

# PROJET MONCHŒUR

Plateforme Web de Chants Choraux pour la Messe

*Document de Structuration Complète*

3 Février 2026

## **Table des Matières**

### **1. PHASE 1 : INITIALISATION ET CADRAGE DU PROJET**

- 1.1. Contexte et Problématique
- 1.2. Objectifs (Général et Spécifiques)
- 1.3. Acteurs et Utilisateurs
- 1.4. Analyse QQCCP
- 1.5. Analyse SWOT
- 1.6. Analyse de Faisabilité

### **2. PHASE 2 : DÉFINITION DES EXIGENCES**

- 2.1. Besoins Fonctionnels
- 2.2. Besoins Non Fonctionnels
- 2.3. Priorisation MoSCoW

### **3. PHASE 3 : CONCEPTION ET PLANIFICATION**

- 3.1. Architecture Technique
- 3.2. Structure de Décomposition (WBS)
- 3.3. Modélisation UML
- 3.4. Maquettage et Prototypage
- 3.5. Planning Agile (Sprints)

### **4. PHASE 4 : RÉALISATION ET PILOTAGE**

- 4.1. Méthodologie Agile
- 4.2. Backlog et User Stories
- 4.3. Gestion des Versions (Git)

### **5. LIVRABLES ET DOCUMENTATION**

## 6. ANNEXES

# PHASE 1 : INITIALISATION ET CADRAGE DU PROJET

## 1.1. Contexte et Problématique

### Nom du Projet

MonChœur (alternatives : Choralia, Gloria in Voce)

### Thème

Site web exhaustif de tous les chants de choral à la messe, organisés selon les voix (Soprano, Alto, Ténor, Basse).

### Contexte

Actuellement, l'apprentissage des chants de la chorale se fait de manière traditionnelle et présente plusieurs limitations :

- Les chants sont appris en personne, limitant l'apprentissage à la disponibilité des répétitions
- L'apprentissage dépend de quelqu'un qui connaît déjà les chants
- La recherche des paroles est fastidieuse et décentralisée
- Les voix spécifiques (particulièrement les voix de basse) sont difficiles à trouver
- Il n'existe pas de ressource centralisée et structurée pour l'apprentissage autonome

### Problématique

*Comment créer une plateforme web qui permette à toute personne (particulièrement les choristes débutants et les voix de basse) d'apprendre facilement les chants de la chorale, avec accès aux paroles, aux partitions et aux enregistrements audio par voix ?*

## 1.2. Objectifs (Général et Spécifiques)

### Objectif Général

Créer un site web pour recueillir, organiser et diffuser les chants de la chorale afin de faciliter leur apprentissage.

### Objectifs Spécifiques

**OS1 :** Faciliter l'apprentissage des chants de la chorale grâce à des ressources structurées et accessibles

**OS2 :** Permettre à quiconque veut apprendre (le peuple de Dieu) d'y arriver de manière autonome

**OS3 :** Apprendre aux personnes à chanter à travers des ressources audio par voix et des vocalises

**OS4 :** Constituer une base de données exhaustive et pérenne des chants pour les générations futures

**OS5 :** Offrir un outil de référence aux chefs de chœur pour organiser les chants selon la liturgie

## 1.3. Acteurs et Utilisateurs

Rôle	Description
<b>Utilisateurs Finaux (Choristes)</b>	<ul style="list-style-type: none"><li>Débutants qui ne savent pas chanter</li><li>Choristes confirmés cherchant des partitions</li><li>Personnes voulant apprendre seules chez elles</li></ul>
<b>Contributeurs</b>	<ul style="list-style-type: none"><li>Administrateur principal (vous)</li><li>Administrateurs de confiance</li><li>Chanteurs bénévoles (4 voix : S, A, T, B)</li></ul>
<b>Chefs de Chœur</b>	<ul style="list-style-type: none"><li>Organisateurs de la liturgie</li><li>Sélectionneurs de chants pour les messes</li></ul>
<b>Développeur</b>	<ul style="list-style-type: none"><li>Vous (niveau débutant)</li><li>Assistance par Google Antigravity (IA Gemini 3)</li></ul>

