

# SAE23

METTRE EN PLACE UNE  
SOLUTION INFORMATIQUE

ANDJIB Houdhayf BUT1 R&T  
BOUALI Idriss BUT1 R&T

Nom:

Email:

Téléphone:

Adresse:

Ajouter

Voir les  
clients et  
sites

---

<b>Introduction:</b>	<b>3</b>
<b>I- Réflexion de la base de données:</b>	<b>4</b>
Conception initiale et concertation	4
Diagramme de classe	4
<b>I.I-Mise en oeuvre de la base de données</b>	<b>5</b>
<b>II-Guide technique</b>	<b>6</b>
Utilisation de XAMPP	6
Problème courant	9
<b>III-Test et validation</b>	<b>11</b>
1. Ajout d'un client :	11
Stylisation du site :	11
Mise en place de rôles utilisateurs :	12
<b>Outils utilisé:</b>	<b>13</b>
<b>Conclusion :</b>	<b>14</b>

---

---

## **Introduction:**

Ce rapport présente le développement et la mise en œuvre d'une plateforme de gestion des informations réseau pour notre entreprise. En réponse aux besoins spécifiques définis dans le cahier des charges, nous avons conçu et réalisé une solution informatique complète visant à automatiser l'attribution des adresses IP et à centraliser la gestion des informations client.

La première étape du projet a été la conception de la base de données. Nous avons élaboré une base de données centralisée, incluant la définition des diagrammes de classe et des schémas relationnels, ainsi que la mise en place des clés primaires et étrangères, garantissant une structuration optimale des données.

Ensuite, nous avons développé des interfaces utilisateurs intuitives pour faciliter l'ajout et la suppression de clients, ainsi que pour permettre une visualisation claire et automatique des configurations client. Ces interfaces ont été conçues pour offrir une expérience utilisateur fluide et efficace, répondant aux besoins quotidiens de gestion et de consultation des données.

Tout au long du projet, nous avons scrupuleusement suivi les spécifications du cahier des charges pour nous assurer que toutes les fonctionnalités requises ont été mises en œuvre. Cela inclut l'automatisation de l'adressage IP et la gestion des VLAN, éléments essentiels pour la fluidité et la précision de notre système.

Enfin, une série de tests rigoureux a été menée pour vérifier la fiabilité et la performance de la plateforme. Ces tests incluent des tests unitaires et des tests d'intégration, garantissant que toutes les composantes fonctionnent harmonieusement et répondent aux attentes.

Ce rapport fournit une vue détaillée du processus de développement, soulignant notre approche méthodique et les solutions techniques adoptées pour répondre aux besoins spécifiques de notre entreprise. Grâce à cette nouvelle plateforme, nous sommes en mesure d'améliorer considérablement l'efficacité et la précision de la gestion des informations réseau, offrant ainsi un service optimisé à nos clients.

---

## **I- Réflexion de la base de données:**

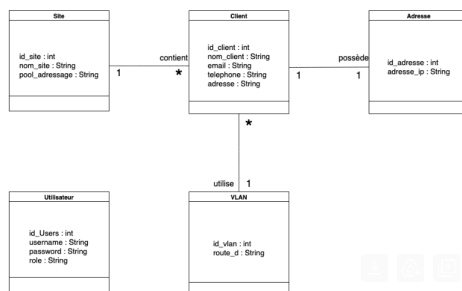
Dans cette section, nous détaillons le processus de mise en place de la base de données pour notre projet, en respectant scrupuleusement les exigences définies dans le cahier des charges. Notre objectif principal était de créer une base de données centralisée et efficace pour la gestion des informations réseau et l'automatisation de l'attribution des adresses IP aux clients.

### ***Conception initiale et concertation***

Nous avons commencé par une phase de réflexion approfondie sur la structure de la base de données. Cette étape a impliqué une analyse minutieuse des besoins exprimés dans le cahier des charges et une concertation avec d'autres binômes travaillant sur des parties connexes du projet. Cette collaboration nous a permis de garantir une cohérence et une intégration harmonieuse de notre base de données avec les autres composants du système global. De cette concertation est venu le choix d'utiliser SQL pour établir notre base de données car via ceci nous pourrions mettre en place une base de données robuste et fiable mais aussi d'avoir une gestion centralisée des différentes données.

