



Introduction	4
Documentation Technique : Méthode de Validation	4
Cahier de Tests	4
Tests Automatisés	14
Tests Unitaires	14
Tutoriel d'installation et d'utilisation	15
Prérequis	15
Étape 1 : Configuration de la machine virtuelle	16
Étape 2 : Installation de PostgreSQL	16
Étape 3 : Installation de XAMPP	17
Étape 4 : Configuration de l'application web	17
Étape 5 : Utilisation de l'application	18
Présentation de l'outil utilisé pour le développement	18
PostgreSQL	19
XAMPP	19
PHP	20
Visual Studio Code	20
Conclusion	21

Introduction

Le présent rapport détaille le développement d'un site web complet permettant la gestion des clients et des utilisateurs au sein d'une entreprise. Ce projet a été conçu pour offrir une solution robuste et sécurisée, capable de gérer efficacement les informations de plusieurs groupes d'utilisateurs avec des permissions spécifiques. En utilisant PostgreSQL pour la gestion de la base de données et PHP pour la logique de l'application, le site garantit une intégration fluide et une performance optimale. L'objectif principal était de créer une application web qui non seulement gère les informations des clients et des utilisateurs, mais qui assure également la sécurité des données à travers une gestion rigoureuse des permissions et des validations.

Documentation Technique : Méthode de Validation

Cette documentation technique présente la méthode de validation utilisée pour assurer le bon fonctionnement de l'application web développée. La méthode de validation inclut un cahier de tests ainsi que des tests unitaires. Ces tests visent à vérifier que les fonctionnalités de l'application répondent aux spécifications et fonctionnent correctement dans divers scénarios.

Cahier de Tests

Le cahier de tests est une liste structurée de tests à exécuter pour vérifier le bon fonctionnement des différentes fonctionnalités de l'application. Chaque test est décrit de manière détaillée, y compris les prérequis, les actions à effectuer, les résultats attendus et le statut du test.

Tests de Connexion et de Session

1. Connexion avec des informations valides

-Préconditions : L'utilisateur est enregistré dans la base de données.

-Actions :

-Accéder à `form-connexion.php` .

-Saisir un login et un mot de passe valides.

Connexion

Login: admin1

Mot de passe: *****

Se connecter

-Soumettre le formulaire.

Bonjour Admin1 Admin.

Aller au menu

-Résultats attendus : L'utilisateur est redirigé vers `menu.php` et voit un message de bienvenue.

Bienvenue, admin1

Afficher la liste des clients
 Afficher les 10 derniers clients
 Ajouter un client
 Ajouter un utilisateur
 Afficher les utilisateurs
 Suppression d'un utilisateur
 Déconnexion

-Statut : [X] Réussi [] Échoué

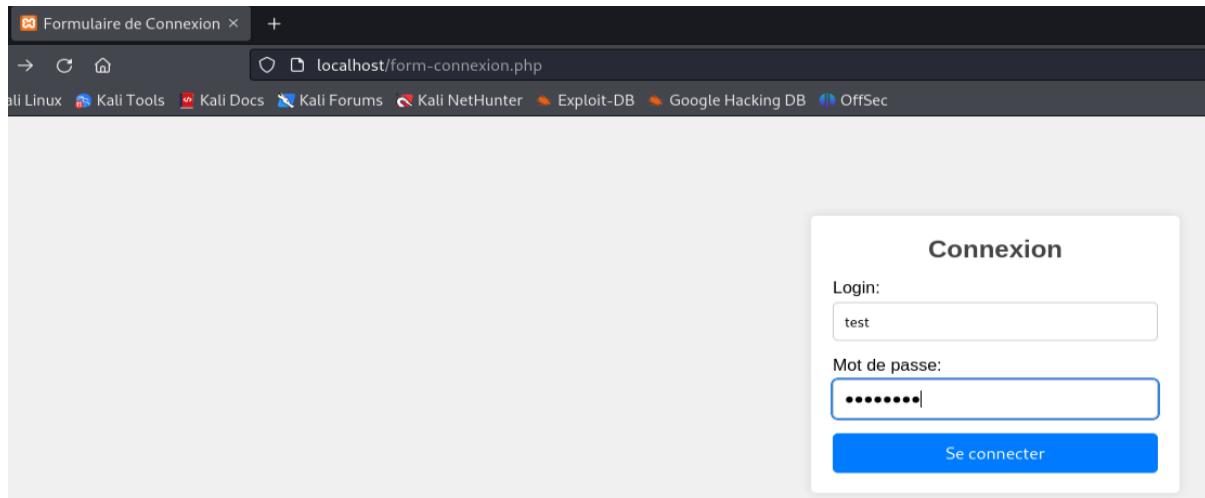
2. Connexion avec des informations invalides

-Préconditions : L'utilisateur est enregistré dans la base de données.