## 1.4. Analyse QQCCP (Cadrage Exhaustif)

Question	Réponse pour le Projet
<b>QUOI ?</b>	<p>Un site web de catalogage et d'apprentissage des chants de messe avec :</p> <ul style="list-style-type: none"> <li>• Paroles complètes</li> <li>• 4 pistes audio (Soprano, Alto, Ténor, Basse)</li> <li>• Partitions (images/PDF)</li> <li>• Organisation selon la structure liturgique</li> <li>• Fonctionnalité d'upload pour administrateurs</li> </ul>
<b>QUI ?</b>	<p>Acteurs :</p> <ul style="list-style-type: none"> <li>• Porteur de projet et admin principal</li> <li>• Admins de confiance</li> <li>• 4 chanteurs bénévoles pour enregistrements</li> </ul> <p>Utilisateurs :</p> <ul style="list-style-type: none"> <li>• Choristes débutants et confirmés</li> <li>• Personnes apprenant seules</li> <li>• Chefs de chœur</li> </ul>
<b>QUAND ?</b>	<p>Deadline : Début Avril 2026 (~2 mois)</p> <p>Planification :</p> <ul style="list-style-type: none"> <li>• Sprint 1 (2 sem) : Setup + Base de données</li> <li>• Sprint 2 (2 sem) : Interface catalogue</li> <li>• Sprint 3 (2 sem) : Upload et admin</li> <li>• Sprint 4 (2 sem) : Tests et déploiement</li> </ul>
<b>COMMENT ?</b>	<p>Stack Technique :</p> <ul style="list-style-type: none"> <li>• IDE : Google Antigravity (assistance IA)</li> <li>• Frontend : React / Next.js (à confirmer)</li> <li>• Backend : Node.js + Base de données</li> <li>• Hébergement : Vercel</li> <li>• Stockage audio : Cloud (AWS S3 / Cloudinary)</li> <li>• Versioning : Git</li> </ul> <p>Méthodologie : Agile (Sprints de 2 semaines)</p>
<b>COMBIEN ?</b>	<p>Budget : 0€ (technologies gratuites)</p> <p>Ressources Humaines :</p> <ul style="list-style-type: none"> <li>• 1 développeur (vous)</li> <li>• 4 bénévoles pour enregistrements audio</li> <li>• Admins bénévoles pour gestion contenu</li> </ul> <p>Charge estimée :</p> <ul style="list-style-type: none"> <li>• 15-20 chants à enregistrer au MVP</li> </ul>

	<ul style="list-style-type: none"><li>• ~80h de développement (10h/semaine)</li></ul>
<b>POURQUOI ?</b>	<p>Raisons d'être :</p> <ul style="list-style-type: none"><li>• Combler le manque de ressources structurées</li><li>• Faciliter l'apprentissage autonome</li><li>• Préserver les chants pour les générations futures</li><li>• Démocratiser l'accès aux voix spécifiques (notamment basse)</li><li>• Aider la formation vocale des choristes</li></ul>

## 1.5. Analyse SWOT (Forces, Faiblesses, Opportunités, Menaces)

Catégorie SWOT	Analyse
<b>FORCES (S) Atouts internes</b>	<ul style="list-style-type: none"> <li>Motivation personnelle forte et besoin réel identifié</li> <li>Accès à Google Antigravity (assistance IA pour débutants)</li> <li>Technologies gratuites et modernes (Vercel, React)</li> <li>Communauté de bénévoles disponibles (chanteurs)</li> <li>Niche spécifique = peu de concurrence directe</li> </ul>
<b>FAIBLESSES (W) Limitations internes</b>	<ul style="list-style-type: none"> <li>Niveau débutant en développement web</li> <li>Deadline serrée (2 mois)</li> <li>Dépendance aux bénévoles pour enregistrements</li> <li>Absence de budget pour outils premium</li> <li>Risque de sous-estimation de la complexité technique</li> </ul>
<b>OPPORTUNITÉS (O) Facteurs externes positifs</b>	<ul style="list-style-type: none"> <li>Besoin croissant d'outils numériques pour églises</li> <li>Pas de solution exhaustive existante pour chants en français</li> <li>Potentiel d'extension à d'autres paroisses/diocèses</li> <li>Tendance à l'apprentissage en ligne</li> <li>Outils IA facilitant le développement pour débutants</li> </ul>
<b>MENACES (T) Risques externes</b>	<ul style="list-style-type: none"> <li>Complexité technique sous-estimée</li> <li>Problèmes de droits d'auteur sur certains chants</li> <li>Qualité audio variable des enregistrements bénévoles</li> <li>Risque d'abandon si trop complexe</li> <li>Difficulté à recruter tous les bénévoles nécessaires</li> </ul>

