    nom VARCHAR(50),
    effetsSecondaires VARCHAR(250),
    effetsTherapeutiques VARCHAR(250),
    description_ VARCHAR(250),
    PRIMARY KEY(idMedicament_)
);
```

```
CREATE TABLE Activite(
    idActivite INT,
    nom VARCHAR(50),
    description_ VARCHAR(250),
    PRIMARY KEY(idActivite)
);
```

```
CREATE TABLE Utilisateurs(
    idUtilisateur INT,
    nom VARCHAR(50),
    prenom VARCHAR(50),
    nomUtilisateur VARCHAR(50) UNIQUE,
    mdpUtilisateur VARCHAR(50),
    PRIMARY KEY(idUtilisateur)
);
```

```
CREATE TABLE Interagit(
    idMedicament_ INT,
    idMedicament__1 INT,
    description VARCHAR(250),
    PRIMARY KEY(idMedicament_,
    idMedicament__1),
    FOREIGN KEY(idMedicament_) REFERENCES
    Medicament(idMedicament_),
    FOREIGN KEY(idMedicament__1)
    REFERENCES Medicament(idMedicament_)
);
```

```
CREATE TABLE Inscription(
    idActivite INT,
    idUtilisateur INT,
    PRIMARY KEY(idActivite, idUtilisateur),
    FOREIGN KEY(idActivite) REFERENCES
    Activite(idActivite),
    FOREIGN KEY(idUtilisateur) REFERENCES
    Utilisateurs(idUtilisateur)
);
```

II - Cahier de suivi Agile

ID	Nom	Imp.	Est.	Démo.	Notes
1	Configuration Git/Github	10	1	Montrer le dépôt Git/Github	Vérifier que les droits d'accès sont bien configurés pour les deux membres.
2	Architecture MVC	10	1	Présenter la structure des dossiers (Modèle, Vue, Contrôleur).	S'assurer que chaque dossier est bien organisé et respecte l'architecture.
3	Intégration Bootstrap	7	1	Montrer une page avec une mise en page responsive utilisant Bootstrap.	Vérifier que les composants Bootstrap sont bien intégrés et fonctionnels.
4	Intégration de l'API REST	10	2	Afficher le code complet de l'API REST.	
5	Affichage des médicaments	8	2	Montrer une page listant les médicaments avec leurs noms et descriptions.	Vérifier que le responsive design fonctionne si la liste est longue.
6	Détails des médicaments (effets, interactions)	7	2	Afficher les détails d'un médicament (effets, interactions).	Tester avec plusieurs médicaments pour valider l'affichage des données.
7	Affichage des activités	7	2	Montrer une page listant les activités complémentaires.	S'assurer que les activités sont bien catégorisées (si applicables).
8	Formulaire d'inscription	8	2	Remplir et soumettre un formulaire d'inscription à une activité.	Tester la validation des champs et la confirmation d'inscription.
9	Mentions juridiques	2	1	Montrer une page dédiée aux mentions légales (CGU, politique de confidentialité).	Vérifier que les liens vers les mentions sont accessibles depuis le footer.
