

# Cahier des charges — Site web « Groupe Jeunesse & Camp annuel »

## 1. Contexte & objectifs

Le site a pour objectif de : - Présenter le groupe de jeunesse des différents campus de l'église (site vitrine).

- Permettre l'inscription des jeunes au camp annuel, gérer les activités individuelles, et fournir un planning personnalisé.
- Gérer les paiements pour accéder au camp (verrouillage du programme / inscriptions tant que le paiement n'est pas effectué).
- Générer et envoyer automatiquement des PDF (programme / tickets / planning) par e-mail après inscription / paiement.
- Offrir un suivi **temps réel** des activités pendant le camp et recueillir des retours à la fin.

**Principaux bénéfices attendus** : meilleure visibilité du mouvement jeunesse, automatisation des inscriptions et paiements, expérience utilisateur fluide et personnalisée.

---

## 2. Périmètre fonctionnel (découpage en étapes)

**Phase A — Site vitrine (minimum viable)** - Page d'accueil, pages campus, page « À propos », page du camp (infos générales, dates, lieu, FAQ).

- Galerie photos/vidéos et contacts.

**Phase B — Inscription & activités** - Création de compte utilisateur (vérification d'unicité de l'email + email de confirmation).

- Formulaire d'inscription au camp (données personnelles, campus, préférences).
- Liste d'activités avec description, capacité, durée, prix (si applicable).
- Sélection d'activités individuelles pour générer un programme personnalisé. - Téléchargement et envoi automatique du programme personnalisé au format PDF.

**Phase C — Paiement & contrôle d'accès** - Paiement en ligne sécurisé (intégration Stripe/PayPal ou équivalent).

- Verrouillage du détail du programme et de la possibilité d'inscrire des activités tant que le paiement n'est pas validé.
- Webhooks pour marquer le paiement comme "réussi" et déclencher envois (email + PDF).

**Phase D — Suivi temps réel (pendant le camp)** - Page/section « Live » réservée aux participants (authentification requise).

- Mise à jour en temps réel des activités (live status, salles, scores/points, événements).
- Notifications in-app / push pour changements d'horaires ou annonces.

**Phase E — Bilan & retours** - Formulaire de feedback par activité et global.

- Dashboard admin pour consulter/exporter les retours.

### 3. Acteurs / rôles

- **Visiteur** : consulte le site vitrine.
  - **Inscrit / Participant** : crée un compte, s'inscrit au camp, sélectionne activités, paye, accède au programme et au live.
  - **Responsable campus** : peut consulter les inscriptions de son campus, valider participants.
  - **Administrateur (équipe organisatrice)** : gestion complète (création d'activités, capacités, prix, paiements, exports, communication).
  - **Finance/Trésorier** : accès aux rapports paiements, remboursements.
- 

### 4. User stories & critères d'acceptation (exemples clés)

Chaque user story doit être accompagnée d'au moins un cas d'acceptation testable.

#### 4.1 Inscription & vérification

**US-1** : En tant que visiteur, je peux créer un compte avec email et mot de passe.

**AC** : L'email est vérifié unique avant création ; un email de confirmation est envoyé contenant un lien d'activation. Compte inactif tant que l'email n'est pas confirmé.

#### 4.2 Inscription au camp

**US-2** : En tant qu'utilisateur confirmé, je peux remplir le formulaire d'inscription au camp.

**AC** : Tous les champs requis validés ; sauvegarde et message de confirmation envoyé ; génération d'un récapitulatif.

#### 4.3 Sélection d'activités & planning

**US-3** : En tant que participant, je peux sélectionner des activités individuelles pour construire mon planning.

**AC** : Le système vérifie les conflits d'horaire, la capacité restante et n'autorise pas l'inscription si non payée (si règle applic.). Le planning personnalisé est téléchargeable en PDF et envoyé par email.

#### 4.4 Paiement

**US-4** : En tant que participant, je peux payer en ligne pour confirmer mon accès au camp.

**AC** : Lorsqu'un paiement est validé par webhook, l'accès au programme et la possibilité d'enregistrer des activités sont débloqués. En cas d'échec, message d'erreur clair et possibilité de réessayer.

#### 4.5 Suivi en temps réel

**US-5** : En tant que participant, je peux voir le déroulé des activités en direct lors du camp.

**AC** : Les changements d'état (commencé/retardé/terminé) sont poussés aux clients authentifiés sans recharge. Les informations affichées correspondent à celles saisies/validées par l'admin.