## 1.6. Analyse de Faisabilité et Contraintes

Type de Faisabilité	Évaluation
<b>Faisabilité TECHNIQUE</b>	<b>MOYENNEMENT FAISABLE</b> <ul style="list-style-type: none"> <li>Google Antigravity aide les débutants</li> <li>Technologies web standard bien documentées</li> <li>Risque : upload et gestion de fichiers audio</li> <li>Solution : utiliser services cloud existants (Cloudinary)</li> </ul>
<b>Faisabilité HUMAINE</b>	<b>FAISABLE AVEC RISQUES</b> <ul style="list-style-type: none"> <li>1 développeur débutant = charge</li> </ul>

	<p>importante</p> <ul style="list-style-type: none"> <li>• Besoin de 4 bénévoles pour enregistrements</li> <li>• Risque : disponibilité et coordination des bénévoles</li> <li>• Solution : commencer avec moins de chants si nécessaire</li> </ul>
<b>Faisabilité TEMPORELLE</b>	<p>SERRÉ MAIS POSSIBLE</p> <ul style="list-style-type: none"> <li>• 2 mois pour un MVP = réaliste si scope limité</li> <li>• MVP = 15-20 chants, fonctionnalités essentielles</li> <li>• Risque : dépassement si fonctionnalités trop ambitieuses</li> <li>• Solution : MoSCoW strict + Agile pour ajuster</li> </ul>
<b>Faisabilité FINANCIÈRE</b>	<p>TOTALEMENT FAISABLE</p> <ul style="list-style-type: none"> <li>• Budget 0€ = technologies gratuites uniquement</li> <li>• Vercel : plan gratuit suffisant pour MVP</li> <li>• Cloudinary : plan gratuit (limite 25GB)</li> <li>• Risque : dépassement de quotas gratuits</li> <li>• Solution : optimiser taille des fichiers audio</li> </ul>

**VERDICT GLOBAL :** Projet FAISABLE avec un périmètre MVP bien défini et une approche Agile permettant les ajustements. La principale contrainte est le temps, nécessitant une priorisation stricte des fonctionnalités.

## PHASE 2 : DÉFINITION DES EXIGENCES

### 2.1. Besoins Fonctionnels (Le "QUOI" du système)

Les besoins fonctionnels définissent ce que le système doit faire pour les utilisateurs.

#### BF1 - Gestion des Utilisateurs

- Inscription et connexion des utilisateurs (email/mot de passe)
- Distinction entre utilisateurs simples et administrateurs
- Gestion de profil (nom, voix préférée)

#### BF2 - Catalogue de Chants

- Affichage des chants organisés selon la structure liturgique de la messe
- Catégories : Entrée, Kyrie, Psaume, Acclamation Évangile, Prière Universelle, Offertoire, Sanctus, Agnus Dei, Communion, Sortie
- Recherche et filtrage par catégorie, titre, ou mots-clés
- Affichage du nombre de chants disponibles par catégorie

#### BF3 - Page de Détails d'un Chant

- Titre du chant
- Paroles complètes et structurées
- 4 lecteurs audio (Soprano, Alto, Ténor, Basse) avec contrôles play/pause
- Affichage de la partition (image ou PDF)
- Indication visuelle des voix disponibles/manquantes

#### BF4 - Upload et Gestion de Chants (Administrateurs)

- Formulaire d'ajout de chant avec champs : titre, catégorie, paroles, 4 fichiers audio, partition
- Validation automatique : signaler les voix manquantes
- Upload de partition uniquement si format reconnu (PDF, JPG, PNG)
- Modification et suppression de chants existants
- Interface d'administration simple et intuitive

#### BF5 - Apprentissage d'un Chant (Futur)

- Mode karaoke : paroles défilant synchronisées avec l'audio
- Possibilité de ralentir l'audio pour faciliter l'apprentissage
- Marquage des chants "apris" ou "en cours"

#### BF6 - Vocalises et Exercices (Futur)

- Catalogue d'exercices vocaux (échauffement, technique)
- Audio d'exercices de vocalise pour chaque voix

