Cahier des charges – Application Web Parc National

1. Présentation de l’entreprise commanditaire

- **Nom de l’organisation :** Parc National
- **Secteur d’activité :** Gestion et valorisation des espaces naturels protégés
- **Objectif principal :** Moderniser la gestion interne et améliorer l’expérience des visiteurs grâce à une application web sécurisée et fiable.

---

2. Description de l’existant

Actuellement, la gestion repose sur des processus manuels :

- Inscriptions papier des visiteurs.
- Réservations de campings sur Excel.
- Suivi des sentiers et ressources naturelles non centralisé.

Problèmes identifiés :

- Risques d’erreurs humaines.
- Difficultés à obtenir des statistiques fiables.
- Processus long et inefficace.

---

3. Objectifs du futur site web

L’application devra :

- Gérer les **visiteurs** (création, modification, suppression).
- Administrer les **sentiers** (difficulté, longueur, accessibilité).
- Permettre la **réservation en ligne des campings**.
- Suivre les **ressources naturelles** (faune, flore, zones protégées).
- Fournir un **dashboard statistique** clair.
- Garantir la **sécurité et la confidentialité des données personnelles**.

---

4. Exigences en matière de design

- Interface responsive (mobile, tablette, desktop).
- Navigation claire et intuitive.
- Codes couleurs pour identifier sentiers et zones de camping.
- Dashboard ergonomique avec graphiques et indicateurs.

---

5. Fonctionnalités attendues

5.1 Gestion des visiteurs

- Ajouter, modifier et supprimer un visiteur.
- Recherche par nom, email ou numéro d'inscription.
- Stockage sécurisé des informations sensibles (`mot_de_passe`, `email`, `carte_membre`).

5.2 Gestion des sentiers

- CRUD (ajouter, modifier, supprimer).
- Informations : difficulté, longueur, accessibilité.
- Visualisation cartographique.

5.3 Réservations de campings

- Réserver en ligne.
- Annuler / modifier une réservation.
- Vérification disponibilité en temps réel.
- Gestion sécurisée du **statut** de la réservation (protégé).

5.4 Gestion des ressources naturelles

- CRUD sur les ressources.
- Catégorisation (faune, flore, zones protégées).
- Suivi des interventions et observations.

5.5 Administration et statistiques

- Dashboard de suivi (visiteurs, réservations, sentiers).
- Export CSV / PDF.
- Gestion des rôles et permissions (`admin`, `visiteur`).

---

6. Pré-requis techniques

- **Langages :** PHP (backend), React+Vite (frontend).
- **Base de données :** MySQL / MariaDB.
- **Tests :** PHPUnit (backend), Jest ou Cypress (frontend).
- **CI/CD :** GitHub Actions.
- **Frameworks :** Laravel/Symfony, Bootstrap/React.js.

---

7. SEO et accessibilité

- Respect normes W3C.
- Référencement naturel optimisé.
- Accessibilité (contraste, navigation clavier).

---

8. Sécurité

- **Encapsulation des données sensibles :**
  - `mot_de_passe` en `private` avec hash (`password_hash`, `password_verify`).
  - `email`, `role`, `carte_membre`, `statut` en `protected`.
- Protection contre injections SQL et XSS.
- Gestion stricte des rôles et permissions.
- Sauvegardes automatiques de la BDD.
- Journalisation des accès aux données sensibles.

---

9. Planning des tests

- **Semaine 3 :** Tests unitaires backend (≥ 80% couverture).
- **Semaine 4 :** Tests unitaires frontend.
- **Semaine 5 :**
    - J1-J3 → Tests d'intégration.
    - J4-J5 → Tests de performance (temps réponse < 3s, 100 utilisateurs simultanés).
    - J6 → Tests de sécurité (OWASP ZAP, aucune faille critique).
    - J7-J8 → Tests d'acceptation utilisateur (95% validés).
    - J9-J10 → Corrections des anomalies.

Outils : PHPUnit, Jest/Cypress, JMeter, OWASP ZAP, Selenium.

---

10. Contraintes légales

- Conformité **RGPD** (protection des données personnelles).
- Respect de la propriété intellectuelle (textes, images).

---

11. Planning du projet

- **Semaine 1 :** Conception (cahier des charges, MCD, wireframes, users stories).
- **Semaine 2 :** Installation (serveur, BDD, GitHub, tests).
- **Semaines 3-4 :** Développement (front/back).
- **Semaine 5 :** Tests et débogage.
- **Semaine 6 :** Livraison et soutenance.

---

12. Cahier de recette

- Pour chaque fonctionnalité : scénario, résultat attendu, validation.
- Exemple :
    - **Fonction :** Ajouter un visiteur
    - **Scénario :** Saisie des infos + clic sur “Ajouter”
    - **Résultat attendu :** Ajout en BDD et affichage dans la liste
    - **Validation :** PHPUnit + test interface

---

13. Livrables attendus

- Code source complet sur GitHub (branches documentées).
- Base SQL initialisée + données tests.
- Wireframes / maquettes Figma.
- Cahier de tests et résultats.
- Présentation finale et support soutenance.