10	Charte graphique	3	2	Présenter la charte graphique appliquée (couleurs, polices).	S'assurer que la charte est cohérente sur toutes les pages.
11	Tests et corrections	6	3	Montrer un rapport de tests (fonctionnalités testées, bugs corrigés)	Inclure des tests.

On a décidé de découper les sprints en séances pour faciliter la gestion du temps, ici chaque couleur correspond à une séance

Séance 1

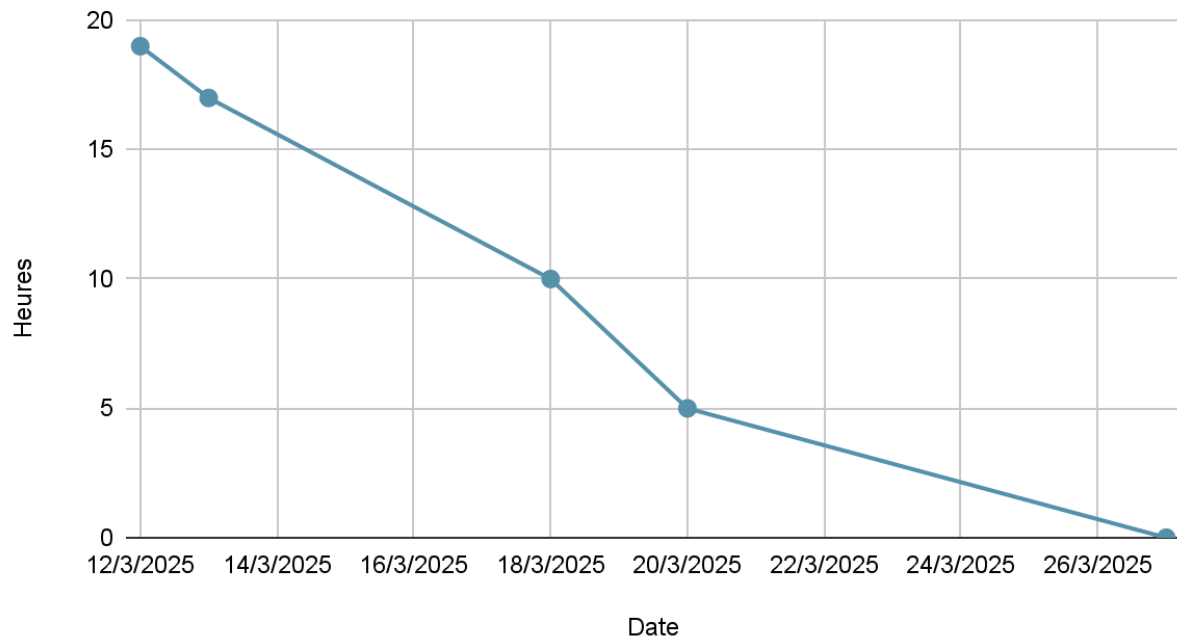
Séance 2

Séance 3

Séance 4

Séance 5

Burn-down chart



Configuration Git/Github

The screenshot shows the GitHub interface for a repository named 'laboGSB'. The repository is private and has 1 watch, 0 forks, and 0 stars. The main branch is 'main'. The repository contains a file named 'index.php' and a directory named 'view'. The repository was created 5 days ago and has 3 commits. The repository is co-authored by 'lebaado' and 'laboGSB'.

laboGSB Private

main 1 Branch 0 Tags

Go to file Add file Code

lebaado Co-authored by: laboGSB 701ac5f · 5 days ago 3 Commits

- api Co-authored by: laboGSB 5 days ago
- controller Co-authored by: laboGSB 5 days ago
- model Co-authored by: laboGSB 5 days ago
- view Co-authored by: laboGSB 5 days ago
- index.php Co-authored by: laboGSB 5 days ago

About

No description, website, or topics provided.

Activity

- 0 stars
- 1 watching
- 0 forks

Releases

No releases published

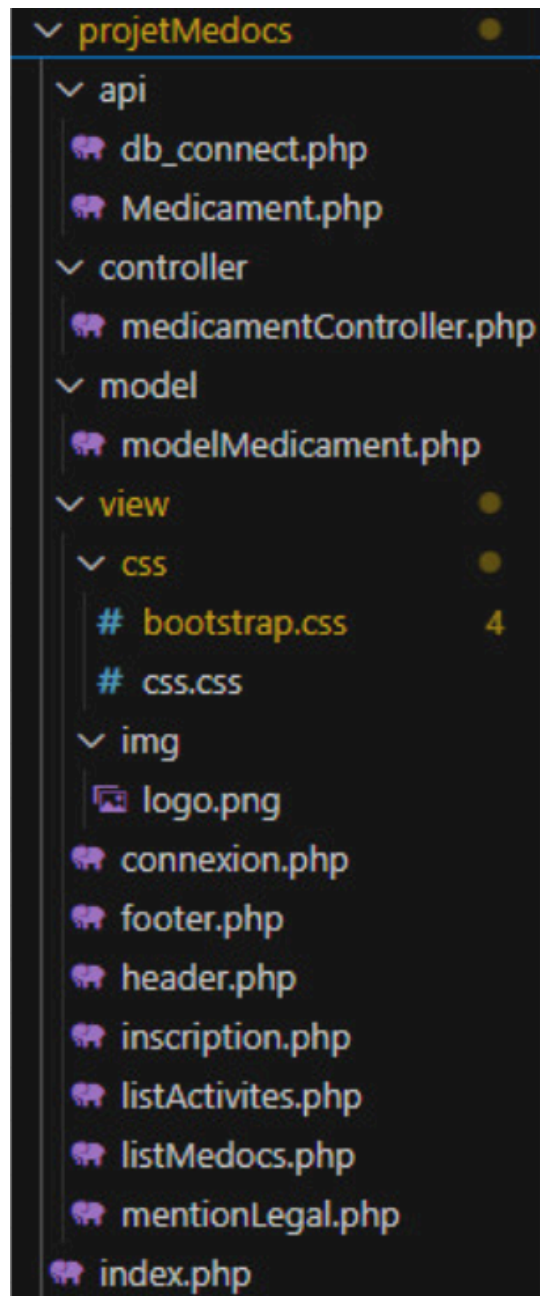
[Create a new release](#)

Contributors 2

The contributors section lists two contributors:

- lebaado** (Abdel Majid)
- laboGSB** (Rémy)

Architecture MVC



Dossier "racine"

Dossier API

Fichier pour connexion à la BD

Fichier API

Dossier Controleur

Fichier controleur

Dossier modèle

Fichier modèle

Dossier vue

Dossier CSS

Fichier bootstrap

Fichier pour style personnel

Dossier image

Image logo

Page Connexion

Page php qui contient le footer

Page php qui contient le header

Page d'inscription

Page des activités

Page des médicaments

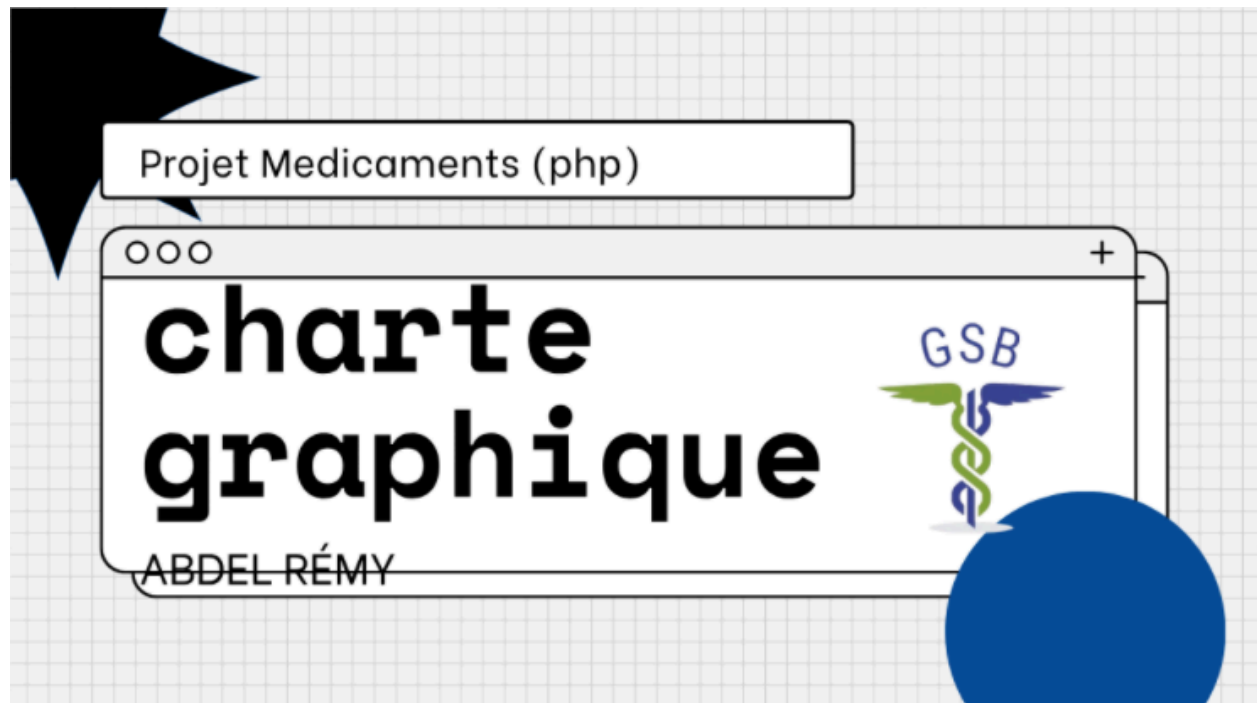
Page Mention Légales

Page index

Nous avons ici une structure MVC2 avec API car on y trouve les parties essentielles du MVC (**Modèle, Contrôleur, Vue**) en plus d'une page **index**, chargée de gérer ce que l'on va voir, ensuite nous y avons ajouté un dossier pour **l'API**.

Charte graphique

La charte graphique est établie pour garantir une identité visuelle cohérente, elle est normalement jointe avec ce document



(Première page de la charte graphique)

Affichage des médicaments & intégration bootstrap

L'affichage de médicaments se fait à l'aide d'un tableau stylisé grâce à bootstrap ce qui permet aussi que le tableau soit responsive.

Nom	Action
Paracétamol	Afficher