-Actions :

-Accéder à `form-connexion.php` .

-Saisir un login ou un mot de passe invalide.



Connexion

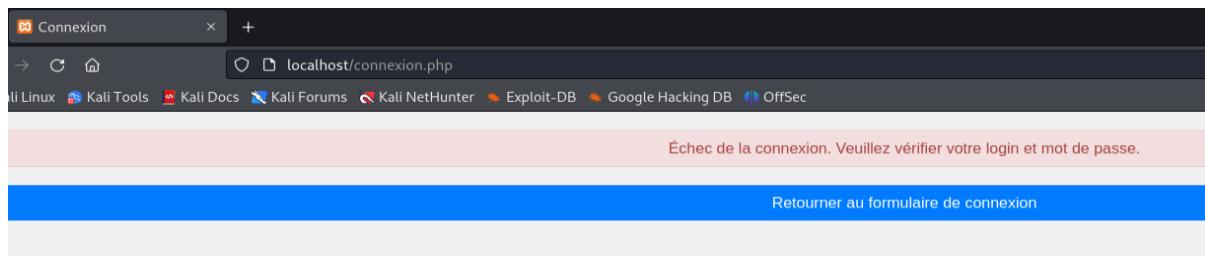
Login:

Mot de passe:

Se connecter

-Soumettre le formulaire.

-Résultats attendus : L'utilisateur voit un message d'erreur indiquant des informations de connexion incorrectes.



Connexion

localhost/connexion.php

Échec de la connexion. Veuillez vérifier votre login et mot de passe.

