**1. Description du Projet**

**Objectif : Développer une application de chat en ligne permettant à des utilisateurs de communiquer en temps réel. Le système doit supporter l'authentification des utilisateurs et permettre l'envoi de messages privés ou collectifs.**

**2. Besoins Utilisateurs**

* **Authentification : Les utilisateurs doivent pouvoir se connecter pour accéder au chat.**
* **Envoi de Messages : Les utilisateurs peuvent envoyer des messages à un utilisateur spécifique ou à tous les utilisateurs.**
* **Gestion des Utilisateurs : Les utilisateurs avec des droits administratifs (niveau 2) peuvent ajouter d'autres utilisateurs.**

**3. Spécifications Techniques**

* **Technologies Utilisées : PHP pour le backend, HTML et CSS pour le frontend, et MySQL pour la gestion de la base de données.**
* **Gestion de Session : Utilisation des sessions PHP pour maintenir l'état de connexion des utilisateurs.**

**4. Modélisation des Données**

**Base de Données**

* **Table user : Contient nom, autres détails des utilisateurs.**
* **Table messages : Stocke pseudo, message, date, destinataire.**

**Relations**

* **Un utilisateur peut envoyer plusieurs messages.**
* **Un message est destiné à un utilisateur spécifique ou à tous.**

**5. Diagrammes UML**

**Diagramme de Cas d'Utilisation**

* **Utilisateurs authentifiés : Envoyer des messages, lire des messages, se déconnecter.**
* **Utilisateurs administrateurs : Ajouter des utilisateurs.**

**Diagramme de Séquence**

* **Envoi de Message : Montre le processus depuis la saisie du message jusqu'à son enregistrement dans la base de données.**

**Diagramme de Classes**

* **Classe Utilisateur : Gère les informations et les actions des utilisateurs.**
* **Classe Message : Gère la création et la récupération des messages.**

**6. Interface Utilisateur**

**Maquettes**

* **Page de Connexion : Simple formulaire de connexion.**
* **Chat Principal : Zone d'affichage des messages et formulaire d'envoi de nouveaux messages.**

**7. Plan de Test**

* **Tests Fonctionnels : Vérifier que les utilisateurs peuvent se connecter, envoyer des messages, et que les administrateurs peuvent ajouter des utilisateurs.** **Une image contenant texte, Police, logiciel, capture d’écran

  Description générée automatiquement**Une image contenant texte, Police, capture d’écran

  Description générée automatiquement
* Une image contenant texte, capture d’écran, Police

  Description générée automatiquement
* **Une image contenant texte, capture d’écran, Police, conception

  Description générée automatiquement**
* Une image contenant texte, capture d’écran, Police

  Description générée automatiquement
* Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

  Description générée automatiquement

**8. Document de Synthèse**

**Résumé du Projet**

**Le projet consistait à développer une application de chat en ligne utilisant PHP et SQL, permettant aux utilisateurs de communiquer en temps réel à travers une interface web. Les principales fonctionnalités incluent l'authentification des utilisateurs, l'envoi de messages privés ou à tous les utilisateurs, et des privilèges administratifs pour ajouter des utilisateurs.**

**Technologies Impliquées :**

* **Frontend : HTML et CSS pour la mise en page et le style.**
* **Backend : PHP pour la logique métier et la gestion des sessions.**
* **Base de données : MySQL pour le stockage des données utilisateur et des messages.**

**Méthodologie de Développement : Le développement a suivi une approche agile, avec des itérations courtes permettant des ajustements rapides en fonction des retours des utilisateurs testeurs. Le code a été versionné avec Git, facilitant la collaboration et le suivi des modifications.**

**Réflexions**

**Challenges Rencontrés et Solutions Apportées :**

1. **Sécurité - Prévention des Injections SQL :**
   * **Challenge : Le système était initialement vulnérable aux injections SQL, un risque majeur pour l'intégrité de la base de données.**
   * **Solution : Utilisation de requêtes préparées dans PHP pour sécuriser toutes les interactions avec la base de données. Cela a non seulement sécurisé l'application contre les injections SQL mais a également rendu le code plus robuste et moins sujet aux erreurs.**
2. **Gestion de la Base de Données - Optimisation des Requêtes :**
   * **Challenge : Les requêtes initiales étaient inefficaces, causant des latences et des charges élevées sur le serveur de base de données lors de l'accès simultané par de nombreux utilisateurs.**
   * **Solution : Révision et optimisation des requêtes SQL pour améliorer la performance. L'indexation des colonnes fréquemment recherchées, comme les destinataires et les dates des messages, a également été mise en place pour accélérer les recherches.**
3. **Gestion des Sessions :**
   * **Challenge : Les sessions PHP ne se terminaient pas correctement après la déconnexion des utilisateurs, posant des risques de sécurité potentiels.**
   * **Solution : Implémentation d'un mécanisme de déconnexion robuste qui efface et régénère la session, assurant ainsi que les informations d'authentification ne peuvent pas être exploitées après la déconnexion.**

**Réflexions Personnelles : Le projet a été une excellente opportunité pour appliquer des compétences théoriques en développement web et gestion de bases de données dans un contexte pratique. La nécessité de penser à la sécurité dès le début du développement a été une leçon précieuse, soulignant l'importance de coder de manière défensive. En outre, le projet a permis de renforcer la collaboration en équipe et les compétences en résolution de problèmes, essentielles pour la gestion des imprévus techniques et des exigences changeantes.**