

# Mise en œuvre d'un serveur HTTP sous GNU/Linux

par "pj23dwh"

Remis le 30 octobre 2023

# Table des matières

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>5</b>  |
| 1.1      | Consignes du projet . . . . .   | 5         |
| 1.2      | Contexte du projet . . . . .  | 5         |
| 1.3      | Objectifs du projet et du rapport . . . . .   | 5         |
| <b>2</b> | <b>Fondements théoriques</b>  | <b>7</b>  |
| 2.1      | Qu'est-ce qu'un serveur HTTP ? . . . . .  | 7         |
| 2.2      | Qu'est-ce que GNU/Linux ? . . . . .   | 9         |
| 2.3      | Qu'est-ce qu'Apache ? . . . . .   | 9         |
| 2.4      | Qu'est-ce que sont mySQL & phpMyAdmin ? . . . . .   | 10        |
| 2.5      | Que sont des logiciels "libres" ? . . . . .   | 10        |
| <b>3</b> | <b>Mise en place du serveur HTTP sous GNU/Linux</b>   | <b>11</b> |
| 3.1      | Installation et configuration de l'environnement Linux . . . . .                            | 11        |
| 3.2      | Installation d'Apache2 comme serveur HTTP . . . . .   | 12        |
| <b>4</b> | <b>Gestion de la base de données mySQL</b>  | <b>13</b> |
| 4.1      | Installation et configuration de mySQL et phpMyAdmin . . . . .                              | 13        |
| 4.2      | Création de la base de données et des tables nécessaires . . . . .                          | 14        |
| <b>5</b> | <b>Développement et mise en œuvre du site web</b>   | <b>17</b> |
| 5.1      | Fonctionnalités du site démontrant l'interaction avec la base de données ("CRUD") . . . . . | 17        |
| 5.2      | Création de mon site et exemples d'utilisation d'une base de données . . . . .              | 21        |
| <b>6</b> | <b>Conclusion</b>   | <b>23</b> |
| 6.1      | Conclusion . . . . .  | 23        |
| 6.2      | Ressources . . . . .  | 25        |
| <b>7</b> | <b>Annexes</b>  | <b>27</b> |
| 7.1      | Résumé de mon TM . . . . .  | 27        |
| 7.2      | Captures d'écran supplémentaires . . . . .  | 28        |
| 7.2.1    | Serveur HTTP . . . . .  | 28        |
| 7.2.2    | Pages du Site Web . . . . .   | 29        |
| 7.2.3    | PHPmyAdmin . . . . .  | 34        |

# Table des Illustrations

|     |  |    |
|-----|--|----|
| 2.1 | Schéma d'une requête client-serveur[1] . . . . .   | 7  |
| 2.2 | L'un des centres de serveurs de Google[2]. . . . .   | 8  |
| 4.1 | Installation de phpmyadmin partie 1 . . . . .  | 13 |
| 4.2 | Installation de phpmyadmin partie 2 . . . . .  | 14 |
| 4.3 | Page d'accueil de phpmyadmin . . . . .   | 14 |
| 4.4 | Exemple de tables pour un serveur de vente . . . . .   | 15 |
| 4.5 | Exemple de contenus de table "users" . . . . .   | 15 |
| 5.1 | Code des fonctions "CRUD" (ce code représente une version simplifiée des fonctions afin de gagner en place, nous allons par la suite étudier l'exemple d'une fonction complète). . . . .   | 18 |
| 5.2 | Deuxième partie du code des fonctions "CRUD". . . . .  | 19 |
| 5.3 | Code d'une fonction du "CRUD", gérant les requêtes SQL préparées . . . . .   | 20 |
| 5.4 | Exemple d'utilisation du "CRUD" dans la page /account.php . . . . .  | 22 |
| 7.1 | Cette commande affiche le statut du serveur apache. Il est allumé et fonctionne normalement. . . . .   | 28 |
| 7.2 | Voici l'arborescence des fichiers de ce projet. . . . .  | 28 |
| 7.3 | Voici la page d'accueil du site web. Les deux images présentes sur cette dernière proviennent de la table de données "produits" et redirigent vers les pages de sélection des variantes des produits, le logo, lui, provient du fichier "/var/www/html/images/logo.png". Il est aussi important de noter que depuis cette page, sous la section du header "Mon compte", il est possible d'accéder aux pages d'inscriptions et de connexion du site. Si l'utilisateur est déjà connecté, la page lui proposera de modifier son compte ou de se déconnecter à la place des deux options cités plus haut. . . . . | 29 |
| 7.4 | Dans cette page "account.php", l'utilisateur peut modifier son nom d'utilisateur et son mot de passe. . . . .  | 30 |
| 7.5 | Dans cette page "login/signup.php", l'utilisateur peut s'inscrire dans la base de donnée, puis se connecter au site web. . . . .   | 30 |
| 7.6 | Plusieurs configurations des produits en vente sur le site web sont disponibles. Voici l'option de configuration "Ultra", présente dans la page "product/product1.php" du site. Veuillez noter que chacune de ces configurations sont réellement fonctionnelles, leur compatibilité a été vérifiée à l'aide du site "pcpartpicker". . . . .  | 31 |
| 7.7 | Voici l'option de configuration "Performance", présente dans la page "product/product1.php" du site. . . . .   | 31 |
| 7.8 | Voici l'option de configuration "Apex", présente dans la page "product/product2.php" du site. . . . .  | 32 |
| 7.9 | Voici l'option de configuration "Ecotech", présente dans la page "product/product2.php" du site. . . . .   | 32 |

|  |    |
|--|----|
| 7.10 Une fois le bouton rouge "acheter" préssé dans une des pages de variantes (product/product1 ou /product/product2), l'utilisateur entre les informations permettant la livraison et le paiement, puis valide la commande dans la page "/product/validation.php". Il arrive ensuite, si la commande est validée par le serveur, dans la page "/product/validation.php" qui indique que la commande est confirmée. . . . . | 33 |
| 7.11 Voici la liste des tables de la base de données "siteweb" de mon serveur. Les autres bases de données -visibles sur la gauche de l'écran - sont présentes par défaut . . . . .  | 34 |
| 7.12 Voici la structure de la table "Commandes". Nous pouvons observer que comme pour plusieurs des prochaines tables, elle possède une entrée "id" unique à chaque élément qui est générée automatiquement et une entrée de date (également automatique) enregistrant la date de création de la ligne de donnée. . . . .  | 35 |
| 7.13 Voici le contenu de la table "produits" . . . . .   | 35 |
| 7.14 Voici la structure de la table "produits" . . . . .   | 36 |
| 7.15 Voici le contenu de la table "variantes" . . . . .  | 36 |
| 7.16 Voici la structure de la table "variantes" . . . . .  | 37 |
| 7.17 Voici la structure de la table "users" . . . . .  | 37 |

# Chapitre 1

## Introduction

### 1.1 Consignes du projet

"Ce travail consiste en la mise en œuvre complète et documentée d'un serveur HTTP. La démarche comprendra l'installation d'un serveur HTTP Apache sous GNU/Linux, ainsi que la gestion des permissions d'accès, en vue de mettre en œuvre une plateforme PHP avec une base de données mySQL. Il s'agira de n'utiliser que du logiciel libre. (Ce projet peut être réalisé sur Raspberry Pi)."

- Gymnase de la Cité, 2023

### 1.2 Contexte du projet

Ce travail a été effectué dans le cadre de mon travail de maturité<sup>1</sup> ("TM") au Gymnase de la Cité à Lausanne. Il s'est déroulé durant l'année 2023, il est donc important pour toute personne voulant reproduire ce projet de noter que les logiciels et méthodes utilisés peuvent avoir évolués ou disparus.

### 1.3 Objectifs du projet et du rapport

Le but principal de ce serveur est de répondre à toutes les exigences mentionnées plus tôt dans ce document. Pour ce faire nous devrons :

1. Installer linux sur notre ordinateur ou sur une machine virtuelle (je reviendrai sur ce qu'est une telle machine dans la section 3.1).
2. Installer un logiciel libre permettant de créer un serveur HTTP : "Apache2".
3. Installer et configurer notre base de données avec phpmyadmin et mySQL (je reviendrai sur ce que sont ces programmes dans la section 2.4).
4. Construire un site web permettant de mettre en valeur, de part ses fonctionnalités, la réalisation de notre serveur et de notre base de données.

---

1. Un TM selon la DGEP, la direction générale des études post-obligatoires du canton de Vaud, est un travail courant sur 2 semestres dans lequel "l'élève exerce et démontre son aptitude à chercher, à évaluer, à exploiter et à structurer l'information ainsi qu'à communiquer ses idées[...]."<sup>[3]</sup> Ce travail aboutit à une note pour la dernière année du gymnase qui est l'équivalent suisse du lycée.

Ce rapport a pour but de permettre à tous de réaliser un serveur comprenant une base de données et un site web à partir d'un simple ordinateur et d'une connexion internet. L'avantage de cette réalisation est qu'elle ne vous coûtera rien – mis à part le coût inhérent à votre ordinateur – et que vous aurez le contrôle complet de votre serveur.

# Chapitre 2

## Fondements théoriques

Il est important que tout le monde comprenne bien de quoi nous traitons ici. Veuillez noter que ce chapitre s'adresse aux personnes ne maîtrisant pas encore les outils utilisés dans ce projet ; si ce n'est pas votre cas, veuillez passer directement au chapitre 3.

### 2.1 Qu'est-ce qu'un serveur HTTP ?

Pour créer un serveur HTTP nous devons d'abord répondre aux questions suivantes :

- Qu'est-ce qu'un serveur ?
- Qu'est-ce que HTTP signifie ?

Un serveur est un ordinateur dont le but est de fournir des services ou des ressources à d'autres ordinateurs nommés "client(s)".