Ibuprofène	Afficher
Amoxicilline	Afficher
Aspirine	Afficher
Cetirizine	Afficher

Tableau dans une page de taille normale

Nom	Action
Paracétamol	Afficher
Ibuprofène	Afficher
Amoxicilline	Afficher
Aspirine	Afficher
Cetirizine	Afficher

Dans une page
réduite

Effets et interactions entre les médicaments

Les effets (que ce soit secondaires ou thérapeutiques s'affichent quand on clique sur le bouton afficher du tableau)

Nos Médicaments

Nom	Effets Secondaires	Effets Thérapeutiques
Paracétamol	Nausées, allergies	Antidouleur, antipyrétique

Choisir un médicament :

[Retour](#)

Pour afficher les interactions avec d'autres médicaments, on sélectionne le médicament dans la liste déroulante, puis l'interaction s'affiche

Nos Médicaments

Nom	Effets Secondaires	Effets Thérapeutiques
Paracétamol	Nausées, allergies	Antidouleur, antipyrétique

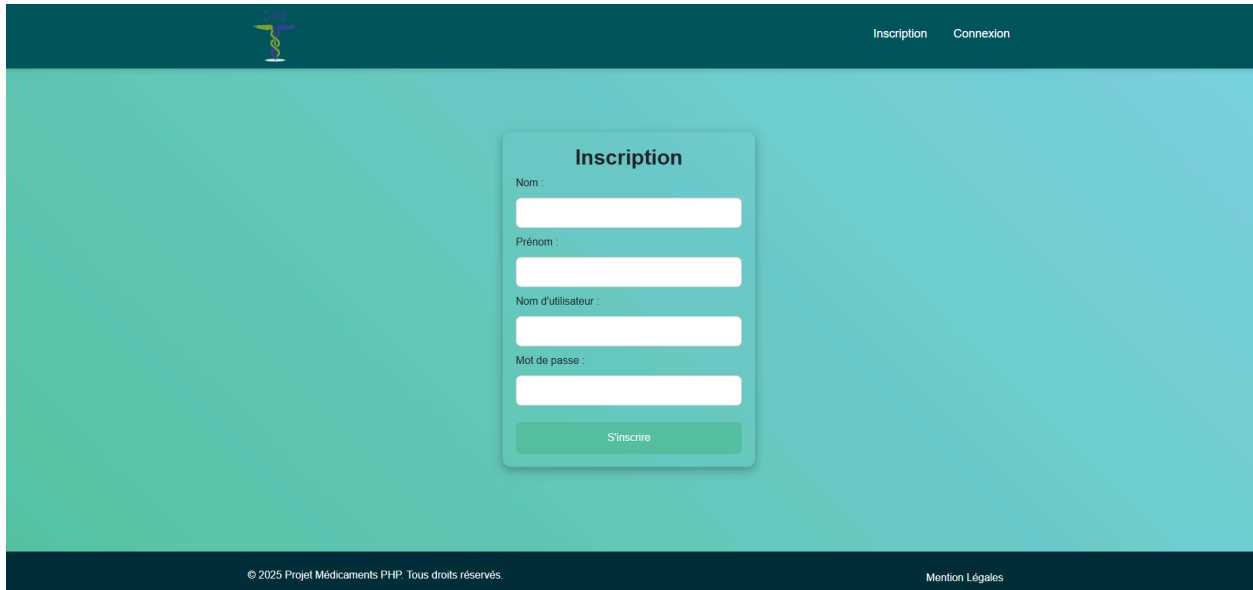
Choisir un médicament :

Interaction : Risque d'ulcères gastro-intestinaux

[Retour](#)

Formulaire d'inscription / connexion

Voici un aperçu des pages pour l'inscription(créer un compte) et de connexion(se connecter a un compte existant)



The screenshot shows a web page with a dark teal header and footer. The header contains a logo on the left and links for 'Inscription' and 'Connexion' on the right. The main content area has a light teal background. In the center is a white rounded rectangle titled 'Inscription'. It contains four input fields: 'Nom', 'Prénom', 'Nom d'utilisateur', and 'Mot de passe'. Below these fields is a green button labeled 'S'inscrire'. The footer contains the copyright notice '© 2025 Projet Médicaments PHP. Tous droits réservés.' and a link for 'Mention Légales'.

Inscription

Nom :

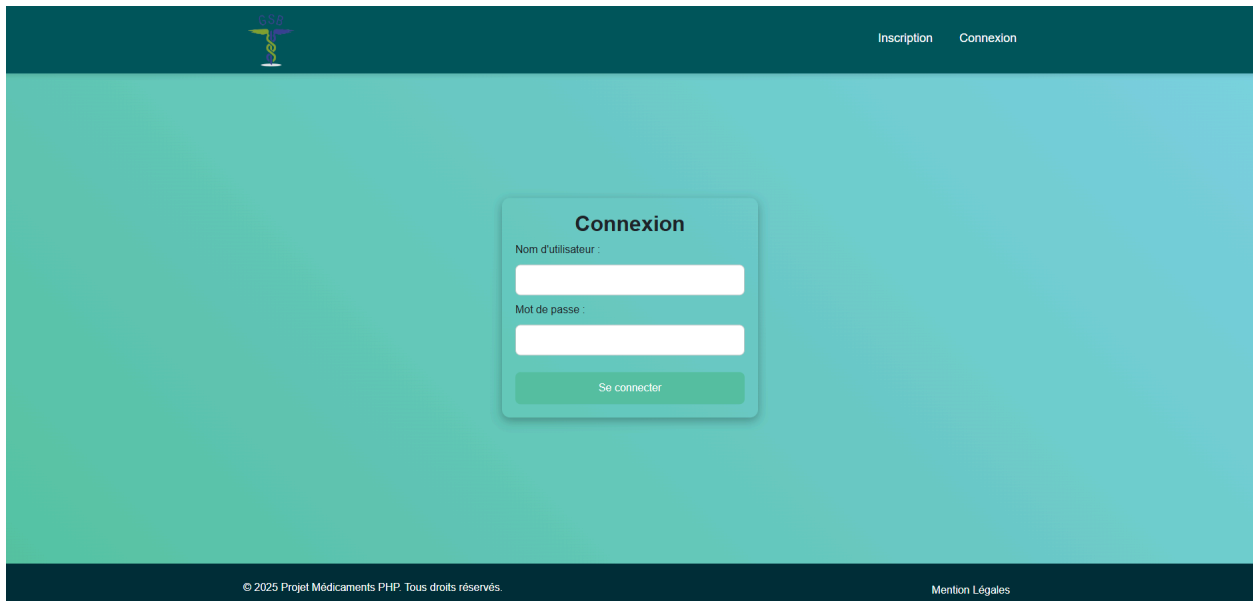
Prénom :

Nom d'utilisateur :

Mot de passe :

[S'inscrire](#)

© 2025 Projet Médicaments PHP. Tous droits réservés. [Mention Légales](#)



The screenshot shows a web page with a dark teal header and footer. The header contains a logo on the left and links for 'Inscription' and 'Connexion' on the right. The main content area has a light teal background. In the center is a white rounded rectangle titled 'Connexion'. It contains two input fields: 'Nom d'utilisateur' and 'Mot de passe'. Below these fields is a green button labeled 'Se connecter'. The footer contains the copyright notice '© 2025 Projet Médicaments PHP. Tous droits réservés.' and a link for 'Mention Légales'.

Connexion

Nom d'utilisateur :


Mot de passe :

[Se connecter](#)

© 2025 Projet Médicaments PHP. Tous droits réservés. [Mention Légales](#)