### ***Diagramme de classe***

Le diagramme de classe illustre les entités principales de notre base de données ainsi que leurs relations. Voici une version simplifiée du diagramme :



Notre base de données est composée de quatres tables permettant de répartir et gérer correctement les différentes informations, la table VLAN permettra de gérer les différents VLAN attribués, la table site permettra à chacun des trois sites de pouvoir cohabiter et gérer correctement leur différents clients. Quant à celle utilisateur, elle va permettre la connexion de chaque utilisateur grâce à leur mot de passe et leur nom d'utilisateur, mais aussi de permettre la création de deux différent rôles administrateur et client que nous expliquerons plus tard dans ce rapport. Enfin la table adresse qui gèrera les différentes adresses IP et permettra d'attribuer automatiquement les adresses IP aux clients tout en vérifiant que cette dernière est disponible et pas déjà attribuée.

Une fois après avoir mis en place la base de données, les différents binômes se sont séparés afin de mettre en place l'IHM des différents sites.

---

## 1.1-Mise en oeuvre de la base de données

Dans cette partie nous allons voir la création de notre base de donnée et de ces différentes tables la composant ainsi que les valeurs initiales que nous avons inséré dans certaines d'entre elles.

```
CREATE TABLE Users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) NOT NULL UNIQUE,  
  password VARCHAR(50) NOT NULL,  
  role ENUM('admin', 'client') NOT NULL  
);  
  
CREATE TABLE Sites (  
  id_site INT AUTO_INCREMENT PRIMARY KEY,  
  nom_site VARCHAR(100) NOT NULL,  
  pool_adressage VARCHAR(18) NOT NULL  
);  
  
-- Insérer des sites  
INSERT INTO Sites (nom_site, pool_adressage) VALUES ('Site 1', '164.166.1.0/24');  
INSERT INTO Sites (nom_site, pool_adressage) VALUES ('Site 2', '164.166.2.0/24');  
INSERT INTO Sites (nom_site, pool_adressage) VALUES ('Site 3', '164.166.3.0/24');
```

Ci-dessus nous pouvons voir la création de deux tables **users** et **sites**, l'une va nous permettre de créer les deux rôles évoqués plutôt qui sont **admin** et **client** afin de séparer les différents droit des personnes qui se connectent. La table **site** quant à elle va nous permettre de définir les différents pools d'adressage liés aux différents sites après sa création nous avons inséré comme valeurs initiales les différentes adresses du pool d'adressage des différents sites.

```
CREATE TABLE VLAN (  
  id_vlan INT AUTO_INCREMENT PRIMARY KEY,  
  route_d VARCHAR(100) NOT NULL  
);  
  
-- Insérer des VLANs  
INSERT INTO VLAN (route_d) VALUES ('VLAN 1');  
INSERT INTO VLAN (route_d) VALUES ('VLAN 2');  
INSERT INTO VLAN (route_d) VALUES ('VLAN 3');
```

Ici ci-dessus la table **VLAN** ainsi que ces différents VLAN la composant, ici nous avons attribué à chaque site un VLAN qu'il va pouvoir configurer et gérer ainsi que pouvoir mettre des clients dans ce dernier, bien évidemment nous avons aussi la possibilité d'augmenter le nombre de VLAN géré par sites.

```
CREATE TABLE Clients (  
  id_client INT AUTO_INCREMENT PRIMARY KEY,  
  nom_client VARCHAR(100) NOT NULL,  
  email VARCHAR(100) NOT NULL,  
  telephone VARCHAR(100) NOT NULL,  
  adresse VARCHAR(100) NOT NULL,  
  id_site INT,  
  id_vlan INT,  
  FOREIGN KEY (id_site) REFERENCES Sites(id_site) ON DELETE CASCADE,  
  FOREIGN KEY (id_vlan) REFERENCES VLAN(id_vlan) ON DELETE SET NULL  
);  
  
CREATE TABLE Adresse (  
  id_adresse INT AUTO_INCREMENT PRIMARY KEY,  
  adresse_ip VARCHAR(15) NOT NULL,  
  id_client INT,  
  FOREIGN KEY (id_client) REFERENCES Clients(id_client) ON DELETE CASCADE  
);  
  
-- Ajout d'un utilisateur admin pour tester  
INSERT INTO Users (username, password, role) VALUES ('admin', 'admin', 'admin');  
INSERT INTO Users (username, password, role) VALUES ('client', 'client', 'client');
```