Par exemple, quand vous entrez "google.com" dans la barre de recherche de votre navigateur, votre requête atteint un des serveurs de Google et ce dernier calcule en fonction de son code et de ses fichiers stockés ce qu'il doit vous renvoyer comme réponse. Dans le cas d'une requête vers google.com, le serveur vous renverra la page d'accueil de Google.

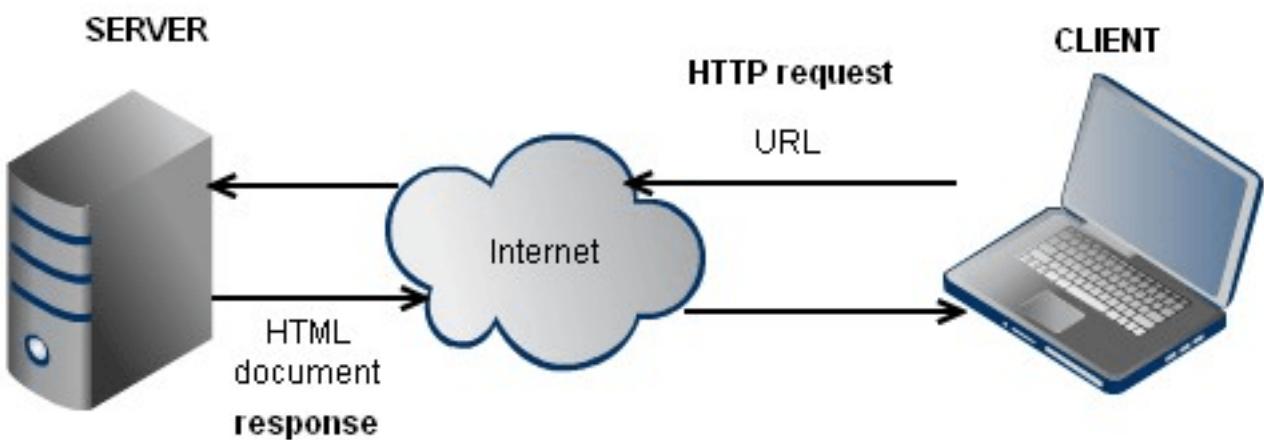


FIGURE 2.1 – Schéma d'une requête client-serveur[1]

Pour créer un serveur, il nous faut donc un ordinateur capable de stocker des fichiers et de calculer suffisamment vite les réponses à renvoyer aux clients. Dans notre cas, quasiment n'importe quel ordinateur pourrait faire l'affaire car le trafic ne sera pas très conséquent. Et non, un serveur n'a pas besoin d'être une grande armoire d'ordinateurs comme chez les géants de la tech ! Vous n'auriez besoin de ce genre d'ordinateurs que dans le cas d'un immense trafic sur votre serveur.

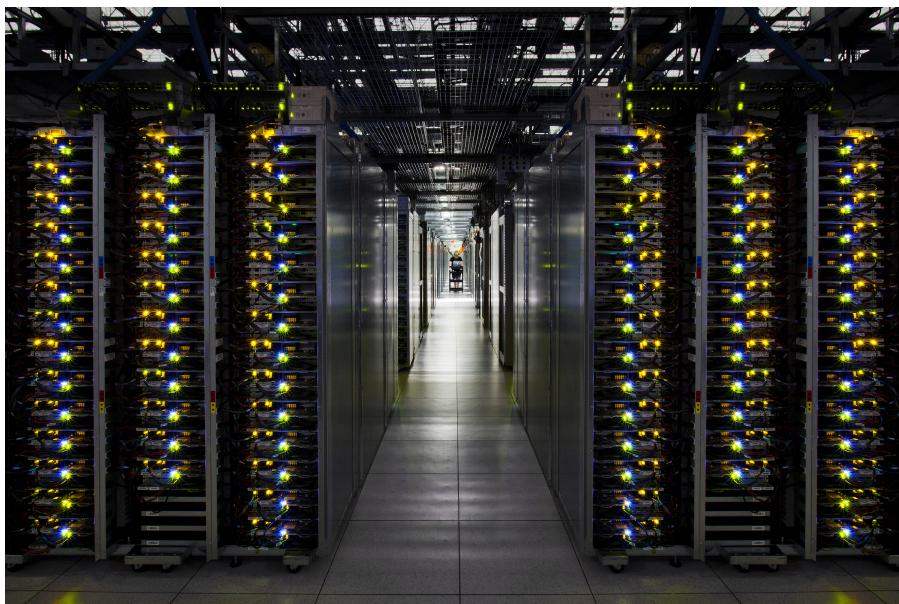


FIGURE 2.2 – L'un des centres de serveurs de Google[2].

Pour ce qui est de l'abréviation "HTTP"<sup>1</sup>, il s'agit là du protocole standard utilisé pour communiquer entre un serveur et un client<sup>2</sup>. Mais vous connaissez certainement mieux le protocole HTTPS – le "S" d'HTTPS signifiant "sécurisé" – qui reprend la même base que HTTP et qui est très présent à travers le web. La principale différence entre ces deux protocoles se trouve dans le fait que HTTPS chiffre les données entre le client et le serveur avant de les transmettre. Cela rend l'échange d'informations plus sécurisé, ce qui est utile en cas de transmission de données sensibles.

Nous nous en tiendrons cependant au protocole HTTP dans ce projet afin de rendre la réalisation de ce dernier plus simple et de respecter au mieux les consignes de ce présent TM.

---

1. "Hyper Text Transfer Protocol"

2. Un client est un logiciel ou programme permettant d'envoyer des requêtes à un serveur. Un navigateur web est par exemple un client.

## 2.2 Qu'est-ce que GNU/Linux ?

Selon le site officiel de GNU/Linux :

"GNU/Linux est un système d'exploitation, une grosse pièce logiciel qui gère un ordinateur. Il est similaire à Microsoft Windows, mais il est entièrement gratuit. Le nom exact est GNU/Linux mais « Linux » est utilisé plus souvent.

GNU/Linux n'est pas le produit d'une entreprise, mais un certain nombre d'entreprises et de groupes de personnes y contribuent. En fait, le système GNU/Linux est un composant central, qui est décliné en de nombreux produits différents. On les appelle des distributions.

Les distributions changent complètement l'apparence et les fonctions de GNU/Linux. Elles vont des grands systèmes complets entièrement pris en charge (approuvés par les entreprises) aux systèmes légers qui tiennent sur une clé USB ou qui fonctionnent sur de vieux ordinateurs (souvent développés par des bénévoles)."<sup>[4]</sup>

GNU/Linux ou "Linux" est donc la base d'une multitude de systèmes d'exploitation (ou "OS" pour "operating system"). Nous aurons besoin de Linux dans le cadre de notre projet de serveur pour deux raisons. Premièrement, n'importe quel serveur est un ordinateur et n'importe quel ordinateur a besoin d'un OS car c'est ce programme-là qui permet d'interagir avec les composants de la machine. Deuxièmement, quitte à devoir choisir un OS, autant en choisir un qui n'embarque pas (trop) de superflus et qui nous laisse beaucoup de liberté sur la gestion de l'ordinateur. Car en effet, la plupart des "grands" OS tels que Windows ou MacOS (pour les Macs) sont livrés avec de nombreuses applications et fonctionnalités inutiles à la réalisation d'un serveur. Celles-ci pourraient même poser de graves problèmes de performance si nous décidions d'installer ces OS sur un ordinateur peu puissant. De plus, ces systèmes d'exploitation sont conçus pour être utilisés par une majorité d'utilisateurs "classiques" ou peu expérimentés. Ils empêchent donc l'utilisation de certaines commandes afin d'éviter qu'un utilisateur qui ne sait pas ce qu'il fait casse accidentellement son système. Seulement, cela devient vite ennuyeux lorsque l'on a besoin d'une liberté totale sur l'ordinateur, comme dans le cas de l'installation d'un serveur.

C'est pourquoi j'ai décidé d'utiliser une version répandue mais simple de Linux : Debian 11. J'ai choisi cette distribution car elle est suffisamment populaire pour avoir l'assurance que les logiciels utilisés auront une version compatible avec elle, mais suffisamment simple pour ne pas s'encombrer de fonctionnalités inutiles qui pourraient ralentir le serveur.

## 2.3 Qu'est-ce qu'Apache ?

Apache, est un logiciel permettant à un ordinateur de prendre le rôle d'un serveur web. Donc, de rendre accessible, depuis le web, l'accès à certaines pages spécifiques de l'ordinateur. Apache fait partie des serveurs web HTTP et HTTPS les plus populaires et les plus fiables. Il permet d'interpréter les requêtes des clients, de récupérer le/les fichier(s) demandé(s) et de renvoyer le tout aux clients afin qu'une page web s'affiche sur leurs navigateurs. Il est efficace, robuste, flexible et compatible sur de multiples OS tels que Linux, Windows ou MacOS. C'est pourquoi de nombreuses entreprises dans le monde utilisent Apache. Ce logiciel est donc le choix parfait pour notre projet de serveur.

## **2.4 Qu'est-ce que sont mySQL & phpMyAdmin ?**

MySQL (my Structured Query Language) est un système de gestion de base de données très populaire dans les serveurs web. Phpmyadmin, quant à lui, permet de gérer ce dernier de manière efficace. Il possède une interface permettant de visualiser la/les base(s) de données du serveur web et d'interagir avec celle(s)-ci. Ensemble, MySQL et phpmyadmin forment des outils puissants très utiles pour créer et utiliser des bases de données. C'est pourquoi nous les utiliserons dans le cadre de la réalisation de notre site web.

