Cahier des Charges

Qonect

Yoann Mallet - Sohane Chaman - Mareva Dumain

Sommaire

Qonect **1**
1. Présentation générale du projet **3**
2. Objectifs du projet **3**
3. Fonctionnalités principales **4**
Côté Utilisateur : 4
Côté Référent : 4
Côté Administrateur : 5
Fonctionnalités additionnelles : 5
4. Public cible **5**
5. Parcours utilisateur **5**
6. Technologies utilisées **6**
● Front-end : 6
● Back-end et BDD : 6
● Outils & Hébergement : 6
7. Schéma de la base de données **6**
8. Planning prévisionnel **7**
9. Contraintes **7**
10. Gestion des risques **8**
11. Livrables prévus **8**
12. Conclusion **9**

1. Présentation générale du projet

Nom du projet : Konect
Équipe projet : Yoann Mallet, Maréva Dumain, Sohanne Chamen
Liens utiles :

- **Prototype Figma :** Figma – Qonect
- **Site en production :** Qonect

Contexte :
Konect est une plateforme web visant à faciliter la gestion des communautés et des événements. Elle permet aux utilisateurs de rejoindre des communautés, de s'inscrire à des événements, et aux référents et administrateurs de gérer facilement l'ensemble des activités.

2. Objectifs du projet

- Créer une plateforme responsive et moderne qui permet :
    - Aux **utilisateurs** de rechercher, rejoindre et participer à des événements.
    - Aux **référents** de créer, éditer et valider des événements et des communautés.
    - Aux **administrateurs** de superviser et de modérer toutes les entités (utilisateurs, événements, communautés).
- Livrer un **MVP** fonctionnel et hébergé (Vercel) avec une **base de données Supabase**.
- Proposer une interface claire et intuitive conçue à partir des maquettes Figma.

3. Fonctionnalités principales

Côté Utilisateur :

- Inscription et connexion.
- Accès au catalogue de communautés et d'événements.
- Inscription à un événement ou à une communauté.
- Profil utilisateur (informations personnelles et historique).

Côté Référent :

- Création, modification et suppression (CRUD) d'événements.
- Gestion des membres d'une communauté.
- Validation des inscriptions aux événements.

Côté Administrateur :

- Gestion complète des communautés, événements et utilisateurs.
- Tableau de bord centralisé.
- Statistiques et vue globale des activités.

Fonctionnalités additionnelles :

- Système de notifications et alertes.
- Recherche par mots-clés ou catégories.
- Interface responsive (mobile-first).

4. Public cible

- **Utilisateurs réguliers** : personnes recherchant des communautés ou des événements selon leurs centres d'intérêts.
- **Référents** : créateurs et organisateurs d'événements.
- **Administrateurs** : modérateurs supervisant l'ensemble des entités.

5. Parcours utilisateur

1. **Un utilisateur s'inscrit ou se connecte** via un formulaire.
2. **Il rejoint une communauté** en consultant sa fiche détaillée.
3. **Il s'inscrit à un événement** (ou se désinscrit).
4. **Le référent** valide ou modifie les informations.
5. **L'admin** garde un contrôle global (ajout/suppression de communautés ou événements).

6. Technologies utilisées

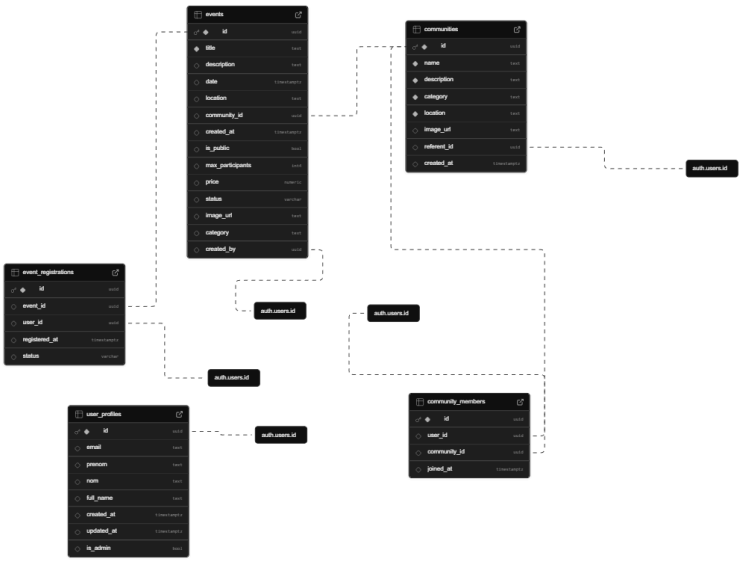
- Front-end :
  - HTML5, CSS3, JavaScript Vanilla.
  - Responsive design (Flexbox/Grid).
- Back-end et BDD :
  - **Supabase** (PostgreSQL, API REST).
  - Gestion des rôles et authentification intégrée.
- Outils & Hébergement :
  - **Figma** pour le design et l'UX.
  - **GitHub** pour le versioning.

- **Vercel** pour le déploiement en production.
- **Discord** et **Kanban GitHub** pour la gestion de projet.

7. Schéma de la base de données

La base de données Supabase se compose de 6 tables :

- **user_profiles** : données des utilisateurs.
- **communities** : gestion des communautés.
- **events** : gestion des événements.
- **community_members** : lien entre utilisateurs et communautés.
- **event_registrations** : inscriptions aux événements.
- **auth.users** : gestion de l'authentification.



8. Planning prévisionnel

Sprint	Objectifs clés
S1	Cadrage du projet, cahier des charges, maquettes Figma.
S2	Mise en place de la BDD Supabase et authentification.
S3	CRUD communautés + design pages principales.
S4	CRUD événements + Dashboard admin.
S5	Tests, recettage, mise en production (Vercel).

9. Contraintes

- **Techniques :** JavaScript Vanilla (pas de framework), utilisation de Supabase.
- **UI/UX :** Responsive et conforme aux maquettes Figma.
- **Sécurité :** Gestion des droits d'accès (utilisateur, référent, admin).
- **Production :** Hébergement Vercel avec disponibilité >2 semaines après soutenance.

10. Gestion des risques

Risque	Impact	Probabilité	Solution
Retards de développement	Élevé	Moyen	Sprint hebdomadaire + Kanban.
Bug API Supabase	Moyen	Faible	Tests et sauvegardes fréquentes.
UI non responsive	Moyen	Moyen	Tests multi-supports.
Perte de données	Faible	Faible	Backups et versioning GitHub.

11. Livrables prévus

- Cahier des charges (PDF).
- **Prototype Figma interactif**.
- **Code source complet** (hébergé sur GitHub).
- **README.md** (installation, lancement, liens).
- **Schéma BDD + MCD/UML**.
- **Rapport d'audit technique (C4.10)**.
- **Procédure de déploiement (C4.8)**.
- **PV de recettage (C4.3)**.

12. Conclusion

Qonect est pensé comme une plateforme simple, efficace et évolutive. Le MVP couvre la gestion des communautés et événements, avec un socle technique solide (Supabase + JavaScript Vanilla) et un design validé sur Figma. Ce cahier des charges constitue la base de référence pour la finalisation du projet.