## 2.2. Besoins Non Fonctionnels (Le "COMMENT" et les qualités)

Catégorie	Exigences
<b>Sécurité et Confidentialité</b>	<ul style="list-style-type: none"> <li>Authentification sécurisée (hachage des mots de passe)</li> <li>Protection contre les injections SQL</li> <li>HTTPS obligatoire pour tout le site</li> <li>Seuls les admins peuvent ajouter/modifier du contenu</li> </ul>
<b>Performance</b>	<ul style="list-style-type: none"> <li>Temps de chargement de la page &lt; 3 secondes</li> <li>Streaming audio optimisé (pas de téléchargement complet)</li> <li>Compression des fichiers audio (MP3, 128-192 kbps)</li> </ul>
<b>Accessibilité et Compatibilité</b>	<ul style="list-style-type: none"> <li>Site responsive (mobile, tablette, desktop)</li> <li>Compatible navigateurs : Chrome, Firefox, Safari, Edge</li> <li>Interface simple et épurée (débutants)</li> </ul>
<b>Ergonomie et Simplicité</b>	<ul style="list-style-type: none"> <li>Navigation intuitive en 3 clics maximum</li> <li>Design clair avec emojis liturgiques   </li> <li>Formulaires simples avec validation en temps réel</li> </ul>
<b>Fiabilité et Disponibilité</b>	<ul style="list-style-type: none"> <li>Disponibilité 99% (tolère quelques interruptions)</li> <li>Sauvegarde automatique de la base de données</li> <li>Gestion des erreurs avec messages clairs</li> </ul>
<b>Maintenabilité et Évolutivité</b>	<ul style="list-style-type: none"> <li>Code commenté et structuré (facilite la reprise)</li> <li>Architecture modulaire (ajout facile de fonctionnalités)</li> <li>Base de données scalable (support 100+ chants futurs)</li> </ul>

## 2.3. Priorisation MoSCoW (Définition du MVP)

La méthode MoSCoW permet de définir ce qui doit absolument être dans le MVP (Must Have) et ce qui peut attendre (Should/Could/Won't Have).

### MUST HAVE - Fonctionnalités ESSENTIELLES pour le MVP

*Ces fonctionnalités DOIVENT être présentes, sinon le produit n'est pas viable.*

- MH1 - Catalogue de chants organisé selon la structure de la messe (10 catégories)
- MH2 - Page de détails d'un chant : titre + paroles + 4 audio + partition
- MH3 - Upload de chants par administrateurs (formulaire simple)
- MH4 - Authentification administrateur basique (login/password)
- MH5 - Base de données pour stocker les chants et métadonnées
- MH6 - Interface responsive (mobile-friendly)
- MH7 - Au minimum 15-20 chants au lancement

### SHOULD HAVE - Fonctionnalités IMPORTANTES (Version 1.1)

*Importantes, mais le MVP peut fonctionner sans elles.*

- SH1 - Recherche et filtrage de chants par titre/catégorie
- SH2 - Indicateur visuel des voix disponibles/manquantes
- SH3 - Modification et suppression de chants (interface admin)
- SH4 - Gestion de profil utilisateur (préférences, voix)
- SH5 - Page d'accueil avec présentation du projet

### COULD HAVE - Fonctionnalités SOUHAITABLES (Améliorations futures)

*Souhaitables si le temps le permet.*

- CH1 - Mode apprentissage : paroles défilant synchronisées
- CH2 - Marquage des chants "favoris" ou "en cours d'apprentissage"
- CH3 - Statistiques : chants les plus consultés
- CH4 - Export de la liste de chants (PDF)
- CH5 - Système de notation/commentaires sur les chants

### WON'T HAVE - Hors périmètre pour cette version

*Fonctionnalités explicitement exclues du MVP.*

- WH1 - Epic Vocalises (exercices vocaux) → Version 2.0
- WH2 - Ralentissement de l'audio → Version 2.0
- WH3 - Upload public (n'importe qui peut ajouter) → Trop risqué pour MVP
- WH4 - Application mobile native → Web mobile suffit
- WH5 - Multilingue (autres langues que français) → Futur