On a décidé de mettre en place un système de connexion en y ajoutant une sécurisation avec l'intégration d'un salage (hachage) et l'utilisation de htmlspecialchars.

Affichage des activités



[Liste Médicaments](#)[Liste Activités](#)[Se Déconnecter](#)[Supprimer le compte](#)

Bienvenue babab ababa

Nos Activités


Nom	Action
Yoga	Afficher
Randonnée	Afficher
Natation	Afficher
Musculation	Afficher
Danse	Afficher

© 2025 Projet Médicaments PHP. Tous droits réservés.

Mention Légales

Inscription / désinscription à une activité

Voici l’affichage d’une activité à laquelle on est inscrits



[Liste Médicaments](#)[Liste Activités](#)[Se Déconnecter](#)[Supprimer le compte](#)

Nos Activités

Nom	Description	Action
Yoga	Séance de relaxation et d'étirements	Se désinscrire

Retour

© 2025 Projet Médicaments PHP. Tous droits réservés.

Mention Légales

Voici l’affichage d’une activité ou on n’est pas inscrits (ou on se désinscrit)

GSB

Liste Médicaments Liste Activités Se Déconnecter Supprimer le compte

Nos Activités

Nom	Description	Action
Yoga	Séance de relaxation et d'étirements	S'inscrire

[Retour](#)

© 2025 Projet Médicaments PHP. Tous droits réservés. [Mention Légales](#)

Mentions juridiques

On a mis les mentions juridiques dans des “cards” ce qui permet un affichage agréable et optimisé pour le responsive, voici deux aperçus:

Si connecté :

GSB

Liste Médicaments Liste Activités Se Déconnecter Supprimer le compte

Mentions Légales, CGU & Politique de Confidentialité

Dernière mise à jour : 27/03/2025

Éditeur du Site

Ce projet est développé et maintenu par :

- BOUKHLIK Abdel
- SAILLET Rémy

Ce site a pour objectif de fournir des informations sur les médicaments et les activités associées dans un cadre éducatif et non médical.

Hébergement

Le site est hébergé localement dans le cadre d'un projet PHP.

Collecte de Données

Les utilisateurs peuvent s'inscrire via un formulaire qui collecte :

- Nom et prénom
- Nom d'utilisateur
- Mot de passe (haché et salé)

Sécurité

- Mots de passe **hachés (avec salage)**
- Protection contre les injections SQL/XSS
- Utilisation de `htmlspecialchars()`
- Requêtes préparées (PDO)

CGU

- Accès réservé aux utilisateurs inscrits
- Informations à titre informatif seulement
- Pas d'avis médical
- CGU modifiables

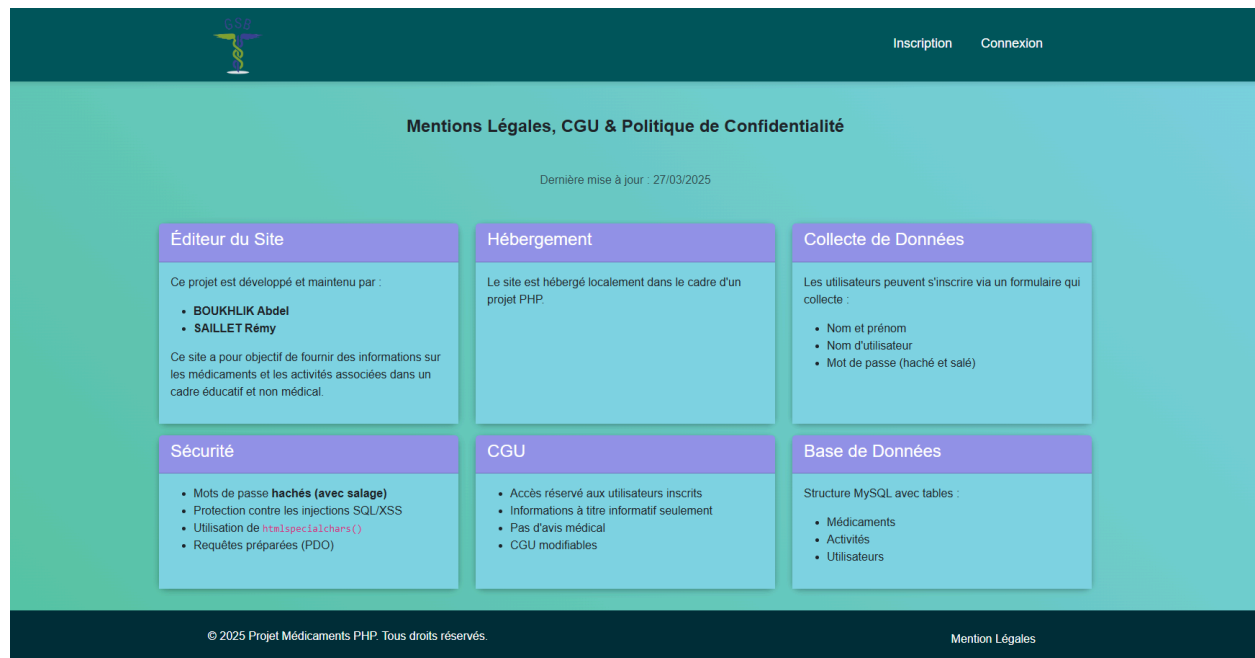
Base de Données

Structure MySQL avec tables :

- Médicaments
- Activités
- Utilisateurs

© 2025 Projet Médicaments PHP. Tous droits réservés. [Mention Légales](#)

Si non connecté :



Intégration de l'API REST

On a gardés l'api pour la fin car celle ci a évoluée tout au long du projet
voici son code

db_connect.php :