## **2.5 Que sont des logiciels "libres" ?**

Un logiciel libre est un logiciel dont le code est accessible au public. Il est modifiable et publiable par les utilisateurs, tout cela légalement. L'un des avantages d'un logiciel libre est que son code est connu. On peut donc avoir confiance en ce que fait le logiciel, puisque ces fonctions sont transparentes. Par exemple, il est possible de vérifier si le logiciel récupère ou non nos données. Un autre avantage des logiciels libres est que grâce à leurs droits de partage et à leur esprit communautaire, l'utilisation de ceux-ci est gratuite.

# Chapitre 3

## Mise en place du serveur HTTP sous GNU/Linux

### 3.1 Installation et configuration de l'environnement Linux

Premièrement, il nous faudra un ordinateur pour réaliser ce serveur. Vous pourriez toute à fait utiliser le vôtre, mais il vous faudra bien penser au fait que l'ordinateur devra être en marche 24 heures sur 24 afin de maintenir le serveur actif. Un raspberry PI<sup>1</sup> ou un vieil ordinateur pourrait donc s'avérer plus approprié en raison de leur faible consommation d'énergie et de leur prix abordable. Mais pour rendre la réalisation de notre serveur plus aisée, j'ai choisi de le réaliser au sein d'une machine virtuelle de Debian11.

Une machine virtuelle est un ordinateur qui est simulé au sein d'un ordinateur. Cela permet de ne rien endommager dans votre système en cas de problème avec la machine virtuelle. De plus, vous pouvez ainsi utiliser des systèmes d'exploitation tels que Linux depuis votre ordinateur habituel (qui est souvent sous Windows ou MacOS). Notez cependant que cette solution n'est pas viable pour un serveur qui doit être accessible en permanence depuis internet. Pour maintenir un serveur sur le net, il est donc recommandé de recourir aux options citées plus tôt. Heureusement, la marche à suivre pour réaliser un serveur est quasiment la même entre les différentes options citées précédemment. Je vous recommande donc de commencer par créer votre serveur sur une machine virtuelle, de le modifier à vos goûts et de vérifier son bon fonctionnement, avant de finalement le déplacer sur un raspberry PI ou un vieil ordinateur.

Pour en revenir à notre marche à suivre, nous allons commencer par installer le logiciel nous permettant d'utiliser des machines virtuelles. Nous utiliserons ici Virtual Box. J'ai pour ma part réalisé cette installation ainsi que la création de ma machine virtuelle Debian11 grâce à cette vidéo : <https://www.youtube.com/watch?v=wX75Z-4MeoM>[5]

Puisque les consignes de ce TM ne portent pas sur l'installation d'une machine virtuelle, je ne vais pas entrer dans le détail de sa mise en place. Mais sachez qu'à l'aide de cette vidéo youtube, vous devriez être en mesure de réaliser une telle opération. Je tiens tout de même à vous rendre attentif au fait qu'une fois votre machine Debian 11 installée, le clavier par défaut de celle-ci ne correspondra pas au clavier français ou suisse. Je vous recommande donc de modifier les paramètres de langue de ce clavier afin d'éviter des erreurs de frappe (qui sont particulièrement problématiques dans l'utilisation des mots de passe).

---

1. Le Raspberry Pi est un ordinateur de la taille d'une carte de crédit qui offre une plateforme abordable et polyvalente pour l'apprentissage, le prototypage et le développement. Il est largement utilisé dans le domaine de l'informatique embarquée.

## 3.2 Installation d'Apache2 comme serveur HTTP

À présent que votre machine virtuelle ou votre ordinateur sous Debian 11 est configuré, nous pouvons commencer l'installation et la configuration d'Apache. Il est important de noter que lorsque nous parlons d'"Apache" dans ce document, nous faisons référence à "Apache2", le successeur d'Apache – premier du nom – qui est aujourd'hui utilisé de la même manière que l'était son prédecesseur. Mais la plupart des utilisateurs utilisent le diminutif "Apache" pour parler d'Apache2. Nous ferons donc de même pour des raisons de simplicité.

Pour réaliser cette installation il vous faut tout d'abord ouvrir votre terminal. Pour ce faire :

- Démarrez votre machine Debian 11
- Rendez-vous sur le bureau de Debian
- Cliquez sur le bouton "Activities" en haut à gauche de votre bureau Debian 11
- Recherchez "terminal" dans la barre de recherche
- Cliquez sur votre terminal
- Mettez à jour votre système avec les commandes "**sudo apt update**" et "**sudo apt upgrade**"<sup>2</sup>

Une fois votre terminal ouvert et à jour, tapez la commande "**sudo apt install Apache2**" afin d'installer Apache2 puis entrez la commande "**sudo systemctl status Apache2**" qui permet de vérifier l'état – "on" ou "off" – du serveur Apache2 fraîchement installé (attention, pour quitter la visualisation de l'état du serveur dans le terminal, vous pourriez avoir besoin d'entrer le raccourci "control + c" qui cesse la tâche en cours du terminal). En cas de problème ou de besoin de précisions supplémentaires, n'hésitez pas à consulter le tutoriel que j'ai moi-même utilisé.[6] Ce dernier est en sources à la fin de ce document.

---

2. Environ une fois par jour, il est important de vérifier si votre système et vos applications sont à jour grâce aux commandes "sudo apt update" et "sudo apt upgrade". Cela permet d'éviter d'éventuels bugs.

# Chapitre 4

## Gestion de la base de données mySQL

### 4.1 Installation et configuration de mySQL et phpMyAdmin

Nous allons maintenant pouvoir procéder à l'installation des 2 logiciels qui nous permettront de créer et de gérer notre base de données : mySQL et phpmyadmin.

Afin d'installer mySQL, il vous faudra entrer les commandes suivantes dans votre terminal : "**sudo apt update**" et "**sudo apt upgrade**" (ce sont deux commandes qui doivent toujours être exécutées avant une installation de façon à éviter d'éventuels problèmes). Après cela, si vous n'avez pas le paquet wget installé par défaut, entrez la commande "**sudo apt install wget**". Pour continuer l'installation de mySQL, tapez "**wget https://dev.mysql.com/get/mysql-apt-config\_0.8.22-1\_all.deb**" et "**sudo apt install ./mysql-apt-config\_0.8.22-1\_all.deb**" (ces deux commandes servent respectivement à télécharger et installer mySQL, cependant, il est important de noter que la version de mySQL à télécharger peut changer avec le temps. Remplacez donc la version notée dans les deux commandes précédentes par la version actuelle trouvable à cette adresse : <https://dev.mysql.com/downloads/repo/apt/>).

Ensuite, exécutez à nouveau les commandes "**sudo apt update**" et "**sudo apt upgrade**" afin de mettre à jour mySQL et enchaînez par "**sudo apt install mysql-server**" .

Une fenêtre devrait à présent s'ouvrir pour vous demander de définir un mot de passe pour l'utilisateur "root" du serveur mySQL. N'oubliez pas de bien noter ce mot de passe. Pour finir l'installation, il vous suffit à présent de choisir les options par défaut (à l'aide des flèches directionnelles et de la touche "enter") lorsque l'installateur vous demandera de choisir vos préférences d'installation.

Maintenant que mySQL est installé, nous pouvons procéder à l'installation de phpmyadmin. Pour cela, commencez par taper "**sudo apt install php phpmyadmin phpmysql**" dans le terminal et acceptez toutes les options par défaut comme suit :

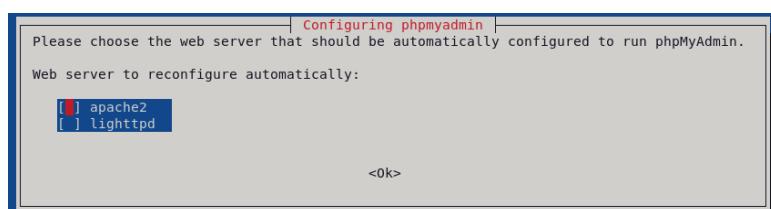


FIGURE 4.1 – Installation de phpmyadmin partie 1

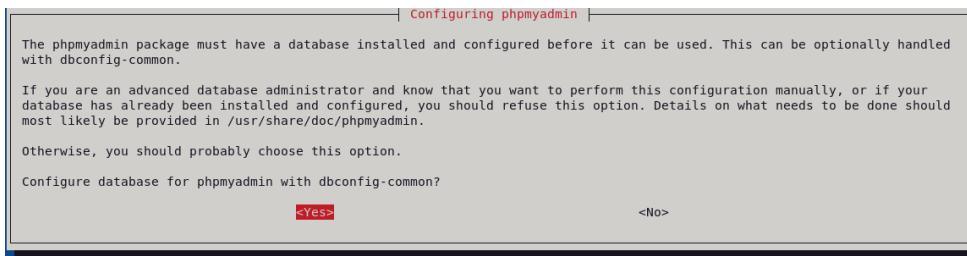


FIGURE 4.2 – Installation de phpmyadmin partie 2

À présent, ouvrez votre navigateur web et entrez dans la barre de recherche du haut (celle avec laquelle on peut taper directement l'url d'un site) "**localhost**". Si tout à bien été configuré, vous devriez pouvoir accéder à la page principale de votre site web. Initialement la page d'accueil par défaut de votre site est celle d'Apache mais vous pourrez remplacer cette page par une page personnalisée dès que vous le souhaiterez.

## 4.2 Création de la base de données et des tables nécessaires

Il est désormais temps de créer la base de données de notre site web. Pour cela, rendez-vous à l'url **localhost/phpmyadmin** et connectez vous à votre compte phpmyadmin avec le nom d'utilisateur et le mot de passe précédemment enregistrés.

Une fois entré dans phpmyadmin, vous devriez avoir une page qui ressemble à peu près à celle-ci :

FIGURE 4.3 – Page d'accueil de phpmyadmin

Sur la gauche de la page, se trouvent les bases de données de notre serveur. En plus de celles pré-installées, nous allons en créer une dédiée à notre site web. Pour ce faire, cliquez sur la case "Databases" en haut de la page et remplissez le champ réservé au nom de la nouvelle base

de données avant de cliquer sur "create". Un bon nom de base de données pourrait être par exemple "siteweb" ; car ce nom est court, sans caractère pouvant poser un problème d'encodage et décrit bien sa propre fonction.