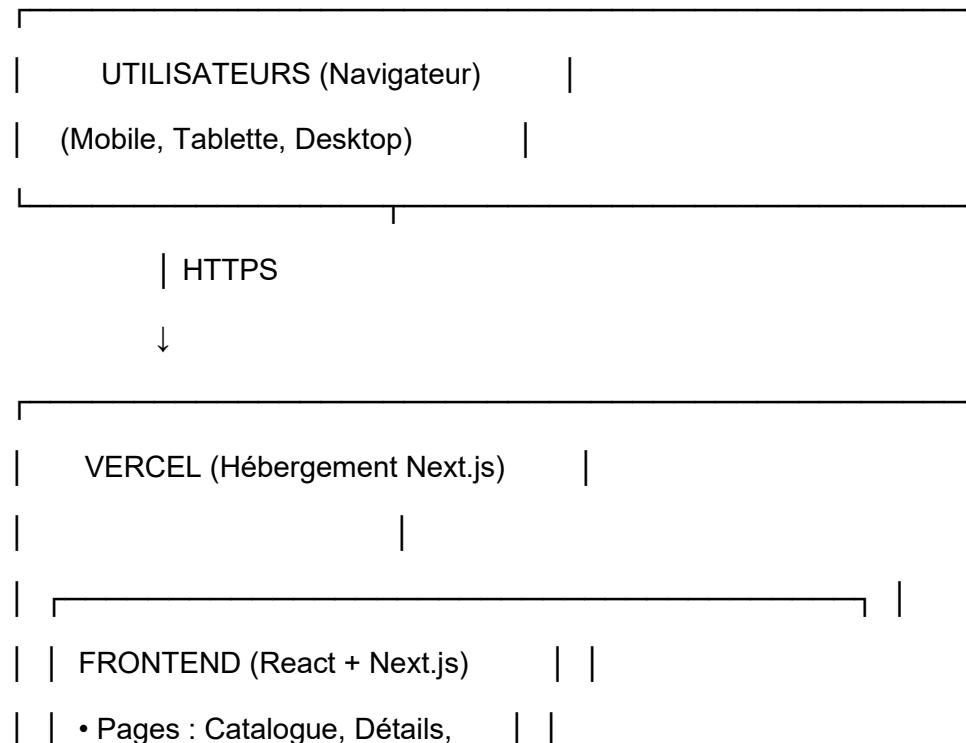
## PHASE 3 : CONCEPTION ET PLANIFICATION