```
<?php
$sv_name = 'localhost';
$user = 'root';
$pwd = '';
$conn = new PDO("mysql:host=$sv_name;dbname=bd_labo", $user,$pwd);
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
?>
```

Medicament.php :

```
<?php
include ("db_connect.php");
$http_method = $_SERVER["REQUEST_METHOD"];

switch($http_method)
{
    case 'GET':
        if(!empty($_GET["id"]))
        {
            if (!empty($_GET["id2"]))
            {
```

```

        $id = intval($_GET["id"]);
        $id2 = intval($_GET["id2"]);
        getInteraction($id,$id2);
    }
    else
    {
        $id = intval($_GET["id"]);
        getMedicament($id);
    }
}
else
{
    if(!empty($_GET["nom"]))
    {
        $nom = $_GET["nom"];
        $mdp = $_GET["mdp"];
        getUtilisateur($nom,$mdp);
    }
    else
    {
        if(!empty($_GET["activite"]))
        {
            $id = intval($_GET["idActivite"]);
            getActivite($id);
        }
        else
        {
            if(!empty($_GET["idUtilisateur"]))
            {
                $id = intval($_GET["idUtilisateur"]);
                getActiviteInscrit($id);
            }
            else
            {
                getMedicament();
            }
        }
    }
}
break;
case 'POST':
    if (isset($_GET["ajoutU"]))
    {
        addUtilisateur();
    }
    else
    {
        addActivite();
    }
}

```