Enfin la création des tables **clients** et **adresse** ces dernières sont étroitement liées lorsqu'un client sera ajouté à la base de données certaines informations dont son nom, son id et autres seront ajouté à la table **client**, puis une fois cela fait l'adresse IP qui lui sera automatiquement attribué après vérification du script afin de s'assurer qu'elle ne soit pas déjà attribué va être inséré dans la table **adresse** afin de notifier le script que cette adresse est prise et qu'elle ne soit pas associé aussi à un autre utilisateur. Et nous pouvons aussi remarquer l'insertion des valeurs initiales de la table **users** avec deux utilisateurs admin et client qui nous permettra par la suite de réaliser des tests au niveau des droits accordés aux différents utilisateurs.

---

## II-Guide technique

Dans cette section, nous expliquons en détail le processus de tests et de validation de notre plateforme, en utilisant l'outil XAMPP pour simuler un environnement de serveur local, mais avant cela voici un guide technique de configuration et d'utilisation de XAMPP.

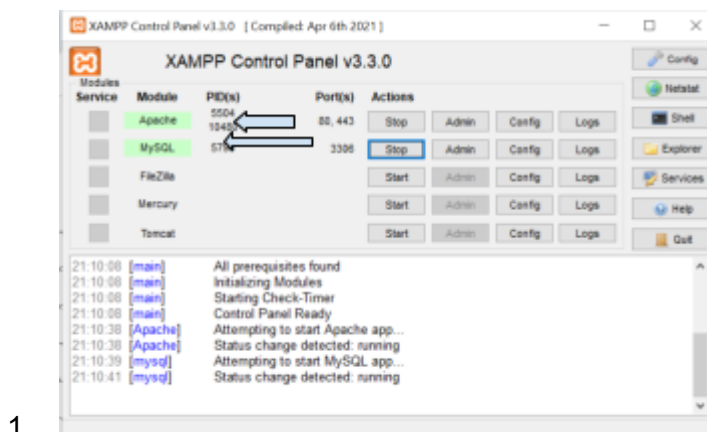
### Utilisation de XAMPP

Pour les tests, nous avons utilisé XAMPP, une solution de serveur web multiplateforme qui inclut Apache, MySQL, et PHP. Cet outil nous a permis de configurer rapidement un serveur local pour déployer et tester notre application web.

Nous présenterons des captures d'écran et des tutoriels détaillés sur l'utilisation de XAMPP, mais voici un résumé des étapes principales :

**Installation de XAMPP** : Téléchargement et installation de XAMPP sur notre machine.

**Configuration du serveur Apache** : Démarrage du module Apache à partir du panneau de contrôle de XAMPP.

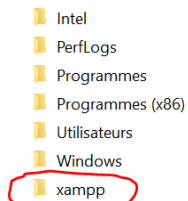


Une fois XAMPP installé puis lancé nous obtenons ce panneau de contrôle, une fois sur ce dernier démarrer le serveur Apache ainsi que MySQL.

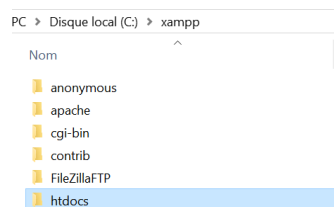
---

**Déploiement de l'application** : Copie des fichiers de notre application web dans le répertoire **htdocs** de XAMPP.

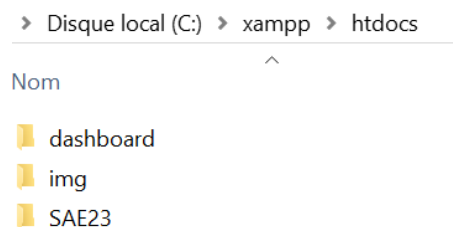
Se rendre dans la localisation où se trouve XAMP