Une fois votre base de données créée, il faudra lui ajouter des "tables" ; une table fonctionne comme un tableau à double entrée<sup>1</sup> qui sert à stocker des informations importantes (c.f. exemples figure 4.4 et 4.5).

FIGURE 4.4 – Exemple de tables pour un serveur de vente

FIGURE 4.5 – Exemple de contenus de table "users"

Comme vous pouvez le voir, la table "users" sert par exemple à stocker les informations concernant les utilisateurs telles que leur mot de passe, leur nom, leur id et autres. Vous pouvez définir à votre guise quelles informations sont stockées dans la base de données afin de convenir au mieux aux besoins de votre site.

Pour ajouter des tables à votre base de données, cliquez sur l'onglet "new" présent à gauche du menu phpmyadmin, dans votre base de données (vous pouvez le voir aux figures 4.4 et 4.5).

Bien que vous puissiez toujours modifier vos tables ou en ajouter de nouvelles, il est important de bien réfléchir à la fonction de votre site web afin de créer des tables utiles et fonctionnelles. En effet, des tables bien réfléchies faciliteront la création de votre site. Par exemple, pour une table utilisateur il est important de créer une colonne "id" (pour identifier un utilisateur unique), une colonne "mot de passe", une colonne "nom d'utilisateur" ou encore "adresse de livraison" dans le cas d'un site marchand. Pour la colonne "id", veillez à bien cocher l'option "auto-increment", afin que la base de données crée un id unique et nouveau pour chaque utilisateur de manière automatique. Pour une colonne image, il est important de changer son type

1. Un tableau à double entrées est une structure de données qui organise des informations en lignes et en colonnes de manière à créer une matrice bidimensionnelle.

en "blob" (binary large object) ou en "longblob", afin que la table puisse allouer une place suffisante à l'image dans la base de données et qu'elle reconnaisse les entrées de cette colonne comme des images et non comme du texte .

J'ai pour ma part commencé par créer une table "user" comprenant les colonnes suivantes :

- id
- nom d'utilisateur
- mot de passe
- date de création du compte

Ainsi qu'une table produit avec les colonnes :

- id
- nom
- caractéristiques techniques
- catégories
- prix
- images

Et pour finir une table commande avec comme colonnes :

- id
- id de l'utilisateur
- date de la commande
- montant de la commande
- nom du client
- adresse de facturation
- adresse de livraison
- etc...

# Chapitre 5

## Développement et mise en œuvre du site web

### 5.1 Fonctionnalités du site démontrant l'interaction avec la base de données ("CRUD")

À présent que notre serveur est prêt et que notre base de données est configurée, nous allons pouvoir terminer notre projet en codant notre site. Pour ce faire, je vous recommande d'installer un éditeur de code tel que Visual Studio Code (vs code) et de vous rendre à l'endroit où le code de votre serveur se trouve. Dans la plupart des cas, le code se situe dans le dossier **/var/www/html** de votre machine. Pour éditer facilement le code présent dans ce dossier, je vous recommande d'ouvrir votre terminal et d'y inscrire une commande telle que "**code /var/www/html**". "Code" servant ici à ouvrir visual studio code et "/var/www/html" spécifiant le dossier dans lequel on souhaite éditer le code. Une fois le dossier contenant le code de votre site web ouvert, commencez par remplacer le fichier index.html par défaut par votre propre fichier index.php (généralement, le fichier "index" est la page qu'un navigateur web affiche par défaut lorsque un utilisateur recherche un site).

Vous vous demandez peut-être pourquoi créer un fichier index.php et non pas un fichier index.html ? L'utilisation du langage php sert à afficher du code html modifié en fonction de scripts. Il permet donc de gérer l'affichage d'une page en fonction de certaines conditions, tandis que le html affiche une page identique pour tous les utilisateurs. L'emploi du php nous permettra donc de nous servir de certaines informations provenant directement de la base de données et d'afficher les pages en fonction de celles-ci. De plus, le php est pratique pour modifier la base de données de part le fait qu'il peut exécuter des scripts côté serveur en fonction des actions de l'utilisateur.

Si besoin de plus de précisions sur les extraits de code suivants, veuillez consulter la page GitHub de ce TM dans laquelle j'ai rendu publique le code de toutes les pages de mon site web. Celle-ci est disponible dans la section ressources de ce document[7].

Pour débuter la réalisation de notre site web, créons une page connexion.php. Dans cette page, nous allons entrer les informations de connexion à la base de données mySQL. Cette page sert à définir la variable \$con (notez bien que toutes les variables php commencent par le symbole "\$") qui permettra à notre futur code php de se connecter à la base de données. La page consiste en cette ligne de code : `<?php $con = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname ?>` Dans cette ligne, les balises php "<?php" et "?>" servent à encadrer le code et

à préciser au serveur qu'il s'agit là de code php. La fonction "\$con" est ensuite définie comme une connexion à la base de données. Elle utilise la fonction mysqli\_connect pour se connecter et les variables \$dbhost, \$dbuser, \$dbpass et \$dbname respectivement pour définir l'adresse de connexion (dans notre cas "Localhost"), le nom d'utilisateur de l'administrateur de la base de données, le mot de passe de cet administrateur et pour finir le nom de la base de données à laquelle nous souhaitons nous connecter (dans notre cas "siteweb"). Il est à présent nécessaire de définir ces variables selon vos propres informations de connexion en suivant cet exemple : <?php \$con = mysqli\_connect("localhost","admin\_user","password1234","siteweb") ?> Pour plus de lisibilité, vous pouvez également conserver les informations de connexion dans la fonction sous forme de variables et définir celles-ci plus haut dans la page.

La dernière étape permettant à la variable \$con d'être définie partout dans le site et donc de pouvoir se connecter à la base de données est de rajouter cette ligne de code au début de chaque page utilisant la variable \$con (je vous recommande d'inscrire cette ligne au début de chaque page php de votre site) : **include("/var/www/html/connexion.php");** Cette ligne spécifie le chemin d'accès absolu du fichier connexion.php et inclut son code dans la page actuelle. Veuillez noter que ce chemin d'accès peut varier selon la configuration de vos fichiers, il est donc important de modifier cette ligne de code si besoin.

Une fois la connexion avec la base de données réalisée, une des premières fonctionnalités que nous devons coder pour notre site est sa capacité à interagir avec celle-ci. Pour cela, nous devons incorporer les fonctions suivantes :

- "Create", afin d'insérer des données dans une table.
- "Read", afin d'obtenir des données depuis une table.
- "Update", afin de modifier les données d'une table.
- "Delete", afin de supprimer certaines données d'une table.

Ces fonctions, vitales pour un serveur possédant une base de données, sont regroupées sous l'appellation "CRUD". Pour réaliser notre "CRUD", nous allons commencer par créer un fichier fonctions.php dans lequel nous pourrons créer les fonctions suivantes de notre site web (voir figure 5.1 & 5.2) :

```
function db_read($con, $table, $colonne, $data)
{
    $query = "select * from $table where $colonne = '$data' limit 1";
    $result = mysqli_query($con, $query);
    if (!$result) {
        echo "Erreur lors de la lecture de la base de donnée : " . mysqli_error($con);
    }
    return $result;
}

3 références
function db_update($con, $table, $colonne, $db_id, $data, $id)
{
    $query = "update $table set $colonne = '$data' WHERE $db_id = '$id'";
    $result = mysqli_query($con, $query);
    if (!$result) {
        echo "Erreur lors de la mise à jour de la base de donnée : " . mysqli_error($con);
    }
}
```

FIGURE 5.1 – Code des fonctions "CRUD" (ce code représente une version simplifiée des fonctions afin de gagner en place, nous allons par la suite étudier l'exemple d'une fonction complète).

```

function db_delete($con, $table, $colonne, $data )
{
    $query = "DELETE FROM $table WHERE $colonne = '$data'";
    $result = mysqli_query($con, $query);
    if (!$result) {
        echo "Erreur lors de la suppression des données : " . mysqli_error($con);
    }
}

1.reference
function db_create($con, $table, $colonne , $data)
{
    $query = "INSERT INTO $table ($colonne) VALUES ('$data')";
    $result = mysqli_query($con, $query);
    if (!$result) {
        echo "Erreur lors de l'insertion des données : " . mysqli_error($con);
    }
}

```

FIGURE 5.2 – Deuxième partie du code des fonctions "CRUD".

Nous pouvons observer plusieurs récurrences dans le code de ces fonctions. Premièrement, la variable \$query contient le code mySQL qui sera envoyé à la base de données, et la fonction mysqli\_query se charge ensuite d'envoyer cette instruction SQL.

Deuxièmement, la fonction s'assure que la base de données ait bien exécutée le code en vérifiant le résultat de celle-ci. Elle affiche ensuite un message d'erreur si nécessaire.

Il est cependant important de noter que la query change selon les fonctions. Par exemple, pour lire les données, nous utilisons SELECT, qui renvoie les résultats correspondants aux conditions qui ont été entrées dans la ligne de code. Ces dites conditions sont spécifiées dans la ligne par :

- "\*" qui signifie que nous souhaitons obtenir toutes les informations d'une ou plusieurs lignes de données
- "from \$table" qui indique dans quelle table la base de données doit chercher l'information
- "where \$colonne = "\$data"" qui précise la valeur *x* de la colonne *y* de la(des ligne(s) d'informations désirée(s) dans la table de données.

Par exemple, si nous cherchons toutes les informations de l'utilisateur "user-a" qui a pour id "5" dans la table de données "users", nous entrerons dans la query : " **SELECT \* from users where id = "5"** ". Une fois le résultat de la fonction obtenu, il est renvoyé avec la fonction "return" vers la page qui a utilisé db\_read.