```

        break;
    case 'DELETE':
        $idU = intval($_GET["idU"]);
        if (!empty($_GET["id"]))
        {
            $id = intval($_GET["id"]);
            deleteInscription($id,$idU);
        }
        else
        {
            deleteCompte($idU);
        }
        break;
    default:
        header("HTTP/1.0 405 Method Not Allowed");
        break;
}

function getMedicament($id=0)
{
    global $conn;
    if ($id != 0)
    {
        $query = $conn->prepare("SELECT * FROM medicament WHERE
idMedicament = ? LIMIT 1");
        $query->execute([$id]);
    }
    else
    {
        $query = $conn->query("SELECT * FROM medicament");
    }
    echo json_encode($query->fetchAll(PDO::FETCH_ASSOC),
JSON_PRETTY_PRINT);
}

function getActivite($id=0)
{
    global $conn;
    if ($id != 0)
    {
        $query = $conn->prepare("SELECT * FROM activite WHERE idActivite =
? LIMIT 1");
        $query->execute([$id]);
    }
    else
    {
        $query = $conn->query("SELECT * FROM activite");
    }
}

```

```

        echo json_encode($query->fetchAll(PDO::FETCH_ASSOC),
JSON_PRETTY_PRINT);
    }

function getActiviteInscrit($id)
{
    global $conn;
    $query = $conn->prepare("SELECT * FROM inscription WHERE
idUtilisateurs = ?");
    $query->execute([$id]);
    echo json_encode($query->fetchAll(PDO::FETCH_ASSOC),
JSON_PRETTY_PRINT);
}

function getInteraction($id, $id2)
{
    global $conn;
    $query = $conn->prepare("SELECT * FROM interaction WHERE (idMedicament
= ? AND idMedicament_1 = ?) OR (idMedicament = ? AND idMedicament_1 = ?)
LIMIT 1");
    $query->execute([$id, $id2, $id2, $id]);
    echo json_encode($query->fetchAll(PDO::FETCH_ASSOC),
JSON_PRETTY_PRINT);
}

function addUtilisateur()
{
    global $conn;
    try {
        $query = $conn->prepare("INSERT INTO utilisateurs (nom, prenom,
nomUtilisateur, mdpUtilisateur) VALUES (?, ?, ?, ?)");
        $success = $query->execute([$ _POST["nom"], $ _POST["prenom"],
$ _POST["username"], $ _POST["password"]]);
        echo json_encode(['status' => 1, 'status_msg' => 'Utilisateur
ajouté'], JSON_PRETTY_PRINT);
    } catch (PDOException $e) {
        echo json_encode(['status' => 0, 'status_msg' => 'Erreur ajout: '
. $e->getMessage()], JSON_PRETTY_PRINT);
    }
}

function addActivite()
{
    global $conn;
    $query = $conn->prepare("INSERT INTO inscription (idActivite,
idUtilisateurs) VALUES (?, ?)");
    $success = $query->execute([$ _POST["idActivite"],
$ _POST["idUtilisateurs"]]);

```

```

        echo json_encode(['status' => $success ? 1 : 0, 'status_msg' =>
$success ? 'Inscription ajoutée' : 'Erreur ajout'], JSON_PRETTY_PRINT);
    }

function getUtilisateur($nomU, $motDePasseU)
{
    global $conn;
    $query = $conn->prepare("SELECT * FROM utilisateurs WHERE
nomUtilisateur = ?");
    $query->execute([$nomU]);
    $util = $query->fetch(PDO::FETCH_ASSOC);

    if ($util && password_verify($motDePasseU, $util["mdpUtilisateur"]))
    {
        echo json_encode($util, JSON_PRETTY_PRINT);
    }
    else
    {
        echo json_encode(false, JSON_PRETTY_PRINT);
    }
}

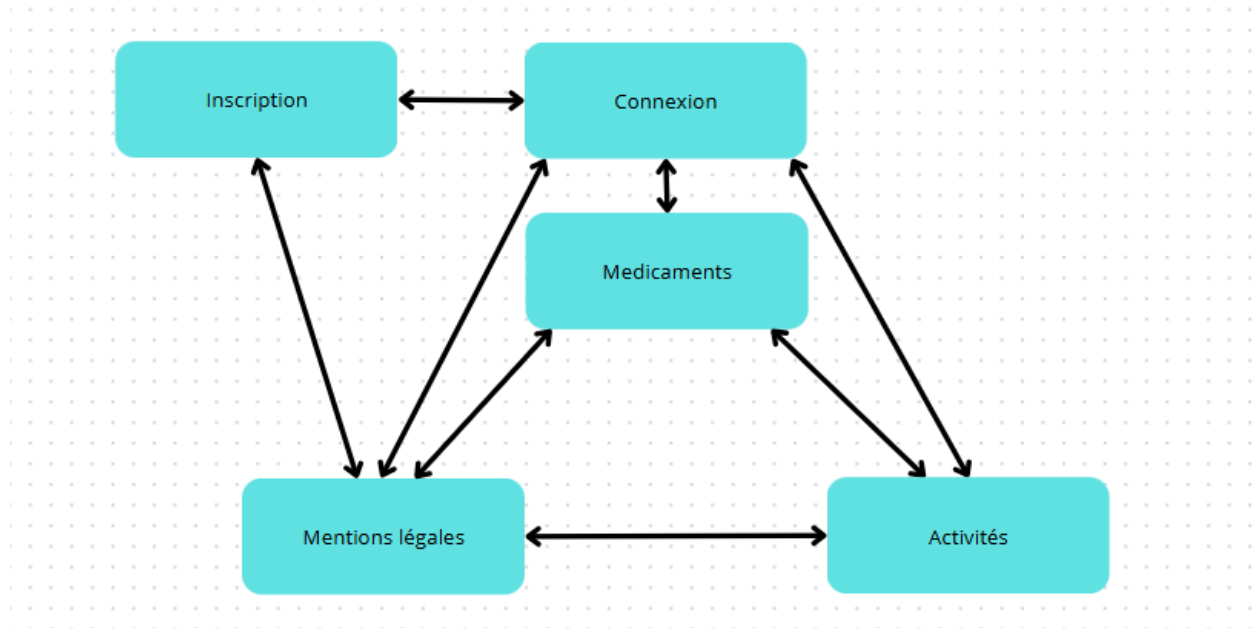
function deleteInscription($id, $idU)
{
    global $conn;
    $query = $conn->prepare("DELETE FROM inscription WHERE idActivite = ?
AND idUtilisateurs = ?");
    $success = $query->execute([$id, $idU]);
    echo json_encode(['status' => $success ? 1 : 0, 'status_msg' =>
$success ? 'Inscription supprimée' : 'Erreur suppression'],
JSON_PRETTY_PRINT);
}