Retourner au formulaire de connexion

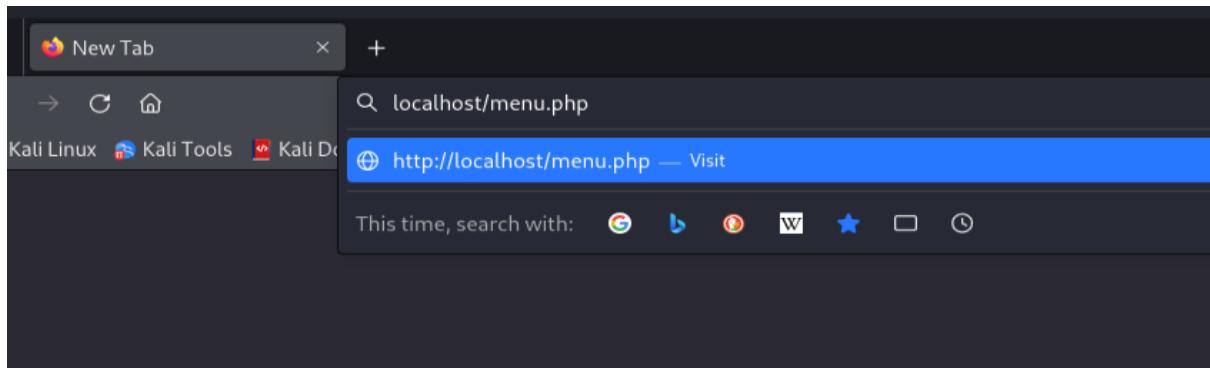
-Statut : [X] Réussi [] Échoué

3. Accès à une page sans être connecté

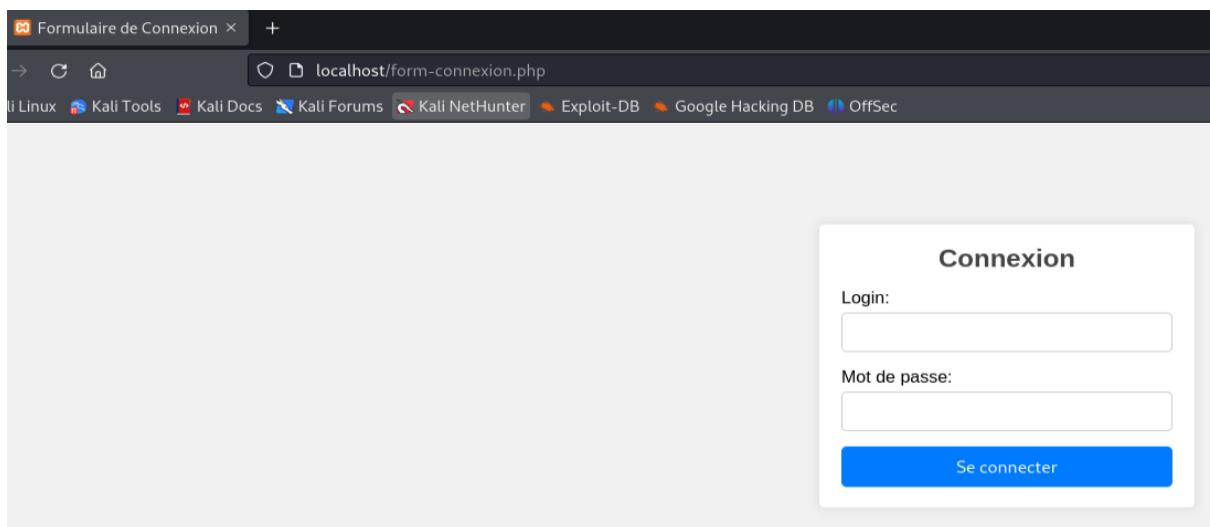
-Préconditions : Aucune session utilisateur active.

-Actions :

- Accéder directement à `menu.php` .



-Résultats attendus : L'utilisateur est redirigé vers `form-connexion.php`.



-Statut : [X] Réussi [] Échoué

Tests de Gestion des Utilisateurs

1. Ajout d'un utilisateur par un administrateur

-Préconditions : L'administrateur est connecté.

-Actions :

-Accéder à `form-ajout-util.php`.

-Saisir les informations nécessaires pour un nouvel utilisateur.

Ajouter un utilisateur

Nom: Said Farah

Prénom: Rayan

Login: Rayan

Mot de passe:

Catégorie: Administrateur

Ajouter l'utilisateur

Retour au menu principal

-Soumettre le formulaire.

-Résultats attendus : Le nouvel utilisateur est ajouté à la base de données avec une adresse IP et un VLAN approprié.

Debug: Appel de getAvailableIp avec groupe = 1

Debug: Adresse IP trouvée = 164.166.1.3

Debug: adresse_ip = 164.166.1.3

Nouvel utilisateur

Rayan Said Farah a bien été ajouté avec succès.

Retour au menu principal

-Statut : [X] Réussi [] Échoué

2. Suppression d'un utilisateur par un administrateur

-Préconditions : L'administrateur est connecté et un utilisateur existe.

-Actions :

- Accéder à `fiche-util.php?id_util=ID` .

Fiche d'un utilisateur	
Nom	Said Farah
Prénom	Rayan
Login	Rayan
Mot de passe	rayan123456
Catégorie	Administrateur
Adresse IP	164.166.1.3
VLAN	1

[Suppression de l'utilisateur](#)

[Retour à la liste des utilisateurs](#)

-Cliquer sur le lien de suppression.

-Résultats attendus : L'utilisateur est supprimé de la base de données et redirigé vers `liste-util.php` avec un message de confirmation.

- Admin1 Admin
- Admin2 Admin
- Admin3 Admin
- Nom1 Prénom1
- Nom2 Prénom2
- Nom3 Prénom3

[Retour au menu](#)

-Statut : [X] Réussi [] Échoué

Tests de Gestion des Clients

1. Ajout d'un client par un employé

-Préconditions : L'employé est connecté.

-Actions :

-Accéder à `form-ajout.php` .

-Saisir les informations nécessaires pour un nouveau client.

