1. **L’objectif**

Dans le cadre d’un exercice je devais développer les composants d’accès aux données de type relationnelles.

Cette application web doit conserver des profils de patients et permettre la gestion de ses rendez-vous en ligne.

Les informations clients sont extraite de formulaires puis stocké dans une base de donnée, pour garantir la pérennité de ces informations.

1. **Les étapes**

* Après avoir définit l’objectif, je dessine plusieurs maquettes sur papiers.
* Développer la partie front end, Chaque page disposera d’une barre de navigation pour accéder aux différentes rubriques de mon application.
* Développer la partie back end en PHP.

1. **Développement des composants**

J’utilise le langage PHP qui est un préprocesseur HTML, il fournit en outre une interface (PDO), ensemble de classes pour communiquer avec un système de gestion de base de données (SGBD).

* 1. **Les SGBD**

Ce sont des logiciels servant à stocker, à manipuler ou gérer, et à partager des informations dans une base de données, comporte des mécanismes pour assurer la cohérence des informations, éviter des pertes d'informations dues à des pannes, assurer la confidentialité et permettre son utilisation par d'autres logiciels.

J’ai la possibilité d’utiliser plusieurs SGBD pour la gestion de base de donnée relationnelle : **MySQL**, Microsoft Access, SQLite, j’ai choisi de me servir de MySQL installé avec WAMP.

Le moteur de base de données est le cœur du SGBD (*MyISAM, InnoDB*), mon choix se porte sur **innoDB** car je dispose de 2 table liées *par des clés étrangères*.

* 1. **SQL**

C’est un langage déclaratif normalisé servant à exploiter des bases de données relationnelles.

Il permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles, est reconnu par la grande majorité des SGBD.

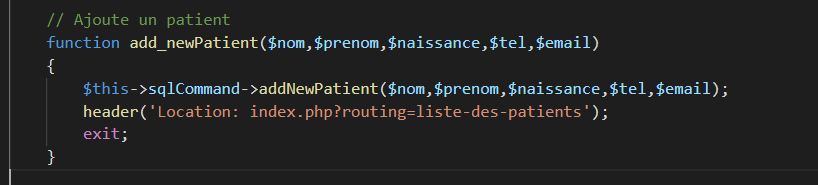
* 1. **PDO** (PHP Data Objects)

C’est une extension PHP qui définit une interface d'accès à une base de données, Son principal avantage est qu'il permet une abstraction pour l'accès aux données. C'est-à-dire que les fonctions pour exécuter des requêtes et pour récupérer des données sont les mêmes, quelque-soit le serveur SQL utilisé (MySQL, PostgreSQL, ...).

Je m’assure que cette extension PHP est activée, via WAMP SERVER.

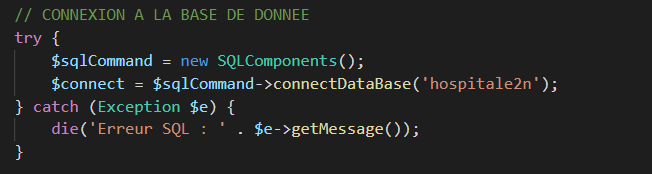
1. **Exemple de code pour l’ajout d’un patient**

Ma première fonction PHP doit ajouter un nouveau patient dans la base de donnée.



* 1. **Connection à la base de données**

J’ai regroupé dans une classe toute les méthodes qui permettent de communiquer avec la base de donnée. Je dois donc instancier cette classe.



* 1. **Manipulation des données (CRUD)**

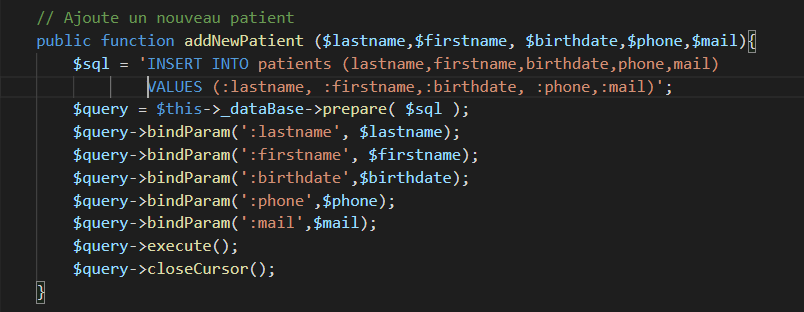
Le CRUD est un acronyme pour désigner les quatre opérations de base pour la persistance des données. Chaque composante de l'acronyme peut être associée à un type de requête SQL, dont voici la liste :

|  |  |  |
| --- | --- | --- |
| **Operations** | **CRUD** | **SQL** |
| Créer | **C**reate | INSERT |
| Lire | **R**ead | SELECT |
| Editer | **U**pdate | UPDATE |
| Supprimer | **D**elete | DELETE |

* 1. **Notions de sécurités (failles ou injection SQL).**

Pour me prémunir des failles de sécurités, en effet l’injection dans une requête en cours d’un morceau de requête non prévu par le système peut compromettre la sécurité et l’intégrité d’un enregistrement d’une table et permettre la suppression de base de données.

Pour cette raison j’utilise une requête préparée.



On retrouve dans cette fonction la notion de CRUD dans la requête SQL avec l’opération INSERT, j’utilise des marqueurs nommés pour éviter les injection SQL puis j’appelle la méthode bindParam pour Lier mes variables PHP au marqueur nommé.