function deleteCompte($idU)
{
    global $conn;
    $query = $conn->prepare("DELETE FROM utilisateurs WHERE idUtilisateurs
= ?");
    $success = $query->execute([$idU]);
    echo json_encode(['status' => $success ? 1 : 0, 'status_msg' =>
$success ? 'Compte supprimé' : 'Erreur suppression'], JSON_PRETTY_PRINT);
}
?>

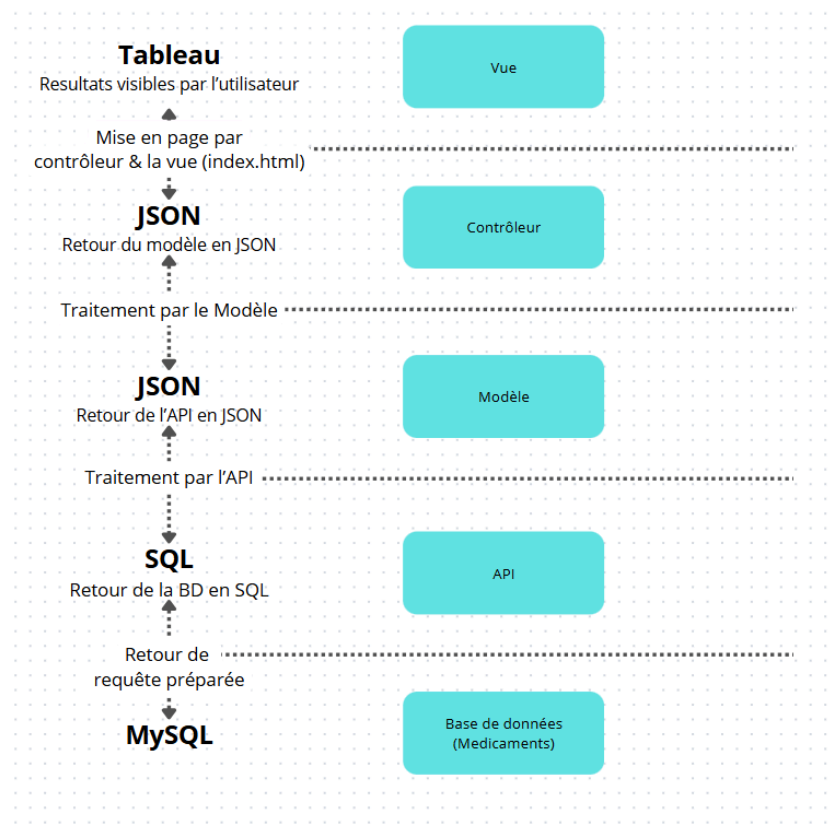
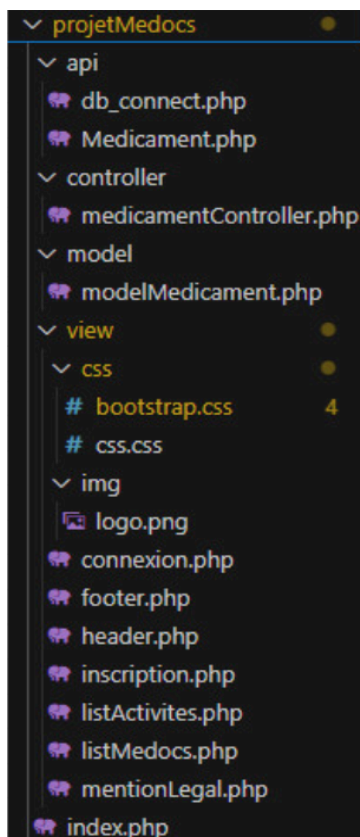
```


III - Schéma d'enchaînement des écrans

Voici notre schéma d'enchaînement des écrans. Grâce au header et au footer implémentés dans le site, il est plus facile de naviguer entre les pages.



IV - Schéma de structure applicative



V - Script de sauvegarde de la bd

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1:3306
-- Generation Time: Mar 27, 2025 at 09:12 AM
-- Server version: 8.3.0
-- PHP Version: 8.2.18

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `bd_labo`
--

--
-- -----
--
-- Table structure for table `activite`
--

DROP TABLE IF EXISTS `activite`;
CREATE TABLE IF NOT EXISTS `activite` (
  `idActivite` int NOT NULL,
  `nom` varchar(50) COLLATE utf8mb4_bin DEFAULT NULL,
  `description` varchar(250) COLLATE utf8mb4_bin DEFAULT NULL,
  PRIMARY KEY (`idActivite`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;

--
-- Dumping data for table `activite`
--

INSERT INTO `activite` (`idActivite`, `nom`, `description`) VALUES
(1, 'Yoga', 'Séance de relaxation et d'étirements'),
(2, 'Randonnée', 'Excursion en montagne pour tous niveaux'),
(3, 'Natation', 'Cours de natation et perfectionnement'),
```

```

(4, 'Musculatation', 'Entraînement en salle avec coach'),
(5, 'Danse', 'Cours de danse moderne et classique');

-- -----

--
-- Table structure for table `inscription`
--

DROP TABLE IF EXISTS `inscription`;
CREATE TABLE IF NOT EXISTS `inscription` (
  `idActivite` int NOT NULL,
  `idUtilisateurs` int NOT NULL,
  PRIMARY KEY (`idActivite`,`idUtilisateurs`),
  KEY `idUtilisateurs` (`idUtilisateurs`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;

-- -----

--
-- Table structure for table `interaction`
--

DROP TABLE IF EXISTS `interaction`;
CREATE TABLE IF NOT EXISTS `interaction` (
  `idMedicament` int NOT NULL,
  `idMedicament_1` int NOT NULL,
  `description` varchar(250) COLLATE utf8mb4_bin DEFAULT NULL,
  PRIMARY KEY (`idMedicament`,`idMedicament_1`),
  KEY `idMedicament_1` (`idMedicament_1`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;

--
-- Dumping data for table `interaction`
--

INSERT INTO `interaction` (`idMedicament`, `idMedicament_1`, `description`)
VALUES
(1, 2, 'Risque d'ulcères gastro-intestinaux'),
(2, 3, 'Diminution de l'efficacité de l'antibiotique'),
(3, 4, 'Augmente le risque de saignement'),
(4, 5, 'Peut augmenter la somnolence'),
(5, 1, 'Possible interaction avec les antihistaminiques');

-- -----

--
-- Table structure for table `medicament`
--

DROP TABLE IF EXISTS `medicament`;

