

# CAHIER DES CHARGES

**Projet :** Plateforme Web de Tchatche Multi-Salles

## 1. Présentation Générale du Projet

Le projet consiste à concevoir et développer un site web de discussion en temps réel (tchatche) comportant plusieurs salles thématiques. L'objectif est de permettre aux utilisateurs de choisir une salle en fonction de leurs intentions relationnelles (amitié, rencontre, mariage, etc.), de créer un profil personnalisé et de discuter avec d'autres membres connectés.

## 2. Objectifs du Projet

### 2.1 Objectif Principal

Développer une application web de tchatche multi-salles fonctionnelle, intuitive, sécurisée et évolutive.

### 2.2 Objectifs Spécifiques

- Implémenter un système d'inscription et d'authentification sécurisé.
- Permettre aux utilisateurs de créer et gérer un profil personnalisé.
- Développer une interface utilisateur intuitive.
- Mettre en place un système de tchat en temps réel performant.
- Créer des salles thématiques claires et bien identifiées.
- Permettre une navigation fluide entre les salles.
- Intégrer des outils de modération efficaces.
- Garantir la scalabilité et la maintenabilité de l'application.

## 3. Fonctionnalités Clés

### 3.1 Gestion des Utilisateurs et des Profils

- Inscription / Connexion avec validation des données.
- Création / modification du profil :
  - Nom d'utilisateur
  - Âge
  - Sexe

- Centres d'intérêt
- Intentions relationnelles
- Photo de profil (optionnelle)
- Localisation (optionnelle)
- Recherche d'autres utilisateurs.
- Blocage / Ignorance d'utilisateurs indésirables.

### **3.2 Gestion des Salles Thématiques**

- Affichage de la liste des salles disponibles (Amis, Rencontres, Mariage, etc.).
- Description brève de chaque salle.
- Nombre d'utilisateurs connectés visible par salle.

### **3.3 Système de Tchat en Temps Réel**

- Envoi / Réception de messages instantanés.
- Affichage du message avec nom d'utilisateur et heure.
- (Optionnel) Messages privés dans une même salle.
- (Optionnel) Support des emojis.

### **3.4 Interface Utilisateur**

- Navigation simple entre :
  - Liste des salles
  - Profil utilisateur
  - Interface de tchat
- Design responsive (adapté aux PC, tablettes, smartphones).

### **3.5 Modération**

- Possibilité de signaler des messages (optionnel première version).
- Bannissement temporaire ou définitif par les modérateurs.
- Les modérateurs **ne peuvent pas accéder aux données personnelles**.

## **4. Analyse des Besoins et Conception**

### **4.1 Préférences Utilisateurs à Considérer**

- Navigation simple, rapide et claire.

- Confidentialité des données.
- Communication fluide.
- Design sobre et attrayant.

## **4.2 Définition des Fonctionnalités**

Les fonctionnalités sont priorisées ainsi :

- MVP (Minimum Viable Product) : Tchat en temps réel, profils, multi-salles, authentification.
- Améliorations futures : Messages privés, emojis, signalement, notifications.

## **4.3 Architecture du Site**

- Frontend (Client) : Interface utilisateur dynamique
- Backend (Serveur) : API REST + WebSocket
- Base de données : Stockage utilisateurs, profils, messages, salles

## **4.4 Choix Technologiques (Proposition)**

- Frontend : React.js, Vue.js ou Svelte (Design responsive avec Tailwind CSS ou Bootstrap)
- Backend : Node.js (Express) ou Python (FastAPI)
- Temps réel : Socket.IO (Node) ou WebSocket natif
- Base de données : PostgreSQL ou MongoDB
- Authentification : JWT (JSON Web Token)
- Déploiement : Docker + hébergement (VPS, Render, Vercel, ou autre)

## **5. Phases de Développement**

### **5.1 Phase 1 : Spécifications Techniques et Maquettes**

- Rédaction des spécifications techniques détaillées.
- Conception des wireframes et de l'interface.

## **5.2 Phase 2 : Développement Backend**

- Mise en place de la base de données.
- Développement des routes API (utilisateur, salle, message).
- Intégration WebSocket pour le tchat.

## **5.3 Phase 3 : Développement Frontend**

- Intégration des interfaces utilisateurs.
- Connexion à l'API backend.
- Intégration du tchat en temps réel.

## **5.4 Phase 4 : Tests**

- Tests unitaires (backend).
- Tests d'intégration.
- Tests utilisateurs (UX/UI).
- Tests de performance (temps réel).

## **5.5 Phase 5 : Déploiement**

- Mise en ligne de l'application (hébergeur cloud ou VPS).
- Configuration SSL, sécurité, sauvegardes.

## **6. Contraintes Techniques et Sécuritaires**

- Protection des données utilisateurs (RGPD ou équivalent).
- Mots de passe chiffrés (bcrypt).
- Filtrage des messages pour éviter les abus.
- Prévention des attaques XSS/CSRF/Injection SQL.

## **7. Livrables Attendus**

- Code source documenté (frontend et backend).
- Base de données initiale avec tables/schemas.
- Documentation technique et d'installation.
- Rapport de tests.
- Site web fonctionnel déployé.

## **8. Maintenance et Évolutivité**

- Application conçue pour gérer plusieurs utilisateurs simultanés.
- Possibilité de créer de nouvelles salles sans refonte.
- Mise à jour des dépendances via outils CI/CD.
- Interface d'administration pour la gestion du contenu.