Une fois cela fait entrer dedans et ensuite entrer dans htdocs



Une fois dedans créer un dossier qui contiendra la totalité des script

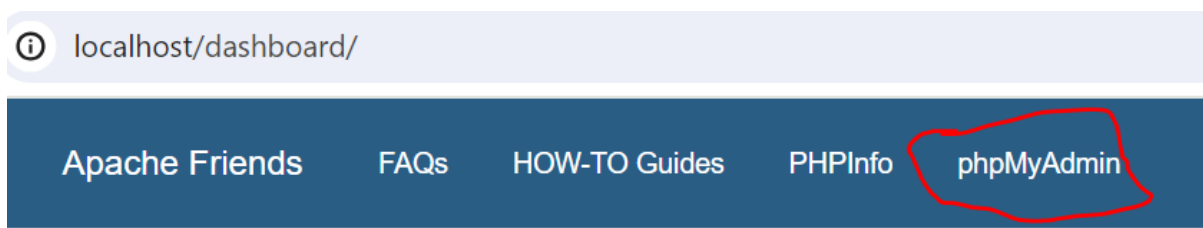


Ici j'ai créé un dossier SAE23.

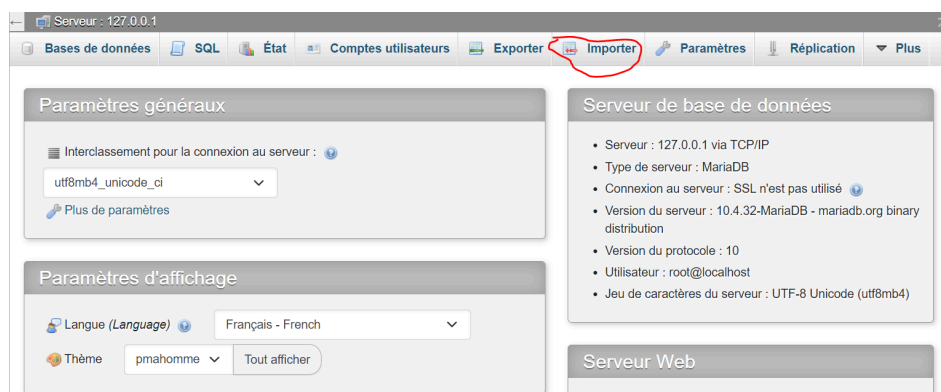
**Configuration de la base de données** : Utilisation de phpMyAdmin pour créer les tables nécessaires dans MySQL.



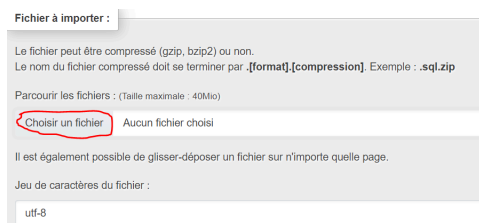
Dans la barre d'URL taper ceci, une fois fait nous serons redirigé ici :



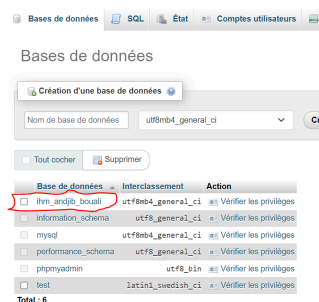
une fois fait sélectionné phpMyAdmin puis nous nous retrouverons dans cette endroit :



Après cela il faut aller dans importer puis on va importer notre base de donnée (fichier SQL) :



Après avoir importé le fichier SQL on pourra se rendre dans **Base de données** et on verra la base de donnée apparente :





---

Puis pour exécuter les différents fichiers il faudra dans la barre de navigateur entrer le lien suivant : `http://localhost/SAE23/index.php`

En oubliant pas de modifier SAE23 par le nom du dossier dans lequel se trouvent les scripts et `index.php` par le fichier à exécuter.