Ajout d'un contact

Nom: Said Farah

Prénom: Rayan

Téléphone: 0783353113

Email: rayan-saidfarah@outlook.fr

Adresse: 95200, Sarcelles

Ajouter

Retourner au menu

-Soumettre le formulaire.

-Résultats attendus : Le nouveau client est ajouté à la base de données avec une adresse IP et un VLAN approprié.

Debug: id_util = 4, groupe = 1, categorie = Employé

Debug: Appel de getAvailableIp avec groupe = 1

Debug: Adresse IP trouvée = 164.166.1.4

Debug: adresse_ip = 164.166.1.4

Nouveau contact

Rayan Said Farah a bien été ajouté à vos contacts.

Retour au menu principal

-Statut : [X] Réussi [] Échoué

2. Suppression d'un client par un administrateur

-Préconditions : L'administrateur est connecté et un client existe.

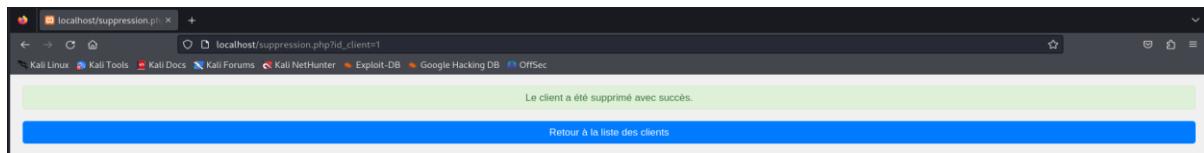
-Actions :

-Accéder à `fiche-client.php?id_client=ID` .

Liste des clients								
ID	Nom	Prénom	Téléphone	Email	Adresse	Adresse IP	VLAN	Action
1	Said Farah	Rayan	0783353113	rayan-saidfarah@outlook.fr	95200, Sarcelles	164.166.1.3	1	Supprimer

-Cliquer sur le lien de suppression.

-Résultats attendus : Le client est supprimé de la base de données et redirigé vers `liste-clients.php` avec un message de confirmation.



-Statut : [X] Réussi [] Échoué

Tests de Validation des Champs

1. Validation du formulaire de connexion

-Préconditions : Aucune.

-Actions :

-Accéder à `form-connexion.php` .

-Soumettre le formulaire avec des champs vides.

-Résultats attendus : Des messages d'erreur indiquant que les champs sont obligatoires apparaissent.

Connexion

Login:

Please fill out this field.

Se connecter

-Statut : [X] Réussi [X] Échoué

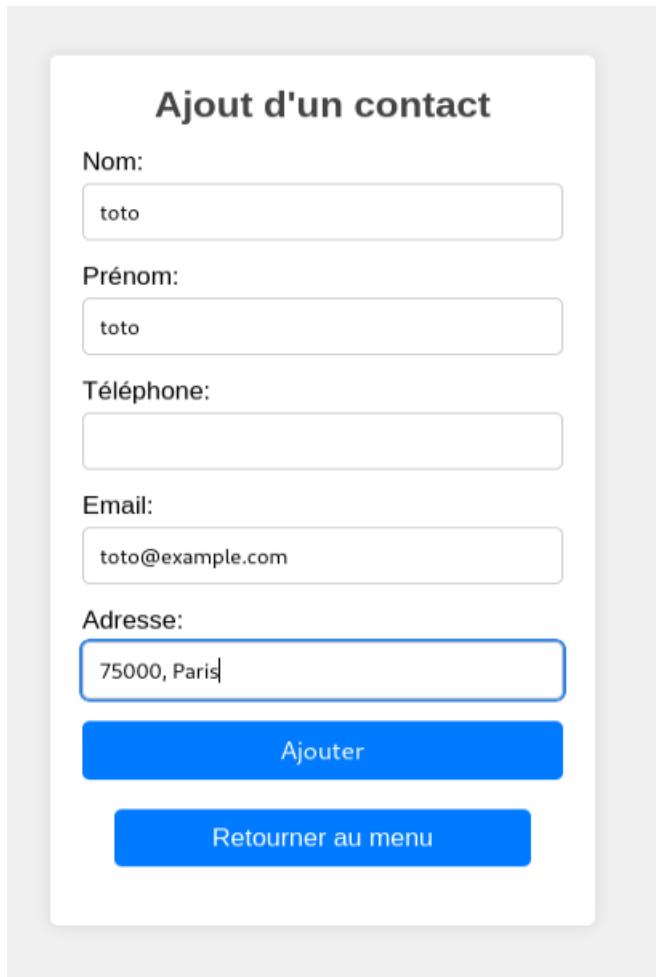
2. Validation du formulaire d'ajout de client

-Préconditions : L'utilisateur est connecté.