#### 4.6 Feedback

**US-6** : Après le camp, je peux déposer un avis/notation pour chaque activité.

**AC** : Les retours sont stockés et exportables par l'admin ; les utilisateurs reçoivent confirmation d'envoi.

---

## 5. Pages & navigation (sitemap simplifié)

- Accueil
  - À propos / Nos campus
  - Camp annuel (infos & FAQ)
  - Activités (catalogue)
  - S'inscrire / Mon compte / Mon planning
  - Paiement
  - Live (accessible aux participants)
  - Dashboard admin (invitations, participants, activités, paiements, exports)
  - Politique de confidentialité / Conditions / Contact
- 

## 6. Flux principaux (résumé)

1. Visiteur consulte le site → clique sur S'inscrire → crée un compte → reçoit email de confirmation → confirme.
  2. Utilisateur connecté complète inscription au camp → sélection d'activités (sauvegardées en brouillon).
  3. Utilisateur paie → webhook confirme paiement → système marque inscription comme "payée" → envoi automatique de PDF + email de bienvenue + accès au contenu premium (programme complet & inscription activités finale).
  4. Pendant le camp, l'admin pousse les mises à jour → participants reçoivent en temps réel.
  5. Après le camp, mail de rappel pour faire un feedback.
- 

## 7. Modèle de données (schéma simplifié)

- **User:** id, prénom, nom, email (unique), password\_hash, campus\_id, phone, role, verified, created\_at
  - **Campus:** id, nom, adresse, contact
  - **Camp:** id, année, date\_debut, date\_fin, lieu, description
  - **Activity:** id, camp\_id, titre, description, heure\_debut, heure\_fin, lieu, capacite, prix, responsable\_id
  - **Registration:** id, user\_id, camp\_id, statut (brouillon/en\_attente\_paiement/payé/annulé), created\_at
  - **ActivitySignup:** id, registration\_id, activity\_id, statut
  - **Payment:** id, registration\_id, provider, montant, currency, status, provider\_reference, created\_at
  - **Feedback:** id, user\_id, activity\_id, note, commentaire, created\_at
- 

## 8. Intégrations recommandées

- **Paiement :** Stripe (recommandé) ou PayPal — gestion webhooks, remboursements.
- **Emails :** SendGrid, Mailgun ou Postmark pour délivrabilité + templates.
- **Fichiers / PDF :** génération server-side (Puppeteer, wkhtmltopdf, ou librairie PDF native) + stockage S3/MinIO.
- **Temps réel :** WebSocket / Socket.IO ou service PaaS (Pusher, Ably) pour scalabilité.

- **Hébergement** : Frontend sur Vercel/Netlify, backend sur Heroku/Render/AWS/GCP selon budget.
- 

## 9. Sécurité & conformité (RGPD)

- HTTPS obligatoire (TLS).
  - Ne pas stocker de données bancaires : utiliser un prestataire externe (Stripe Elements / Checkout).
  - Consentement explicite au stockage des données (case à cocher).
  - Possibilité de suppression/export des données personnelles (conformité RGPD).
  - Mots de passe hachés (bcrypt/argon2).
  - Protection contre injections, XSS, CSRF.
  - Politique de confidentialité et mentions légales présentes et accessibles.
- 

## 10. Non-fonctionnel

- Responsive (mobile / tablette / desktop).
  - Accessibilité : viser WCAG 2.1 AA.
  - Performance : pages vitrine rapides (bonnes pratiques SEO).
  - Log et monitoring (Sentry, Datadog/Logflare).
  - Sauvegardes régulières de la BDD et des fichiers.
- 

## 11. Tests & recette

- Tests unitaires/back-end, tests d'intégration, tests end-to-end (Cypress / Playwright).
  - Scénarios de recette pour chaque user story (inscription, paiement, génération PDF, cas d'erreur paiements, capacité pleine, conflit d'horaire).
  - Tests d'acceptation pour RGPD (export/suppression données), accessibilité et load tests sur page d'inscription.
- 

## 12. Livrables

- Cahier des charges (ce document).
  - Maquettes Figma (wireframes + UI).
  - API spec (OpenAPI / Postman collection).
  - Code source (frontend + backend) avec Dockerfile.
  - Scripts de déploiement / guide d'exploitation.
  - Plan de tests & rapport de recette.
- 

## 13. Priorisation / MVP (proposition)

**MVP (phase minimale)** - Site vitrine + page camp.

- Inscription utilisateur + vérification email.

- Formulaire d'inscription au camp (brouillon).
- Paiement en ligne et déblocage du programme.
- Envoi automatique d'un PDF récapitulatif après paiement.

**Améliorations post-MVP** - Sélection avancée d'activités et planning automatique.

- Fonctionnalités live (temps réel).
  - Dashboard avancé admin / exports / statistiques.
- 

## 14. Annexes pratiques

- **Templates d'emails** : confirmation d'inscription, rappel de paiement, ticket (PDF), lien d'activation, relance feedback.
  - **Gestion des erreurs paiements** : proposer plusieurs moyens de paiement et possibilité d'inscription manuelle (admin) en cas de problème.
  - **Capacité et liste d'attente** : quand une activité est pleine, proposer la mise en liste d'attente et notification si place libérée.
- 

### Prochaines étapes proposées (tu peux choisir une ou plusieurs)

1. Traduire ce cahier en backlog technique (tâches / tickets).
  2. Créer les wireframes Figma (pages principales + composants).
  3. Préparer l'API spec (endpoints principaux + schéma de données).
  4. Prototype de l'écran d'inscription + flux de paiement en maquette interactive.
- 

*Fin du cahier des charges.*