Les autres fonctions possèdent un fonctionnement similaire. Le principal changement entre les fonctions "CRUD" sont les mots clés utilisés au début de la query : "INSERT into" pour la fonction "créer" (create), "SELECT from" pour la fonction "lire" (read), "UPDATE" pour la fonction "mettre à jour" (update) et "DELETE from" pour la fonction "supprimer" (delete).

Une fois ces quatre fonctions enregistrées, il est temps de les ajouter au site web. Pour ce faire, il suffit d'ajouter la ligne **include("/var/www/html/functions.php");**<sup>1</sup> au début de chaque page du site web.

Avant de conclure ce sous-chapitre, j'aimerais apporter une précision à ce "CRUD". Bien que le code des fonctions affichées plus haut fonctionne dans la plupart des cas, si un utilisateur venait à entrer les caractères <"> ou <'> dans \$data, la requête SQL s'exécuterait mal. En effet, les requêtes mySQL finissent par le symbole <"> ou <'><sup>2</sup>. Dans un cas comme celui

1. Cette ligne spécifie le chemin d'accès absolu du fichier functions.php et inclut ces fonctions dans la page où la ligne de code est insérée. Le chemin d'accès vers votre fichier functions.php peut varier. N'oubliez donc pas de modifier cette ligne de code si besoin.

2. Vous pouvez observer qu'effectivement, toutes les requêtes SQL se terminent dans nos fonctions par un

décrit plus haut, mySQL interprète ainsi le texte qui suit le caractère <"> comme une nouvelle instruction. Cela permet à un attaquant de lancer une injection SQL sur notre base de données<sup>3</sup>. Afin d'éviter les injections SQL, nos fonctions "CRUD" doivent donc être capables de gérer les caractères tels que <"> et <'>. Voici un exemple d'une des fonctions de mon "CRUD" gérant cette problématique. Notez que nous avons précédemment étudié des versions simplifiées des fonctions pour des raisons de place et de clarté. La seule différence entre les fonctions simplifiées et celles de mon site est la gestion des injections SQL.

```
function db_read($con, $table, $colonne, $data)
{
    $query = "SELECT * from $table where $colonne = ? limit 1";
    $stmt = mysqli_prepare($con, $query);

    if ($stmt){
        mysqli_stmt_bind_param($stmt, "s", $data); #liaison des valeurs aux espaces réservés

        #exécution de la requête préparée
        if (mysqli_stmt_execute($stmt)){ #cette ligne vérifie que l'exécution de la requête renvoie true
            $result = mysqli_stmt_get_result($stmt);
            mysqli_stmt_close($stmt); #fermeture de la requête préparée
            return $result;
        }
        else {
            echo "Erreur lors de la lecture de la base de donnée: " . mysqli_error($con);
            mysqli_stmt_close($stmt);
            return false;
        }
    }
    else {
        echo "Erreur lors de la lecture de la base de donnée: " . mysqli_error($con);
        return false;
    }
}
```

FIGURE 5.3 – Code d'une fonction du "CRUD", gérant les requêtes SQL préparées

Dans cet extrait de code (voir figure 5.3), vous pouvez observer que la fonction **mysqli\_query** est remplacée par plusieurs fonctions permettant de séparer la requête SQL de sa/ses donnée(s). De plus, ces fonctions échappent<sup>4</sup> automatiquement les caractères pouvant poser problèmes. Cela signifie qu'il n'est (à priori) plus possible pour un attaquant d'effectuer une injection SQL.

Pour réaliser une requête SQL préparée, la requête se prépare avec des données non définies grâce à la fonction **mysqli\_prepare**. Puis, les valeurs sont liées à leur propre espace réservé dans la requête avec **mysqli\_stmt\_bind\_param** et pour finir, la requête est exécutée avec **mysqli\_stmt\_execute**. Les potentiels messages d'erreurs sont ensuite générés et la requête est fermée avec **mysqli\_stmt\_close** afin de libérer les ressources allouées à celle-ci.

---

<">, le symbole <;> servant non pas à terminer la requête, mais à terminer la ligne de code php

3. une injection SQL étant une attaque consistant à exploiter le fait que SQL interprète tout ce qui est écrit après le <"> pour lui faire exécuter du code malveillant

4. "Échapper" signifie changer l'interprétation d'un caractère. Dans ce cas, les caractères <"> et <'> ne peuvent plus être interprétés comme une fermeture de requête SQL.

## 5.2 Création de mon site et exemples d'utilisation d'une base de données

Afin de vous donner un exemple de ce qu'il est possible de faire avec un serveur, je vais à présent vous expliquer les étapes qui m'ont permis de réaliser mon site.

Tout d'abord j'ai commencé par créer une page index.php afin d'y accueillir mes futurs visiteurs. Ensuite, j'ai suivi le tutoriel "Simple signup and login system with php and mySQL database "[8] afin de mettre au point un système de connexion dans mon site. Cela m'a permis d'apprendre comment interagir avec la base de données, comment le code php fonctionne et de trouver comment débuter mon site. J'ai ensuite décidé que mon site serait marchand. J'ai ainsi créé des produits dans ma base de données et je les ai affichés sur ma page d'accueil. J'en ai également profité pour ajouter un "header" sur toutes les pages ne concernant pas la connexion ou le compte de l'utilisateur. Ce "header" comprend le logo de mon site de vente d'ordinateurs "Pessay"<sup>5</sup>, les contacts de Pessay et des liens vers les pages de connexion ou de gestion du compte utilisateur. J'ai ensuite créé des pages permettant de sélectionner la configuration des types d'ordinateurs en vente sur le site. Ces configurations ont été réalisées à l'aide du site [pcpartpicker.com](http://pcpartpicker.com)[9] qui a permis de vérifier la compatibilité des différents composants proposés. Il a aussi su d'estimer le prix des différents choix d'ordinateurs. Une fois cela fait, j'ai fini mon projet en créant une page de commande dans laquelle l'utilisateur remplit ses informations de paiement et de livraison, puis, une courte page de confirmation de commande.

Toutes ces pages ont en partie pour but de démontrer le bon fonctionnement des fonctionnalités du "CRUD" et donc par extension de la base de données. Nous pouvons par ailleurs observer un bon exemple de synergie entre site web, base de données et fonctions du "CRUD" dans la page servant à modifier le compte d'un utilisateur : **account.php**

### [Voir figure 5.4 à la page suivante]

Dans cet exemple, plusieurs éléments vus précédemment interviennent à tour de rôle. Tout d'abord, \$con (permettant de se connecter à la base de données) et les fonctions du "CRUD" sont présentes grâce à la fonction "include". Puis, le code utilise la fonction check\_login(\$con) qui récupère les informations de l'utilisateur grâce à la fonction "READ" du "CRUD" et à la fonction \$con de la page connexion.php. Avec ces informations, la page s'assure que l'utilisateur est connecté et qu'il a donc le droit d'être sur cette page. Cette utilisation des fonctions "READ" et \$con permet donc d'éviter un accès non-autorisé à cette page.

Un autre exemple de synergie des fonctions liées à la base de données est ce qui se passe quand le bouton de changement de nom d'utilisateur ou de mot de passe est appuyé. La fonction "READ" est utilisée pour vérifier que le nouveau username n'est pas déjà pris, et si ce n'est pas le cas, la fonction "UPDATE" est appelée pour modifier les données de l'utilisateur. Ces deux fonctions (trois avec \$con) ont donc fonctionné ensemble afin de permettre la résolution d'une tâche complexe.

Il y a bien sûr plusieurs autres exemples d'utilisation du "CRUD" dans mon projet de maturation. Cependant, je ne vais malheureusement pas tous vous les citer par manque de pertinence. Je vous rappelle tout de même que vous pouvez consulter mon code source à tout moment via le lien GitHub disponible en bibliographie[7].

---

5. "Pessay" est tiré de la sonorité du mot "PC".

```

<?php
session_start();
include("/var/www/html/login/connexion.php");
include("/var/www/html/login/functions.php");
$user_data = check_login($con);
if ($user_data['id'] == ""){
    header("Location: /index.php");
    die;
}
if ($_SERVER['REQUEST_METHOD'] == 'POST' && !isset($_POST['delete_account']))
{
    $new_username = $_POST["new_username"];
    $new_password = $_POST["new_password"];

    $user_id = $user_data['id'];

    if ($new_username != $user_data['user_name']){
        $result = db_read($con, "users", "user_name", $new_username);
        if ($result && mysqli_num_rows($result) > 0){ # verification de que l'username n'est pas déjà pris
            echo '<p style="color: red; font-size: 17px; position: relative; left:43%;">Ce nom d\'utilisateur est déjà pris !</p>';
        }
        else {
            db_update($con, "users", "user_name", "id", $new_username, $user_id);
            db_update($con, "users", "password", "id", $new_password, $user_id);
            header("Location: index.php");
            die;
        }
    }
    else {
        db_update($con, "users", "user_name", "id", $new_username, $user_id);
        db_update($con, "users", "password", "id", $new_password, $user_id);

        header("Location: index.php");
        die;
    }
}
}

```

FIGURE 5.4 – Exemple d'utilisation du "CRUD" dans la page /account.php

# Chapitre 6

## Conclusion

### 6.1 Conclusion

Pour conclure ce travail de maturité sur la mise en oeuvre d'un serveur http sous GNU/Linux, j'aimerais rappeler les objectifs principaux que nous avons réalisés :

1. Installer linux
2. Installer un logiciel libre permettant de créer un serveur HTTP : "Apache2".
3. Installer et configurer une base de données avec phpmyadmin et mySQL.
4. Construire un site web permettant de mettre en valeur, de part ses fonctionnalités, la réalisation du serveur et de la base de données.