-Actions :

-Accéder à `form-ajout.php`.

-Soumettre le formulaire avec des champs vides ou des données incorrectes.



Ajout d'un contact

Nom: toto

Prénom: toto

Téléphone:

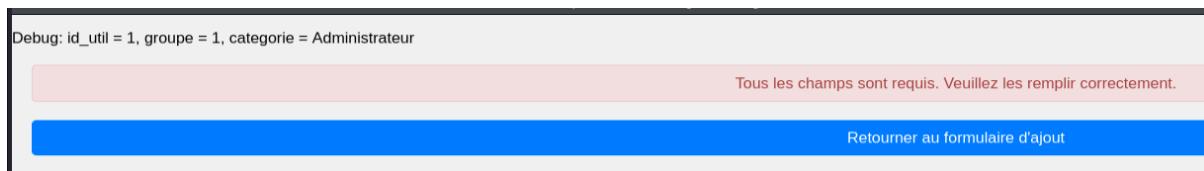
Email: toto@example.com

Adresse: 75000, Paris

Ajouter

Retourner au menu

-Résultats attendus : Des messages d'erreur apparaissent pour les champs obligatoires et les formats incorrects.



Debug: id_util = 1, groupe = 1, categorie = Administrateur

Tous les champs sont requis. Veuillez les remplir correctement.

Retourner au formulaire d'ajout

-Statut : [X] Réussi [] Échoué

3. Validation du formulaire d'ajout d'utilisateur

-Préconditions : L'administrateur est connecté.

-Actions :

-Accéder à `form-ajout-util.php`.

-Soumettre le formulaire avec des champs vides ou des données incorrectes.

Ajouter un utilisateur

Nom:

Prénom:

Login:

Mot de passe:

Catégorie:

Administrateur

Ajouter l'utilisateur

Retour au menu principal

-Résultats attendus : Des messages d'erreur apparaissent pour les champs obligatoires et les formats incorrects.

Ajouter un utilisateur

Nom:

Please fill out this field.

Login:

Mot de passe:

Catégorie:

Administrateur

Ajouter l'utilisateur

Retour au menu principal

-Statut : [X] Réussi [] Échoué

Tests Automatisés

Les tests automatisés permettent de vérifier automatiquement le bon fonctionnement du code via des scripts de test. Nous utilisons PHPUnit pour écrire et exécuter des tests unitaires pour notre application PHP.

1. Installation de PHPUnit

-Installer PHPUnit via Composer :

```
composer require --dev phpunit/phpunit
```

2. Écriture des Tests Unitaires

-Créer des tests unitaires pour les fonctions critiques du projet, telles que `getAvailableIp`, `assignIp`, et `ConnexionBD`.

3. Structure Complète du Projet

```
/opt/lampp/htdocs/
```

```
  └── src/
      └── fonctions.php      # Vos fichiers de code source
  └── tests/
      ├── FunctionsTest.php  # Vos fichiers de tests unitaires pour les fonctions
      ├── ConnexionTest.php  # Vos fichiers de tests unitaires pour la connexion
      └── ExecuteBDTest.php  # Vos fichiers de tests unitaires pour ExecuteBD
  └── composer.json        # Fichier de configuration de Composer
  └── vendor/              # Répertoire des dépendances installées par Composer
      └── bin/
          └── phpunit        # Binaire de PHPUnit installé par Composer
  └── phpunit.xml          # Configuration de PHPUnit (optionnel)
```

Tests Unitaires

Les tests unitaires vérifient que chaque fonction individuelle fonctionne comme prévu. Voici un exemple de tests unitaires pour quelques fonctions critiques.