### Problème courant

Il est possible de rencontrer certains problèmes lors de la tentative d'exécution des fichiers `.php` cela est très souvent lié au problème de connexion à la base de données, mais il existe une solution afin de régler cela pour se faire il faut vérifier le contenu de son fichier de connexion à la base de données :

```
<?php
$servername = "localhost";
$username = "root";
$password = "root"; // Mettez le mot de passe de votre base de données ici
$dbname = "IHM_ANDJIB_BOUALI"; // Assurez-vous que le nom de la base de données est correct

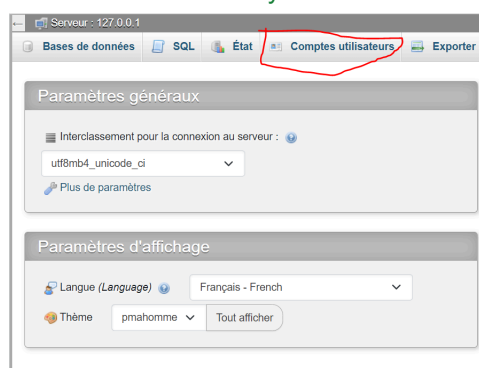
// Créer une connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérifier la connexion
if ($conn->connect_error) {
    die("La connexion a échoué : " . $conn->connect_error);
}
?>
```

Ci-dessus le contenu de notre fichier de connexion à la base de données, sur XAMPP l'utilisateur par défaut sera `root` et son mot de passe ne sera pas défini il n'y aura aucun mot de passe alors dans ce cas dans la variable `$password` laissez vide de cette manière : `$password = ""`;

Si le problème persiste, on va créer un utilisateur et définir son mot de passe.

Se rendre dans **PhpMyAdmin** comme vu plus haut dans le guide technique :



Une fois dedans vous aurez plus qu'à créer un nouvel utilisateur et définir un mot de passe.

Une fois cela fait n'oubliez pas d'entrer le mot de passe nouvellement créé et le nom de l'utilisateur créé ici `root` pour l'utilisateur et `root` pour le mot de passe. **Ne pas oublier lorsque l'on crée l'utilisateur de s'accorder les droits root, il faudra seulement cocher la case l'indiquant.**

```
<?php
$servername = "localhost";
$username = "root";
$password = "root"; // Mettez le mot de passe de votre base de données ici
$dbname = "IHM_ANDJIB_BOUALI"; // Assurez-vous que le nom de la base de données est correct

// Créer une connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérifier la connexion
if ($conn->connect_error) {
    die("La connexion a échoué : " . $conn->connect_error);
}
?>
```

Enfin si l'erreur persiste assurez-vous que le nom de votre base de donnée correspond à celle de votre base créée :

```
#!/php
$servername = "localhost";
$username = "root";
$password = "root"; // Mettez le mot de passe de votre base de données ici
$dbname = "IHM_ANDJIB_BOUALI"; // Assurez-vous que le nom de la base de données est correct

// Créer une connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérifier la connexion
if ($conn->connect_error) {
    die("La connexion a échoué : " . $conn->connect_error);
}
?>
```

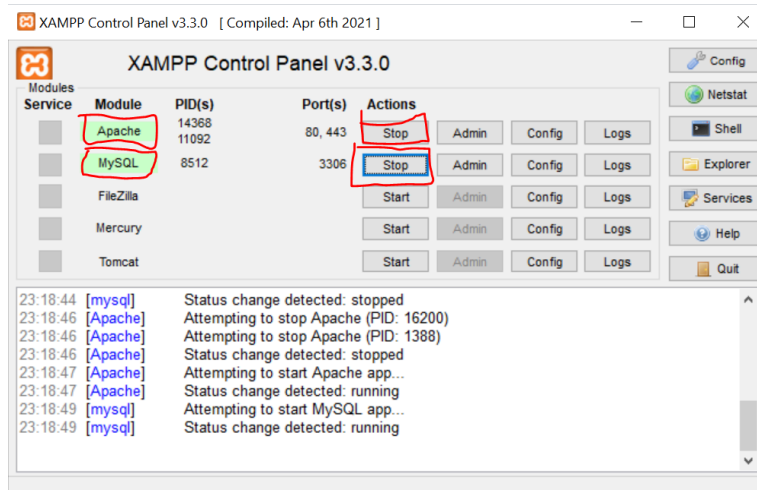
```
CREATE DATABASE IHM_ANDJIB_BOUALI;

USE IHM_ANDJIB_BOUALI;

CREATE TABLE Users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(50) NOT NULL,
    role ENUM('admin', 'client') NOT NULL
);
```

Une fois cela fait, vous allez pouvoir avoir accès à la base de données.

Si vous n'arrivez pas à vous connecter sur le PhpMyAdmin ou encore au localhost vue dans la partie [Guide technique](#), il suffit de vérifier si le serveur Apache et le service MySQL sont toujours bien actifs.