Je tiens à préciser qu'il existe d'autres méthodes et d'autres logiciels permettant de créer un serveur et/ou un site web. J'ai cependant emprunté la route qui fut à mon sens la plus simple pour un utilisateur novice en matière de serveurs, le tout, en restant au plus proche des consignes initiales de ce projet. Il est vrai qu'un utilisateur peu expérimenté pourrait préférer réaliser son serveur et son site web à partir de logiciels tels que Wix ou WordPress, permettant une réalisation plus rapide et plus aisée. Cependant, l'utilisation exclusive de logiciels libres dans ce projet permet de ne dépendre d'aucun tiers. Et le fait d'avoir installé, configuré et codé son propre projet permet une bien meilleure compréhension du site et un contrôle total sur ce dernier. Cela peut s'avérer utile en cas de problème informatique ou en cas d'attentes très précises envers son site. De plus, cela assure qu'aucun tiers ne récupère vos données, chose qui est malheureusement fréquente de nos jours.

Il reste également important de rappeler que le site web qui accompagne ce document pourrait être perpétuellement amélioré. Il pourrait par exemple être intéressant d'y ajouter une fonctionnalité permettant d'ajouter un produit à son "panier", ou encore d'incorporer un vrai système de paiement sécurisé. De nombreux ajouts de ce type pourraient être faits afin de rendre ce projet plus proche d'un vrai site marchand.

Toutefois, le but de ce TM était de mettre en oeuvre un serveur HTTP et une base de données. Mon site web est donc présent pour prouver le bon fonctionnement de ces éléments. J'ai ainsi décidé de me concentrer sur l'implémentation de fonctionnalités démontrant la bonne gestion de la base de données et du "CRUD".

La réalisation de ce TM m'a beaucoup appris de part la gestion de projet mais également de part son aspect technique. Je me suis rapidement rendu compte, au début de ce projet, que j'avais largement sous-estimé le travail qui m'attendait. Car la résolution d'erreurs, la compréhension de nouveaux concepts informatiques et l'apprentissage de nouveaux langages de

programmation prennent un temps considérable. J'ai également dû apprendre à faire des choix dans la réalisation de ce projet et dû donner la priorité à certains éléments comme le "CRUD".

Je pense donc que la création d'un serveur et d'un site web sous GNU/linux en utilisant uniquement des logiciels libres est destinée aux personnes pour qui l'aspect du contrôle total en matière de données, de mise en page et de fonctionnalité est important. J'ajouterais que pour quiconque aimant l'informatique ou la compréhension du fonctionnement des choses en général, ce travail peut être passionnant. Car en plus d'être très enrichissant à étudier, les possibilités qu'apporte un serveur et un site sont presque infinies. Cependant, je ne peux vous recommander de suivre les étapes de ce document si votre but est de réaliser le plus rapidement et simplement possible un site. Ce projet recherche la compréhension et la maîtrise totale des éléments du site, tandis que des options telles que WordPress cherchent le meilleur rapport temps/résultat. Il est donc libre à chacun de décider quelle voie emprunter en fonction de ces informations.

Je souhaite également vous recommander de ne pas hésiter à faire usage d'outils tels que chatGPT ou StackOverflow pour résoudre les potentiels problèmes que vous pourriez encourir dans votre réalisation. En effet, le nombre de problèmes qu'il est possible de rencontrer dans tout projet informatique est immense. Il m'est donc impossible de tous les anticiper. Heureusement, la communauté des développeurs est très active sur internet et apporte souvent de précieuses réponses. De plus, de nombreux outils tels que ceux précédemment cités sont à votre disposition pour vous permettre de comprendre vos erreurs.

Pour conclure, ce TM – bien qu'étant une grande charge de travail – m'a permis d'apprendre énormément. J'ai été surpris de prendre autant de plaisir à coder mon plus grand projet informatique à ce jour et souhaiterai certainement l'améliorer dans le futur mais en suis très fier aujourd'hui. Je recommande à tout gymnasien aimant l'informatique de choisir à leur tour ce sujet de TM. Merci à ma famille et à mes amis pour leur relecture et leur soutien ainsi qu'au hackerspace de Renens "Fixme" pour leur aiguillage sur les premières étapes de mon projet.

**Merci pour votre attention et votre intérêt envers ce TM.**

**Je vous invite à présent à consulter le code de mon site à l'adresse :**

**[https://github.com/pj23dwh/mon\\_serveur](https://github.com/pj23dwh/mon_serveur)**

## 6.2 Ressources

- [1] Lycée Alphonse Benoit. Schéma d'une requête client-server. Source photo de la figure 2.1 ; <https://lycee-benoit.tech/NSI/prem/html/images/client-server.png>.
- [2] CBS News. Exemple photo d'un serveur google, 2012. Source photo de la figure 2.2 ; <https://www.cbsnews.com/news/behind-the-cloud-a-tour-of-googles-secretive-data-facilities/>.
- [3] DGEP. "dgep brochure école de maturité". page 8/107, 2022. [https://www.vd.ch/fileadmin/user\\_upload/organisation/dfj/dgep/dgep\\_fichiers\\_pdf/DGEP\\_brochure\\_EM\\_web.pdf](https://www.vd.ch/fileadmin/user_upload/organisation/dfj/dgep/dgep_fichiers_pdf/DGEP_brochure_EM_web.pdf).
- [4] GNU/Linux. "qu'est-ce que gnu/linux?", 2023. <https://www.getgnulinux.org/fr/linux>.
- [5] NetworkChuck. "you need to learn virtual machines right now!! (kali linux vm, ubuntu, windows)", 2021. Tutoriel utile pour la configuration de la machine virtuelle utilisée dans ce TM disponible à l'adresse <https://www.youtube.com/watch?v=wX75Z-4MEoM>.
- [6] Linode. "apache basics tutorial | how to install and configure apache2", 2021. Vidéo en ligne à l'adresse <https://www.youtube.com/watch?v=1CDxpAzvLKY>.
- [7] Julien Freiburghaus. Repo github 'mon\_serveur', 2023. Le code source de mon site web est disponible à l'adresse suivante : [https://github.com/pj23dwh/mon\\_serveur/](https://github.com/pj23dwh/mon_serveur/). Merci à github d'offrir une plateforme d'échanges et de stockage de code.
- [8] Quick Programming. "simple signup and login system with php and mysql database | full tutorial | how to & source code", 2020. Vidéo très utile pour débuter la création des premières pages de son site web, disponible à l'adresse <https://www.youtube.com/watch?v=WYufSGgaCZ8>.
- [9] PCPartPicker. "[pcpartpicker.com](http://pcpartpicker.com)", 2023. Ce site web a permis de vérifier la compatibilité des configurations d'ordinateurs en vente sur le site web. Il a aussi permis d'estimer le prix de ces configurations.
- [10] Dr. Treffor Bazett. "how to write a thesis using latex \*\*full tutorial\*\*", 2022. Vidéo en ligne à l'adresse <https://www.youtube.com/watch?v=zqQM66uAig0>.
- [11] OpenAI. "chatgpt", 2023. ChatGPT s'est avéré très utile dans la résolution de bugs et dans mon apprentissage de nouveaux concepts.

- [12] OpenClassroom. "créez votre site web avec html5 et css3", 2023. Cours très utile pour pouvoir coder son propre site web. Il est disponible à l'adresse <https://openclassrooms.com/fr/courses/1603881-creez-votre-site-web-avec-html5-et-css3>.
- [13] CompVis Runway and Stability AI. "stablediffusion", 2023. Stablediffusion est une IA générant des images open source. Les images des produits de Pessay et son logo ont été générés par elle.
- [14] Onlinepngtools. "<https://onlinepngtools.com/create-transparent-png>", 2023. Ce site web a permis de rendre le logo de Pessay transparent.

# Chapitre 7

## Annexes

### 7.1 Résumé de mon TM

Ce TM traite de la mise en œuvre d'un serveur HTTP sous GNU/Linux. Pour ce faire, il définit les termes techniques et explicite point par point comment installer linux, comment configurer un serveur HTTP Apache2, comment mettre en place et exploiter avec phpMyAdmin une base de données MySQL, et pour finir, comment réaliser un site web mettant en avant ledit serveur et ladite base de données. Cette réalisation n'utilise que des logiciels libres afin de garder un contrôle et une compréhension maximale sur le rendu final. Le rapport de ce TM apporte plusieurs éléments de compréhension sur ce qu'est un serveur, comment fonctionne une base de données et par où commencer son propre site web. Je recommande donc ce travail à toute personne souhaitant se lancer dans la réalisation de son propre serveur et/ou site web aux moyens de lignes de code.

Pour ce qui est de l'aspect pratique de ce TM, un site web marchand a été réalisé à l'aide de pages de code PHP et CSS. Celui-ci est connecté à une base de données MySQL et contient de nombreux exemple d'utilisation de cette dernière, comme notamment des fonctionnalités de création et de gestion de compte utilisateur ("login") ou encore la possibilité de passer une commande sur le site (ajout de la commande contenant les informations de l'utilisateur dans la base de données).

Ce travail présente une alternative intéressante à des applications tels que Wix ou Wordpress. En effet, la réalisation personnelle et intégrale de son propre serveur et/ou site web permet d'avoir une connaissance totale de ce dernier. De plus, l'utilisation de logiciels "libres" - c'est à dire dont le code source est accessible au public - assure qu'aucun tiers ne peut récupérer les données transitant par le site web. Cela renforce donc l'aspect sécuritaire et privé de la réalisation, chose intéressante dans une société où la vie privée est en voie d'extinction.