Tests pour la fonction `getAvailableIp`

```
└─(root㉿kali)-[/opt/lampp/htdocs]
  └─# ./vendor/bin/phpunit tests/FunctionsTest.php
  PHPUnit 9.6.19 by Sebastian Bergmann and contributors.

  ATTENTION: database "postgres" has a collation version mismatch
  DETAIL: The database was created using collation version 2.37, but the operating system's default collation is 2.38.
  HINT: Rebuild all objects in this database that use the default collation and re-create the database.
  <p>Debug: Appel de getAvailableIp avec groupe = 1</p><p>Debug: Adresse IP trouvée: 192.168.1.10</p>
  DETAIL: The database was created using collation version 2.37, but the operating system's default collation is 2.38.
  HINT: Rebuild all objects in this database that use the default collation and re-create the database.
  2 / 2 (100%)

  Time: 00:00.051, Memory: 4.00 MB

  OK (2 tests, 2 assertions)
```

Tests pour la fonction `ConnexionBD`

```
└─(root㉿kali)-[~/opt/lampp/htdocs]
  # ./vendor/bin/phpunit --bootstrap vendor/autoload.php tests/ConnexionTest.php
PHPUnit 9.6.19 by Sebastian Bergmann and contributors.

ATTENTION: database "postgres" has a collation version mismatch
DETAIL: The database was created using collation version 2.37, but the operating
HINT: Rebuild all objects in this database that use the default collation and run
      .
      .
Time: 00:00.024, Memory: 4.00 MB

OK (1 test, 1 assertion)

└─(root㉿kali)-[~/opt/lampp/htdocs]
  #
```

Tests pour la fonction `ExecuteBD`

```
└─(root㉿kali)-[~/opt/lampp/htdocs]
  # ./vendor/bin/phpunit --bootstrap vendor/autoload.php tests/ExecuteBDTest.php
PHPUnit 9.6.19 by Sebastian Bergmann and contributors.

ATTENTION: database "postgres" has a collation version mismatch
DETAIL: The database was created using collation version 2.37, but the operating
HINT: Rebuild all objects in this database that use the default collation and run
      .
      .
Time: 00:00.023, Memory: 4.00 MB

OK (1 test, 1 assertion)
```

Cette documentation technique présente une méthode de validation complète pour l'application, incluant un cahier de tests détaillé et des tests unitaires. Ces tests permettent de garantir que les fonctionnalités de l'application fonctionnent correctement et répondent aux spécifications définies. L'utilisation de tests unitaires automatisés permet également de détecter rapidement les régressions lors des modifications du code.

Tutoriel d'installation et d'utilisation

Ce tutoriel vous guidera à travers les étapes nécessaires pour installer et configurer l'application web sur une machine virtuelle Kali Linux en utilisant VirtualBox. Vous utiliserez PostgreSQL (psql) pour la base de données et XAMPP pour le serveur web. Assurez-vous d'avoir VirtualBox, Kali Linux, PostgreSQL et XAMPP installés avant de commencer.

Prérequis

- VirtualBox installé sur votre machine hôte
- Image ISO de Kali Linux
- XAMPP installé sur Kali Linux
- PostgreSQL installé sur Kali Linux

Étape 1 : Configuration de la machine virtuelle

1.Créer une nouvelle machine virtuelle dans VirtualBox

- Ouvrez VirtualBox.
- Cliquez sur "New" pour créer une nouvelle machine virtuelle.
- Donnez un nom à votre VM (par exemple, "KaliLinux").
- Sélectionnez le type "Linux" et la version "Debian (64-bit)".
- Allouez la mémoire RAM (au moins 2048 MB recommandé).
- Créez un disque dur virtuel (au moins 20 GB recommandé).
- Suivez les instructions pour terminer la configuration.

2.Installer Kali Linux

- Insérez l'image ISO de Kali Linux dans la machine virtuelle (dans les paramètres de la VM, sous "Storage").
- Démarrez la VM et suivez les instructions pour installer Kali Linux.
- Configurez le réseau, les utilisateurs et les mots de passe selon vos préférences.

