|  |
| --- |
| Stage WEDOO : Cahier des charges du développement |

**Répartition des équipes travaillant sur le pré-site :**

Front End (Design) : Andrew MUTHUGALA, Yamissa FOFANA, Adam CISSÉ, Rayan NADJI, Kévin WANG

Back End (Architecture du site) : Mohamed GAB ALLA, Abinash ROY, Amath DIATTA, Marie ZABOLO

**1. PRÉSENTATION GÉNÉRALE DU PROJET**

1.1 Présentation du projet WeDoo

Projet de réseau social visant à maximiser le potentiel d'internet pour relier et partager des connaissances et des sensibilités.

Objectif de créer un espace pour l’amusement, la joie, et la réflexion collective.

1.2 Cible et Accessibilité

**Pré-site orienté pour un public unilingue (France).**

Cible : Tout public avec une attention particulière à l’accessibilité pour les personnes en situation de handicap.

1.3 Objectifs du Mini-Site

**Qualitatifs :** Présentation du projet, initiation via une fonctionnalité (Club des Chipies), et collecte de retours pour le futur site.

**Quantitatifs** : Suivi via Google Analytics, KPIs tels que le nombre de visiteurs uniques, temps de visite, provenance, taux de rebond.

**2. ASPECTS DESIGN & ERGONOMIE**

2.1 Design

Collaboration avec un tiers pour le design.

Charte graphique en développement (logo, typographie, couleurs, images, icônes).

2.2 Ergonomie

Single Page Application avec site responsive.

Éco-conception.

Évaluation de l’intérêt d’une application mobile.

**3. ASPECTS FONCTIONNELS & TECHNIQUES**

3.1 Arborescence

Sections incluant présentation, Club des Chipies, Inscription, Charte, Module d’interaction, Questionnaires, Générique.

3.2 Description Fonctionnelle

Fonctionnalités pour utilisateurs (partage, création de compte, interaction) et administrateurs (modification du contenu).

3.3 Contenu

Textes, dessins, vidéos, avec une réflexion sur le SEO.

3.4 Contraintes Techniques

Facilité de modification, compatibilité navigateurs, gestion du nom de domaine et de l’hébergement.

3.5 Architecture Technique

Langages de programmation et Frameworks spécifiques.

Sécurité des données et stratégies de sauvegarde.

3.6 Base de données

Choix du SGBD et modélisation de la base de données.

Cryptage des données sensibles.

3.7 API et Intégrations

Documentation des API internes et intégration des services tiers.

3.8 Sécurité et Conformité

Protection contre les attaques web, conformité RGPD, systèmes d'authentification.

3.9 Performance et Scalabilité

Techniques d’optimisation frontend, gestion de la charge et scalabilité.

3.10 Tests et Qualité

Cadre pour les tests automatisés et processus de contrôle qualité.

**4. PRESTATIONS ATTENDUES**

4.1 Création et Hébergement

Création du mini-site et collaboration avec un web designer.

Définition de l’hébergement.

4.2 Planning et Suivi

Définition du planning, modalités d'échanges, aspects juridiques et règlementaires.

4.3 Maintenance et Support

Plan de maintenance et mise en place d'une équipe de support technique.

4.4 Formation et Documentation

Formation des utilisateurs et administrateurs, documentation technique.

4.5 Architecture Technique

**Langages de Programmation :** JavaScript pour le frontend, PHP/Node.js pour le backend).

**Frameworks :** Choix des frameworks adaptés (ex : React ou Angular pour le frontend, Laravel ou Express pour le backend).

4.6 Base de Données

**Système de Gestion de Base de Données (SGBD) :** Choix du SGBD (ex : MySQL, PostgreSQL).

**Modélisation :** Schéma de la base de données, relations entre les tables.

**Sécurité :** Méthodes de cryptage des données sensibles, stratégies de sauvegarde.

4.7 API et Intégrations

**APIs Internes :** Description des API pour la communication entre le frontend et le backend.

**APIs Externes** : Intégration avec des services tiers (ex : Google Analytics, API de réseaux sociaux pour le partage).

Documentation API : Création d'une documentation claire pour les développeurs (ex : Swagger)

4.8 Sécurité et Conformité

**Sécurité Web :** Stratégies pour la protection contre les attaques courantes (ex : injection SQL, XSS).

**Conformité RGPD :** Mesures pour assurer la conformité avec les lois sur la protection des données.

**Authentification :** Systèmes d'authentification et de gestion des sessions utilisateurs.

4.9 Performance et Scalabilité

**Optimisation Frontend :** Techniques pour réduire le temps de chargement (ex : minification, lazy loading).

**Gestion de la Charge :** Stratégies pour la scalabilité et la gestion du trafic (ex : utilisation de serveurs de cache, équilibrage de charge).

4.10 Tests et Qualité

**Tests Unitaires et d’Intégration :** Cadre pour les tests automatisés.

**Tests de Performance :** Tests de charge pour valider la performance sous trafic élevé.

**Assurance Qualité :** Processus de contrôle qualité avant les déploiements.

4.11 Maintenance et Support

**Maintenance :** Plan de maintenance régulière (mises à jour, correction de bugs).

**Support Technique :** Mise en place d'une équipe de support pour les problèmes techniques.

4.12 Formation et Documentation

**Formation des Utilisateurs/Administrateurs :** Sessions de formation pour les utilisateurs du backend.

**Documentation technique :** Mise à disposition d'une documentation complète pour le site.