## 7.2 Captures d'écran supplémentaires

### 7.2.1 Serveur HTTP

```
apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-11-23 20:50:06 CET; 1min 46s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 545 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 612 (apache2)
    Tasks: 6 (limit: 9475)
   Memory: 21.3M
      CPU: 649ms
CGroup: /system.slice/apache2.service
        ├─612 /usr/sbin/apache2 -k start
        ├─626 /usr/sbin/apache2 -k start
        ├─627 /usr/sbin/apache2 -k start
        ├─628 /usr/sbin/apache2 -k start
        ├─629 /usr/sbin/apache2 -k start
        └─633 /usr/sbin/apache2 -k start

Nov 23 20:50:04 DebianSTM systemd[1]: Starting The Apache HTTP Server...
Nov 23 20:50:06 DebianSTM apachectl[574]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using fe80::a00:27ff:fe:cl88. Set the 'ServerName' directive globally to...
Nov 23 20:50:06 DebianSTM systemd[1]: Started The Apache HTTP Server.

lines 1-20/20 (END)
```

FIGURE 7.1 – Cette commande affiche le statut du serveur apache. Il est allumé et fonctionne normalement.

```
admin@Debian5TM:/var/www/html$ ls -R
.:
account.php  images  index.php  login  mysql-apt-config_0.8.22-1_all.deb  phpmyadmin  product  style.css

./images:
logo2.png  logo3.png  logo4.png  logo5.png  logo.png

./login:
connexion.php  functions.php  login.php  logout.php  signup.php

./product:
commande.php  product1.php  product2.php  product3.php  validation.php
admin@Debian5TM:/var/www/html$
```

FIGURE 7.2 – Voici l’arborescence des fichiers de ce projet.

## 7.2.2 Pages du Site Web



### Bienvenue sur Pessay

Contact:  
Adresse : Avenue Lepage  
Téléphone : 021 404 0

Ceci est un site de démonstration.  
aucune commande ne sera prise en compte.

Mon compte:  
[Me connecter](#)  
[M'inscrire](#)

## Bienvenue sur Pessay, la boutique d'ordinateurs conçus pour vous.

Cliquez sur une catégorie et commencez à choisir votre ordinateur !

Nous nous chargeons de monter vos machines et de les livrer chez vous.



PC Gaming

Performant, dédié aux jeux-vidéos et aux programmes gourmands en ressources.



PC Professionnel

Économique et fiable. Déstiné aux entreprises et aux particuliers qui recherchent le meilleur rapport qualité-prix.

FIGURE 7.3 – Voici la page d'accueil du site web. Les deux images présentes sur cette dernière proviennent de la table de données "produits" et redirigent vers les pages de sélection des variantes des produits, le logo, lui, provient du fichier "/var/www/html/images/logo.png". Il est aussi important de noter que depuis cette page, sous la section du header "Mon compte", il est possible d'accéder aux pages d'inscriptions et de connexion du site. Si l'utilisateur est déjà connecté, la page lui proposera de modifier son compte ou de se déconnecter à la place des deux options citées plus haut.

## Modifier mon compte

Bienvenue AbC123 !

Nouveau nom d'utilisateur:  
AbC123

Nouveau mot de passe:

[Enregistrer les modifications](#)

[Supprimer mon compte](#)

[Retour à l'accueil](#)

FIGURE 7.4 – Dans cette page "account.php", l'utilisateur peut modifier son nom d'utilisateur et son mot de passe.

## Inscription

Nom d'utilisateur

Mot de passe

[Inscription](#)

[Se connecter](#)

FIGURE 7.5 – Dans cette page "login/signup.php", l'utilisateur peut s'inscrire dans la base de donnée, puis se connecter au site web.



## Config Gaming - Ultra

Contact:  
Adresse : Avenue Lepart 404a, Kernel  
Téléphone : 021 404 04 04  
  
Mon compte:  
[Me connecter](#)  
[M'inscrire](#)  
[Retour à l'accueil](#)

Sélectionnez la version de votre ordinateur : Ultra



PC Gaming

Performant, dédié aux jeux-vidéos et aux programmes gourmands en ressources.

**5000.- CHF**

localhost/index.php

FIGURE 7.6 – Plusieurs configurations des produits en vente sur le site web sont disponibles. Voici l'option de configuration "Ultra", présente dans la page "product/product1.php" du site. Veuillez noter que chacune de ces configurations sont réellement fonctionnelles, leur compatibilité a été vérifiée à l'aide du site "pcpartpicker".



## Config Gaming - Performance

Contact:  
Adresse : Avenue Lepart 404a, Kernel  
Téléphone : 021 404 04 04  
  
Mon compte:  
[Me connecter](#)  
[M'inscrire](#)  
[Retour à l'accueil](#)

Sélectionnez la version de votre ordinateur : Performance



PC Gaming

Performant, dédié aux jeux-vidéos et aux programmes gourmands en ressources.

**2740.- CHF**

FIGURE 7.7 – Voici l'option de configuration "Performance", présente dans la page "product/product1.php" du site.



## Config Pro - Apex

Contact:  
Adresse : Avenue Lepart 404a, Kernel  
Téléphone : 021 404 04 04  
[Mon compte](#):  
[Me connecter](#)  
[M'inscrire](#)  
[Retour à l'accueil](#)

Sélectionnez la version de votre ordinateur : Apex ▾ Confirmer



PC Professionel

Économique et fiable. Déstiné aux entreprises et aux particuliers qui recherchent le meilleur rapport qualité-prix.

**1500.- CHF**

Acheter

## Config Pro - EcoTech

Contact:  
Adresse : Avenue Lepart 404a, Kernel  
Téléphone : 021 404 04 04  
[Mon compte](#):  
[Me connecter](#)  
[M'inscrire](#)  
[Retour à l'accueil](#)

Sélectionnez la version de votre ordinateur : EcoTech ▾ Confirmer



PC Professionel

Économique et fiable. Déstiné aux entreprises et aux particuliers qui recherchent le meilleur rapport qualité-prix.

**700.- CHF**

Acheter

FIGURE 7.8 – Voici l'option de configuration "Apex", présente dans la page "product/product2.php" du site.



## Validez Votre Commande:

Contact:  
Adresse : Avenue Lepart 404a, Kernel  
Téléphone : 021 404 04 04

Mon compte:  
[Modifier mon compte](#)  
[Me déconnecter](#)

[Retour à l'accueil](#)



Ordinateur Sélectionné:  
PC Gaming - Ultra

Performant, dédié aux jeux-vidéos et aux programmes gourmands en ressources.

Nom:

Prénom:

Pays de livraison:

Adresse de facturation:

Code postal:

Adresse de livraison\*:

Téléphone:

Adresse Email:

Numéro de Carte:

Date d'expiration:  
 xx/xx

Cryptogramme:

FIGURE 7.10 – Une fois le bouton rouge "acheter" pressé dans une des pages de variantes (product/product1 ou /product/product2), l'utilisateur entre les informations permettant la livraison et le paiement, puis valide la commande dans la page "/product/validation.php". Il arrive ensuite, si la commande est validée par le serveur, dans la page "/product/validation.php" qui indique que la commande est confirmée.

### 7.2.3 PHPmyAdmin

The screenshot shows the phpMyAdmin interface for a MySQL server. The left sidebar lists databases: information\_schema, mysql, mysqldatabase-phpmyadmin, performance\_schema, siteweb, sys, and New. The 'siteweb' database is selected. The main area displays the structure of the 'siteweb' database, including four tables: Commandes, Produits, users, and Variantes. A 'Create table' dialog is open in the bottom right corner, showing fields for Name (empty) and Number of columns (set to 4).

| Table           | Action                                    | Rows      | Type          | Collation                 | Size           | Overhead   |
|-----------------|---|-----------|---------------|---------------------------|----------------|------------|
| Commandes       | Browse Structure Search Insert Empty Drop | 9         | InnoDB        | utf8mb4_0900_ai_ci        | 32.0 Kib       | -          |
| Produits        | Browse Structure Search Insert Empty Drop | 3         | InnoDB        | utf8mb4_0900_ai_ci        | 2.5 Mib        | -          |
| users           | Browse Structure Search Insert Empty Drop | 7         | InnoDB        | utf8mb4_0900_ai_ci        | 16.0 Kib       | -          |
| Variantes       | Browse Structure Search Insert Empty Drop | 4         | InnoDB        | utf8mb4_0900_ai_ci        | 16.0 Kib       | -          |
| <b>4 tables</b> | <b>Sum</b>                                | <b>23</b> | <b>InnoDB</b> | <b>utf8mb4_0900_ai_ci</b> | <b>2.6 Mib</b> | <b>0 B</b> |

FIGURE 7.11 – Voici la liste des tables de la base de données "siteweb" de mon serveur. Les autres bases de données -visibles sur la gauche de l'écran - sont présentes par défaut

The screenshot shows the 'Table structure' view for the 'Commandes' table in the 'siteweb' database. The table has 16 columns:

| #  | Name                       | Type     | Collation          | Attributes | Null              | Default | Comments          | Extra | Action   |
|----|----------------------------|----------|--------------------|------------|-------------------|---------|-------------------|-------|--|
| 1  | <b>id</b>                  | int      |                    | No         | None              |         | AUTO_INCREMENT    |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 2  | <b>id_utilisateur</b>      | int      |                    | Yes        | NULL              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 3  | <b>date_commande</b>       | datetime |                    | Yes        | CURRENT_TIMESTAMP |         | DEFAULT_GENERATED |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 4  | <b>montant_total</b>       | int      |                    | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 5  | <b>nom_config</b>          | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 6  | <b>num_client</b>          | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 7  | <b>prenom_client</b>       | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 8  | <b>country</b>             | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 9  | <b>adresse_facturation</b> | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 10 | <b>code_postal</b>         | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 11 | <b>adresse_livraison</b>   | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 12 | <b>telephone_number</b>    | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 13 | <b>email</b>               | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 14 | <b>card_number</b>         | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 15 | <b>expiration_date</b>     | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 16 | <b>cryptogramme</b>        | text     | utf8mb4_0900_ai_ci | No         | None              |         |                   |       | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |

Below the table structure, there are sections for 'Indexes' (showing a primary key on 'id' and an index on 'id\_utilisateur'), 'Partitions' (empty), and 'Information' (showing data and index sizes).