Étape 2 : Installation de PostgreSQL

1. Mettre à jour le système

```
sudo apt update
```

```
sudo apt upgrade
```

2.Installer PostgreSQL

```
sudo apt install postgresql postgresql-contrib
```

3.Démarrer PostgreSQL et vérifier le statut

```
sudo systemctl start postgresql
```

```
sudo systemctl status postgresql
```

4. Configurer l'utilisateur PostgreSQL

- Accéder à l'utilisateur PostgreSQL :

```
sudo -i -u postgres
```

- Créer un utilisateur et une base de données :

```
createuser --interactive
```

```
createdb nom_de_la_base_de_données
```

```
psql
```

```
ALTER USER nom_utilisateur WITH PASSWORD 'mot_de_passe';
```

```
\q
```

Étape 3 : Installation de XAMPP

1. Télécharger XAMPP

- Téléchargez le package XAMPP pour Linux depuis le site officiel :

<https://www.apachefriends.org/index.html>

2. Installer XAMPP

```
chmod +x xampp-linux-x64-7.4.21-0-installer.run
```

```
sudo ./xampp-linux-x64-7.4.21-0-installer.run
```

3. Démarrer XAMPP

```
sudo /opt/lampp/lampp start
```

4. Vérifier l'installation

- Ouvrez un navigateur et accédez à `http://localhost` pour vérifier que XAMPP fonctionne correctement.

Étape 4 : Configuration de l'application web

1. Déployer le code de l'application

- Copiez le code source de votre application web dans le répertoire `htdocs` de XAMPP.

```
sudo cp -r /path/to/your/application /opt/lampp/htdocs/
```

2. Configurer les fichiers de l'application

- Mettez à jour le fichier de configuration `fonctions.php` pour inclure les informations de connexion à la base de données PostgreSQL.

3. Initialiser la base de données

- Utilisez le script `init.sql` pour initialiser la base de données.

```
psql -U nom_utilisateur -d nom_de_la_base_de_données -f /path/to/init.sql
```

Étape 5 : Utilisation de l'application

1. Accéder à l'application

- Ouvrez un navigateur et accédez à `http://localhost/application` pour voir la page d'accueil de votre application.

2. Se connecter à l'application

- Utilisez le formulaire de connexion pour accéder à l'application avec vos informations d'utilisateur.

3. Naviguer dans l'application

- Utilisez les différents formulaires et liens pour ajouter, afficher et supprimer des utilisateurs et des clients.

4. Vérifier les fonctionnalités

- Testez les différentes fonctionnalités de l'application en suivant le cahier de tests fourni.

En suivant ce tutoriel, vous avez configuré une machine virtuelle Kali Linux avec PostgreSQL et XAMPP, et déployé une application web. Vous avez également appris à utiliser les différentes fonctionnalités de l'application. N'oubliez pas de sécuriser votre serveur et votre base de données en appliquant les meilleures pratiques de sécurité.

Présentation de l'outil utilisé pour le développement

Dans ce projet, nous avons utilisé plusieurs outils de développement pour créer une application web robuste et sécurisée. Les principaux outils utilisés sont PostgreSQL pour la gestion de la base de données, XAMPP pour le serveur web, et PHP pour le

développement côté serveur. Cet ensemble d'outils offre une solution complète pour le développement, le test et le déploiement d'applications web.

PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO) open-source, réputé pour sa robustesse, son extensibilité et sa conformité aux standards SQL.

-Caractéristiques principales :

-Fiabilité et stabilité : PostgreSQL est connu pour sa stabilité et sa capacité à gérer de grandes charges de travail de manière efficace.

-Extensibilité : Il permet aux utilisateurs de définir leurs propres types de données, fonctions et opérations.

-Conformité SQL : PostgreSQL suit les normes SQL de manière stricte, assurant ainsi la portabilité des applications.

-Utilisation dans le projet :

-Stockage des données : PostgreSQL est utilisé pour stocker toutes les données des utilisateurs, des clients, ainsi que les adresses IP et les informations VLAN.

-Requêtes SQL : Les fonctions PHP interagissent avec PostgreSQL pour exécuter des requêtes SQL afin de manipuler les données (CRUD : Create, Read, Update, Delete).

XAMPP

XAMPP est une distribution Apache simple et légère qui rend extrêmement facile pour les développeurs de créer un serveur web local pour tester leurs applications web.

-Caractéristiques principales :

-Apache : Serveur HTTP utilisé pour héberger des applications web.