---

## III-Test et validation

### 1. Ajout d'un client :

- Nous avons commencé par ajouter un client via l'interface utilisateur de notre application.

Clients				
id_client	nom_client	email	telephone	adresse
1	Wael	goutzat.wael@gmail.com	0606060606	123 Rue Exemple

**Site**  
Aucune donnée disponible

**Adresse**  
Aucune donnée disponible

**VLAN**  
Aucune donnée disponible

Nous pouvons voir que l'utilisateur a bien été ajouté mais qu'aucun site ni VLAN n'est attribué ni même d'adresse IP alors nous avons modifié notre script ajout.php permettant d'ajouter les clients pour palier cela.

Une fois la modification faite nous avons rajouté un client

Nous avons vérifié que l'adresse IP et le VLAN étaient générés automatiquement conformément aux règles définies dans notre base de données.

Client: Test	
ID:	15
Email:	Test@gmail.com
Téléphone:	0000
Adresse:	Tremblay
Nom du Site:	Site 2
VLAN:	VLAN 2
Adresse IP:	164.166.2.5
Supprimer	

On peut donc voir l'adresse mail, numéro de téléphone, l'adresse, le site, VLAN associé et son adresse IP ainsi que la possibilité de supprimer ce client si ce dernier décide de ne plus vouloir faire appel à nos services.

### Stylisation du site :

Une fois les fonctionnalités de base validées, nous avons stylisé le site pour le rendre plus attractif et convivial.

---

On peut voir l'ajout d'icône afin d'amener un côté attrayant mais aussi une couleur de fond et une seconde couleur qui est le orange qui va ressortir à l'aide du fond sombre qu'est le noir.

### Mise en place de rôles utilisateurs :

Nous avons implémenté deux types d'utilisateurs : un administrateur et un utilisateur client, qui sont les deux rôles dont on vous parle depuis le début lors de la conception de la base de données voyons qu'est-ce qui différencie ces deux rôles.

**Administrateur** : Capable d'ajouter des adresses IP, de visualiser et de gérer toutes les informations clients, y compris les données sensibles. (login : admin et mot de passe : admin)

**Utilisateur client** : Accès restreint à la visualisation des adresses IP et VLAN attribuées, sans accès aux informations sensibles. (login : client et mot de passe : client)

Pour chaque étape, nous avons effectué des tests fonctionnels pour vérifier que les rôles et les permissions étaient correctement appliqués et respectaient les exigences de sécurité et de confidentialité.

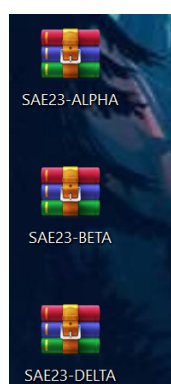
Ci-dessus on peut voir les informations auxquels l'administrateur a accès et ci-dessous celle des clients :

On peut voir que le client n'a pas accès aux données sensibles (nom,téléphone,adresse) et il n'a pas la possibilité de supprimer un autre client aussi.

---

Après cela nous avons testé chaque fonctionnalité une par une, une fois qu'elles se sont avérées fonctionnelles nous avons pu valider l'IHM. Nous venons de vous montrer les tests unitaires où nous avons testé puis validé chaque paramètre de manière isolée, ce qui facilite la détection et la correction des erreurs précoces. La raison pour laquelle nous avons procédé ainsi avec des tests unitaires c'est parce qu'ils encouragent une conception modulaire, ce qui améliore la maintenabilité du code.

Nous avons aussi fait des tests de régression qui maintiennent la stabilité de l'application en s'assurant que les nouvelles modifications n'affectent pas négativement les fonctionnalités existantes. Pour se faire nous avons créé plusieurs sauvegardes de nos codes comme ça si l'une des modifications venaient à ruiner tout le reste nous pourrions revenir en arrière sur une ancienne sauvegarde à l'aide de différentes versions de nos codes comme des versions ALPHA, BETA et DELTA :



Et aussi des tests de performance en ajoutant plusieurs utilisateurs à la fois afin de nous assurer que notre IHM peut gérer la charge prévue et offrir une performance acceptable aux utilisateurs.