```

```

CREATE TABLE IF NOT EXISTS `medicament` (
  `idMedicament` int NOT NULL,
  `nom` varchar(50) COLLATE utf8mb4_bin NOT NULL,
  `effetsSecondaires` varchar(250) COLLATE utf8mb4_bin DEFAULT NULL,
  `effetsTherapeutiques` varchar(250) COLLATE utf8mb4_bin DEFAULT NULL,
  `Description` varchar(250) COLLATE utf8mb4_bin DEFAULT NULL,
  PRIMARY KEY (`idMedicament`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;

--
-- Dumping data for table `medicament`
--

INSERT INTO `medicament` (`idMedicament`, `nom`, `effetsSecondaires`,
`effetsTherapeutiques`, `Description`) VALUES
(1, 'Paracétamol', 'Nausées, allergies', 'Antidouleur, antipyrétique', 'Utilisé
pour la douleur et la fièvre'),
(2, 'Ibuprofène', 'Maux d'estomac, étourdissements', 'Anti-inflammatoire,
antidouleur', 'Aide à réduire l'inflammation et la douleur'),
(3, 'Amoxicilline', 'Diarrhée, éruptions cutanées', 'Antibiotique', 'Utilisé
pour traiter les infections bactériennes'),
(4, 'Aspirine', 'Troubles digestifs, saignements', 'Antidouleur,
antiplaquettaire', 'Utilisé contre la douleur et la prévention des AVC'),
(5, 'Cetirizine', 'Somnolence, sécheresse buccale', 'Antihistaminique',
'Utilisé contre les allergies');

-- -----
--
-- Table structure for table `utilisateurs`
--

DROP TABLE IF EXISTS `utilisateurs`;
CREATE TABLE IF NOT EXISTS `utilisateurs` (
  `idUtilisateurs` int NOT NULL AUTO_INCREMENT,
  `nom` varchar(50) COLLATE utf8mb4_bin DEFAULT NULL,
  `prenom` varchar(50) COLLATE utf8mb4_bin DEFAULT NULL,
  `nomUtilisateur` varchar(25) COLLATE utf8mb4_bin NOT NULL,
  `mdpUtilisateur` varchar(256) COLLATE utf8mb4_bin NOT NULL,
  PRIMARY KEY (`idUtilisateurs`),
  UNIQUE KEY `nomUtilisateur` (`nomUtilisateur`)
) ENGINE=MyISAM AUTO_INCREMENT=167 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;

--
-- Dumping data for table `utilisateurs`
--

INSERT INTO `utilisateurs` (`idUtilisateurs`, `nom`, `prenom`,
`nomUtilisateur`, `mdpUtilisateur`) VALUES
(159, 'azrae', 'azrzer', 'eee',
'$2y$10$8eGr8NmSJWG0UDxgXLJ6jO1025u9TeTPhgFTStHAWahlIeqLuPOoa'),

```

```

(166, 'ababa', 'ababb', 'abc',
'$2y$10$PaQ9ActQdsYZebYigAM4DOHuS.0cFop18uwJu6t09n.bs/PrFJsFK'),
(165, 'ababa', 'bababa', 'abcd',
'$2y$10$CJ75UK5V7Nk5q36q0ur33O17Dacq3DgBqqMEBf6g4Ukuw1Gs.6wwm'),
(162, 'dazdazd', 'dazdaaz', 'sio2',
'$2y$10$gd0GzwalbCW17JPZt1xhbO2k6OBPDMD045ycR9h9KuzBNd6xbY0vu'),
(164, 'ababa', 'babab', 'abd',
'$2y$10$i12PXBLlh6dBh0x8EP6uMuFhNkQ4V7rYF9M9RnIjww6boq1L4o6TO');

--
-- Triggers `utilisateurs`
--
DROP TRIGGER IF EXISTS `before_user_delete`;
DELIMITER $$
CREATE TRIGGER `before_user_delete` BEFORE DELETE ON `utilisateurs` FOR EACH
ROW BEGIN
    DELETE FROM inscription WHERE idUtilisateurs = OLD.idUtilisateurs;
END
$$
DELIMITER ;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```