-MariaDB : SGBD inclus par défaut, bien que nous utilisions PostgreSQL pour ce projet.

-PHP : Langage de script côté serveur utilisé pour le développement de l'application.

-Perl : Langage de programmation également inclus.

-Interface utilisateur : XAMPP propose un tableau de bord simple pour gérer les composants du serveur.

-Utilisation dans le projet :

-Serveur web local : XAMPP fournit l'environnement serveur nécessaire pour exécuter et tester l'application web.

-Gestion des fichiers : Les fichiers PHP de l'application sont placés dans le répertoire `htdocs` de XAMPP, rendant l'application accessible via un navigateur à `http://localhost` .

PHP

PHP (Hypertext Preprocessor) est un langage de script open-source côté serveur, largement utilisé pour le développement web. PHP est particulièrement apprécié pour sa capacité à intégrer facilement du code HTML et interagir avec des bases de données.

-Caractéristiques principales :

-Facilité d'utilisation : Syntaxe simple et bien documentée, facilitant l'apprentissage et le développement rapide.

-Flexibilité : PHP peut être intégré dans des fichiers HTML, et supporte de nombreux SGBD, y compris PostgreSQL.

-Communauté : Large communauté de développeurs, offrant de nombreux frameworks et bibliothèques.

-Utilisation dans le projet :

-Développement côté serveur : PHP est utilisé pour traiter les requêtes des utilisateurs, interagir avec la base de données PostgreSQL et générer du contenu dynamique.

-Gestion des sessions : PHP gère les sessions utilisateur pour contrôler l'accès et maintenir l'état de connexion.

-Validation et traitement des formulaires : Les formulaires d'inscription, de connexion, d'ajout et de suppression sont traités par des scripts PHP.

Visual Studio Code

Visual Studio Code est un éditeur de code source développé par Microsoft pour Windows, Linux et macOS. Il comprend un support intégré pour Git, des fonctionnalités de débogage, une coloration syntaxique, la complétion de code intelligente, des extraits de code et des refactorings de code.

-Caractéristiques principales :

-Éditeur de code puissant : Supporte de nombreux langages de programmation et technologies.

-Extensions : Large bibliothèque d'extensions pour améliorer la productivité (PHP, PostgreSQL, etc.).

-Intégration Git : Facilite la gestion du code source et la collaboration via Git.

-Utilisation dans le projet :

-Écriture de code : Utilisé pour écrire et éditer les fichiers PHP, SQL, CSS et HTML.

-Débogage : Aide à identifier et corriger les erreurs dans le code.

-Gestion des extensions : Extensions spécifiques utilisées pour faciliter le développement en PHP et l'intégration avec PostgreSQL.

L'utilisation de PostgreSQL, XAMPP et PHP, combinée à un éditeur de code puissant comme Visual Studio Code, fournit une solution complète pour le développement d'applications web. Ces outils permettent de créer une application robuste, sécurisée et performante, tout en facilitant le processus de développement grâce à leurs fonctionnalités avancées et leur intégration fluide.

Conclusion

Le développement de ce site web a permis de créer une solution efficace pour la gestion des clients et des utilisateurs, en intégrant des technologies modernes et des pratiques de sécurité robustes. Grâce à PostgreSQL et PHP, nous avons pu concevoir une application web performante, sécurisée et évolutive. Les fonctionnalités mises en place assurent une gestion aisée des informations, tout en respectant les permissions spécifiques à chaque groupe d'utilisateurs.

Les défis rencontrés, notamment en ce qui concerne la gestion des adresses IP et la sécurité des données, ont été surmontés grâce à une planification minutieuse et à une mise en œuvre rigoureuse. Le site offre une expérience utilisateur fluide et sécurisée, tout en étant capable de s'adapter à des besoins futurs grâce à une architecture évolutive.

En conclusion, ce projet a permis de développer une application web solide et fiable, capable de répondre aux besoins actuels et futurs de l'entreprise en matière de gestion des clients et des utilisateurs. Les prochaines étapes envisagées incluent des améliorations de l'interface utilisateur, l'ajout de fonctionnalités supplémentaires, et un renforcement continu des mécanismes de sécurité, assurant ainsi la pérennité et l'efficacité de la solution.