## **Outils utilisé:**

Pour le développement de notre plateforme de gestion des informations réseau, nous avons utilisé plusieurs technologies et langages de programmation, chacun jouant un rôle crucial dans la réalisation de ce projet.

PHP (Hypertext Preprocessor) est un langage de script côté serveur utilisé principalement pour le développement web. Intégré dans le code HTML, il est exécuté sur le serveur pour générer du contenu dynamique sur les pages web. Dans notre projet, PHP a été essentiel pour gérer les formulaires, traiter les données soumises pour l'ajout et la suppression de clients, et interagir avec notre base de données MySQL. Il a également permis de générer du contenu dynamique, affichant les informations des clients, des adresses IP et des VLAN de manière efficace.

HTML (HyperText Markup Language) est le langage standard pour créer et structurer les pages web. Il définit la structure et le contenu d'une page web à l'aide de balises. Nous avons utilisé HTML pour créer la structure de base de toutes les pages de notre application web, notamment les formulaires,

---

les tableaux, et les éléments de navigation. Cela a facilité l'affichage des données des clients, des adresses IP et des VLAN, rendant l'information accessible et bien organisée pour les utilisateurs.

CSS (Cascading Style Sheets) est un langage de style utilisé pour décrire la présentation des documents HTML. Il contrôle l'apparence des éléments sur une page web, y compris la mise en page, les couleurs, les polices, et d'autres aspects de la conception visuelle. Dans notre projet, CSS a été utilisé pour améliorer l'apparence visuelle du site web, le rendant plus attractif et professionnel. Nous avons également appliqué des techniques CSS pour assurer que notre site soit réactif et fonctionne bien sur différents appareils et tailles d'écran.

SQL (Structured Query Language) est un langage de programmation standard pour gérer et manipuler les bases de données relationnelles. Il permet de créer, lire, mettre à jour et supprimer des données dans une base de données. SQL a été essentiel dans notre projet pour définir la structure de notre base de données, incluant les tables **Client**, **IP**, et **VLAN**. Il a permis d'effectuer des opérations sur les données, telles que l'insertion de nouveaux clients, la mise à jour des adresses IP, et la suppression des enregistrements obsolètes. Nous avons également utilisé des requêtes SQL complexes pour extraire et manipuler les données de manière efficace, répondant ainsi aux besoins fonctionnels du projet.

En combinant ces technologies, nous avons pu développer une plateforme complète et fonctionnelle répondant aux besoins spécifiques de gestion des informations réseau pour notre entreprise. PHP a fourni la logique côté serveur nécessaire, HTML et CSS ont créé une interface utilisateur structurée et visuellement agréable, et SQL a assuré une gestion efficace des données, permettant de répondre aux exigences du cahier des charges de manière fiable et performante.

## **Conclusion :**

Ce rapport retrace le processus complet de conception, développement, test et validation de notre plateforme de gestion des informations réseau. En répondant aux exigences spécifiques définies

---

---

dans le cahier des charges, nous avons développé une solution efficace pour l'automatisation de l'attribution des adresses IP et la centralisation de la gestion des informations client.

Le projet a commencé par la conception minutieuse de la base de données, à l'aide d'un diagramme de classe pour garantir une structuration optimale des données. Cette phase a été réalisée en concertation avec d'autres binômes pour assurer la cohérence et l'intégration harmonieuse de notre système.

Nous avons ensuite développé des interfaces utilisateurs intuitives et stylisées, permettant une gestion fluide des informations client. Le respect strict des critères du cahier des charges a été maintenu tout au long du processus, en veillant à ce que chaque fonctionnalité soit implémentée avec précision.

Les tests rigoureux effectués à l'aide de XAMPP ont permis de vérifier la fiabilité et la performance de la plateforme. En simulant un environnement de serveur local, nous avons pu valider l'ajout, la suppression et la gestion des clients, ainsi que la mise en place des rôles d'administrateur et d'utilisateur client.

Les technologies utilisées, notamment PHP, HTML, CSS, et SQL, ont été essentielles pour la réalisation de ce projet. PHP a permis de gérer les interactions côté serveur et de manipuler les données, HTML et CSS ont assuré une structure et une présentation visuellement agréables, et SQL a facilité la gestion efficace des données.