FIGURE 7.12 – Voici la structure de la table "Commandes". Nous pouvons observer que comme pour plusieurs des prochaines tables, elle possède une entrée "id" unique à chaque élément qui est générée automatiquement et une entrée de date (également automatique) enregistrant la date de création de la ligne de donnée.

The screenshot shows the 'Structure' tab for the 'Produits' table in the 'siteweb' database. The table has 5 columns:

|   | <b>id</b> | <b>nom</b>       | <b>description</b>                                    | <b>categorie</b> | <b>image</b>       |
|---|-----------|------------------|---|------------------|--------------------|
| 1 | 1         | PC Gaming        | Performant, dédié aux jeux-videos et aux programme... | Gaming           | [BLOB - 623.3 KiB] |
| 2 | 2         | PC Professionnel | Economique et fiable. Destiné aux entreprises et a... | Professionnel    | [BLOB - 455.4 KiB] |
| 3 | 3         | PC Custom        | Customisez votre propre ordinateur ! Nous nous occ... | Custom           | [BLOB - 1.1 MiB]   |

Below the table, there are sections for 'query results operations' (with options like Print, Copy to clipboard, Export, Display chart, Create view) and 'Bookmark this SQL query' (with a label input field and a checkbox for 'Let every user access this bookmark').

FIGURE 7.13 – Voici le contenu de la table "produits".

The screenshot shows the phpMyAdmin interface for the 'Produits' table in the 'siteweb' database. The table structure is defined as follows:

| # | Name               | Type                                      | Collation                     | Attributes | Null | Default        | Comments | Extra | Action |
|---|--------------------|---|-------------------------------|------------|------|----------------|----------|-------|--------|
| 1 | <b>id</b>          | int                                       | utf8mb4_0900_ai_ci            | No         | None | AUTO_INCREMENT |          |       |        |
| 2 | <b>nom</b>         | varchar(255)                              | utf8mb4_0900_ai_ci            | Yes        | NULL |                |          |       |        |
| 3 | <b>description</b> | text                                      | utf8mb4_0900_ai_ci            | Yes        | NULL |                |          |       |        |
| 4 | <b>categorie</b>   | enum('Gaming', 'Professionnel', 'Custom') | utf8mb4_0900_ai_ci            | Yes        | NULL |                |          |       |        |
| 5 | <b>image</b>       | longblob                                  | Media (MIME) type: image/jpeg | Yes        | NULL |                |          |       |        |

Below the table structure, there are tabs for 'Indexes' and 'Partitions'. The 'Indexes' tab shows a single primary key index named 'PRIMARY' using a BTREE algorithm. The 'Partitions' tab indicates 'No partitioning defined!'. The 'Information' tab provides detailed statistics about the table's storage usage.

FIGURE 7.14 – Voici la structure de la table "produits".

The screenshot shows the phpMyAdmin interface for the 'Variantes' table in the 'siteweb' database. The table contains the following data:

| + Options | - T-+ | categorie | nom    | id          | Prix | CPU  | GPU  | HDD   | SDD   | RAM  | CPU Cooler  |  |
|-----------|-------|-----------|--------|-------------|------|------|--|---|---|--|---|--|
|           |       |           | Gaming | Ultra       | 1    | 5000 | Intel Core i9-13900KS 3 GHz 24-Core              | Asus ROG STRIX GAMING OC GeForce RTX 4090 24 GB       | 2x Seagate Barracuda Compute 8 TB 3.5" 5400 RPM HD... | Samsung 980 Pro 2 TB M.2-2280 PCIe 4.0 X4 NVME SSD   | Corsair Vengeance RGB 32 GB (2 x 16 GB) DDR5          | Corsair iCUE H170i ELITE LCD XT 89 CFM Liquid CPU...   |
|           |       |           | Gaming | Performance | 2    | 2740 | Intel Core i7-12700F 2.1 GHz 12-Core             | MSI VENTUS 3X PLUS OC GeForce RTX 3080 12GB LHR 12... | Seagate Barracuda Compute 8 TB 3.5" 5400 RPM HD...    | Samsung 980 Pro 2 TB M.2-2280 PCIe 4.0 X4 NVME SSD   | Corsair Vengeance RGB Pro 16 GB (2 x 8 GB) DDR4-32... | Corsair iCUE H150i ELITE CAPELLIX 75 CFM Liquid CPU... |
|           |       |           | Pro    | Apex        | 3    | 1500 | Intel Core i7-13700K 3.4 GHz 16-Core Processor   | MSI GeForce RTX 3060 Ventus 2X 12G GeForce RTX 306... | Seagate BarraCuda 2 TB 2.5" 5400 RPM Internal Hard... | Crucial P3 500 GB M.2-2280 PCIe 3.0 X4 NVME Solid... | TEAMGROUP Elite 32 GB (2 x 16 GB) DDR4-3200 CL22 M... | Cooler Master Hyper 212 Black Edition 42 CFM CPU C...  |
|           |       |           | Pro    | EcoTech     | 4    | 700  | Intel Core i3-12100F 3.3 GHz Quad-Core Processor | Cet ordinateur, n'étant pas destiné au graphisme 3... | Seagate BarraCuda 2 TB 2.5" 5400 RPM Internal Hard... | Crucial P3 500 GB M.2-2280 PCIe 3.0 X4 NVME Solid... | Silicon Power GAMING 16 GB (2 x 8 GB) DDR4-3200 CL... | Cooler Master Hyper 212 Black Edition 42 CFM CPU C...  |

Below the table, there are tabs for 'Query results operations' and 'Bookmark this SQL query'. The 'Query results operations' tab includes options like 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

FIGURE 7.15 – Voici le contenu de la table "variantes".

The screenshot shows the phpMyAdmin interface for the 'Variantes' table in the 'siteweb' database. The table has 10 columns:

| #  | Name       | Type                                | Collation          | Attributes | Null | Default | Comments | Extra          | Action   |
|----|------------|-------------------------------------|--------------------|------------|------|---------|----------|----------------|--|
| 1  | categorie  | enum('Gaming', 'Pro', 'Custom', '') | utf8mb4_0900_ai_ci |            | No   | None    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 2  | nom        | text                                | utf8mb4_0900_ai_ci |            | No   | None    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 3  | id         | int                                 |                    |            | No   | None    |          | AUTO_INCREMENT | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 4  | Prix       | int                                 |                    |            | Yes  | NULL    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 5  | CPU        | text                                | utf8mb4_0900_ai_ci |            | Yes  | NULL    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 6  | GPU        | text                                | utf8mb4_0900_ai_ci |            | Yes  | NULL    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 7  | HDD        | text                                | utf8mb4_0900_ai_ci |            | Yes  | NULL    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 8  | SDD        | text                                | utf8mb4_0900_ai_ci |            | Yes  | NULL    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 9  | RAM        | text                                | utf8mb4_0900_ai_ci |            | Yes  | NULL    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 10 | CPU Cooler | text                                | utf8mb4_0900_ai_ci |            | Yes  | NULL    |          |                | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |

Indexes section:

| Action                                    | Keyname | Type  | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---------|-------|--------|--------|--------|-------------|-----------|------|---------|
| <a href="#">Edit</a> <a href="#">Drop</a> | PRIMARY | BTREE | Yes    | No     | id     | 4           | A         | No   |         |

Information section:

| Data      | 16.0 Kib | Format         | dynamic                  |
|-----------|----------|----------------|--------------------------|
| Index     | 0 B      | Collation      | utf8mb4_0900_ai_ci       |
| Overhead  | 0 B      | Next autoindex | 7                        |
| Effective | 16.0 Kib | Creation       | Oct 25, 2023 at 02:52 AM |
| Total     | 16.0 Kib | Last update    | Nov 23, 2023 at 10:02 PM |
|           |          | Last check     | Nov 23, 2023 at 10:02 PM |
|           |          | Row statistics |                          |

FIGURE 7.16 – Voici la structure de la table "variantes".

The screenshot shows the phpMyAdmin interface for the 'users' table in the 'siteweb' database. The table has 4 columns:

| # | Name          | Type         | Collation          | Attributes | Null | Default           | Comments | Extra             | Action   |
|---|---------------|--------------|--------------------|------------|------|-------------------|----------|-------------------|--|
| 1 | id            | int          |                    |            | No   | None              |          | AUTO_INCREMENT    | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 2 | user_name     | varchar(255) | utf8mb4_0900_ai_ci |            | Yes  | NULL              |          |                   | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 3 | password      | varchar(255) | utf8mb4_0900_ai_ci |            | Yes  | NULL              |          |                   | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |
| 4 | date_creation | timestamp    |                    |            | Yes  | CURRENT_TIMESTAMP |          | DEFAULT_GENERATED | <a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a> |

Indexes section:

| Action                                    | Keyname | Type  | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---------|-------|--------|--------|--------|-------------|-----------|------|---------|
| <a href="#">Edit</a> <a href="#">Drop</a> | PRIMARY | BTREE | Yes    | No     | id     | 4           | A         | No   |         |

Information section:

| Data      | 16.0 Kib | Format         | dynamic                  |
|-----------|----------|----------------|--------------------------|
| Index     | 0 B      | Collation      | utf8mb4_0900_ai_ci       |
| Overhead  | 0 B      | Next autoindex | 64                       |
| Effective | 16.0 Kib | Creation       | Oct 27, 2023 at 06:11 PM |
| Total     | 16.0 Kib | Last update    | Nov 23, 2023 at 09:51 PM |
|           |          | Last check     | Nov 23, 2023 at 10:01 PM |
|           |          | Row statistics |                          |

FIGURE 7.17 – Voici la structure de la table "users".