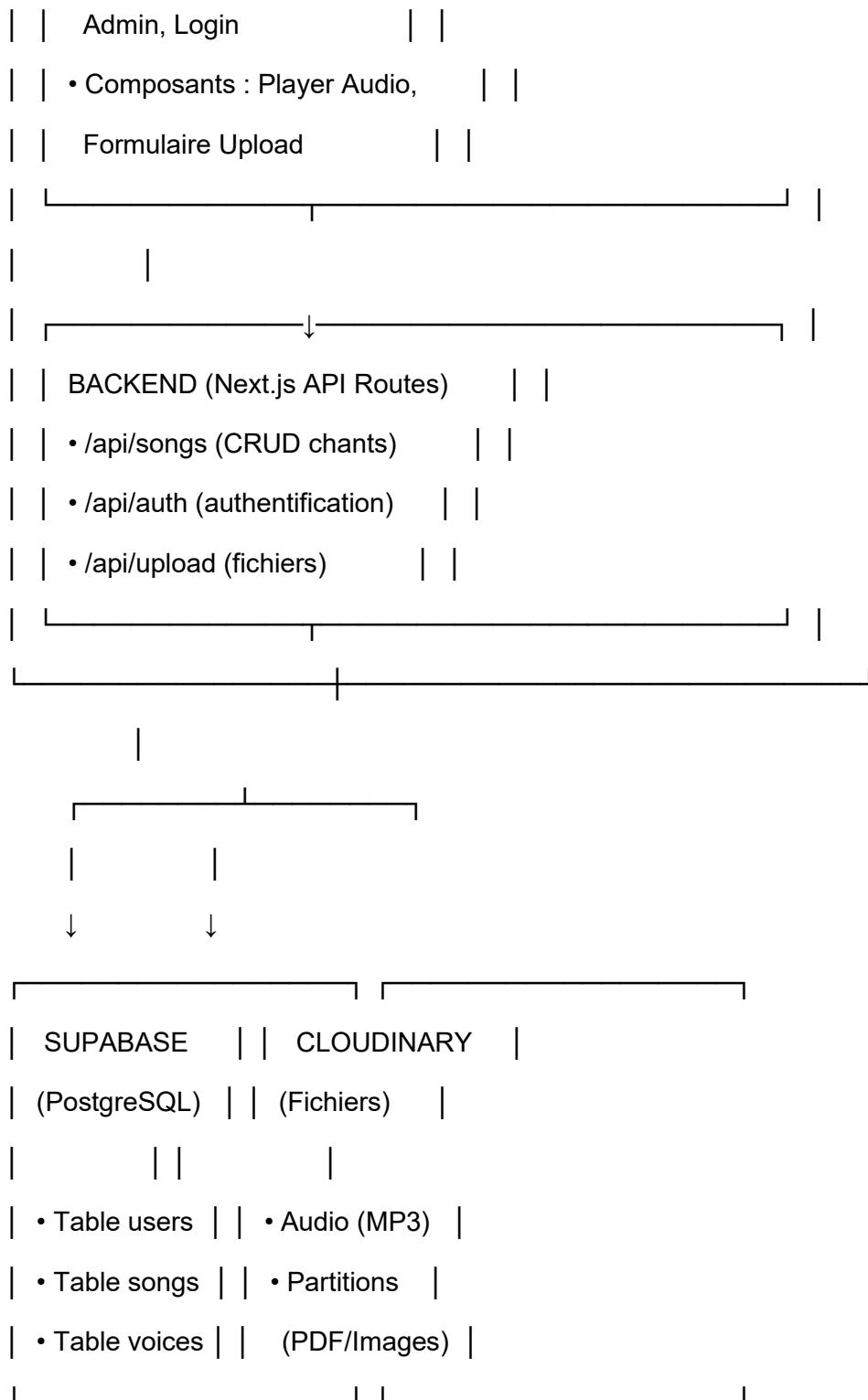
### 3.1. Architecture Technique

Stack Technologique Recommandée

Composant	Technologie Choisie
Frontend	React + Next.js (framework moderne, SSR, optimisé SEO)
Backend	Next.js API Routes (intégré, simplifie l'architecture)
Base de Données	PostgreSQL (Supabase gratuit) ou MongoDB (MongoDB Atlas gratuit)
Stockage Audio/Images	Cloudinary (25GB gratuit) ou AWS S3 (5GB gratuit)
Authentification	NextAuth.js (simple, intégré Next.js)
Hébergement	Vercel (déploiement automatique depuis Git)
Versioning	Git + GitHub (historique, collaboration)
IDE/Assistance	Google Antigravity (assistance IA Gemini 3)

Architecture Système (Schéma Conceptuel)





### **3.2. Structure de Décomposition du Travail (WBS)**

Le WBS décompose le projet en lots de travail gérables.

#### **1. INITIALISATION**

- 1.1. Setup environnement de développement (Google Antigravity)
- 1.2. Création du projet Next.js
- 1.3. Configuration Git et GitHub
- 1.4. Déploiement initial sur Vercel

#### **2. BASE DE DONNÉES**

- 2.1. Setup Supabase
- 2.2. Création du schéma (tables users, songs, voices)
- 2.3. Test de connexion backend ↔ DB

#### **3. AUTHENTIFICATION**

- 3.1. Installation NextAuth.js
- 3.2. Page de login
- 3.3. Middleware de protection admin

#### **4. FRONTEND - CATALOGUE**

- 4.1. Page d'accueil
- 4.2. Page catalogue (liste des chants par catégorie)
- 4.3. Page détails d'un chant
- 4.4. Composant lecteur audio (4 voix)

#### **5. FRONTEND - ADMINISTRATION**

- 5.1. Page d'ajout de chant (formulaire)
- 5.2. Page de modification de chant
- 5.3. Validation du formulaire

#### **6. BACKEND - API**

- 6.1. API GET /songs (liste chants)
- 6.2. API GET /songs/:id (détails)
- 6.3. API POST /songs (créer chant)
- 6.4. API PUT /songs/:id (modifier)
- 6.5. API DELETE /songs/:id (supprimer)
- 6.6. API POST /upload (fichiers audio/partition)

#### **7. STOCKAGE FICHIERS**

- 7.1. Configuration Cloudinary
- 7.2. Intégration upload audio
- 7.3. Intégration upload partition

## 8. TESTS ET QUALITÉ

- 8.1. Tests fonctionnels (navigation, ajout chant)
- 8.2. Tests audio (lecture, streaming)
- 8.3. Tests responsive (mobile, tablette)
- 8.4. Correction des bugs

## 9. CONTENU

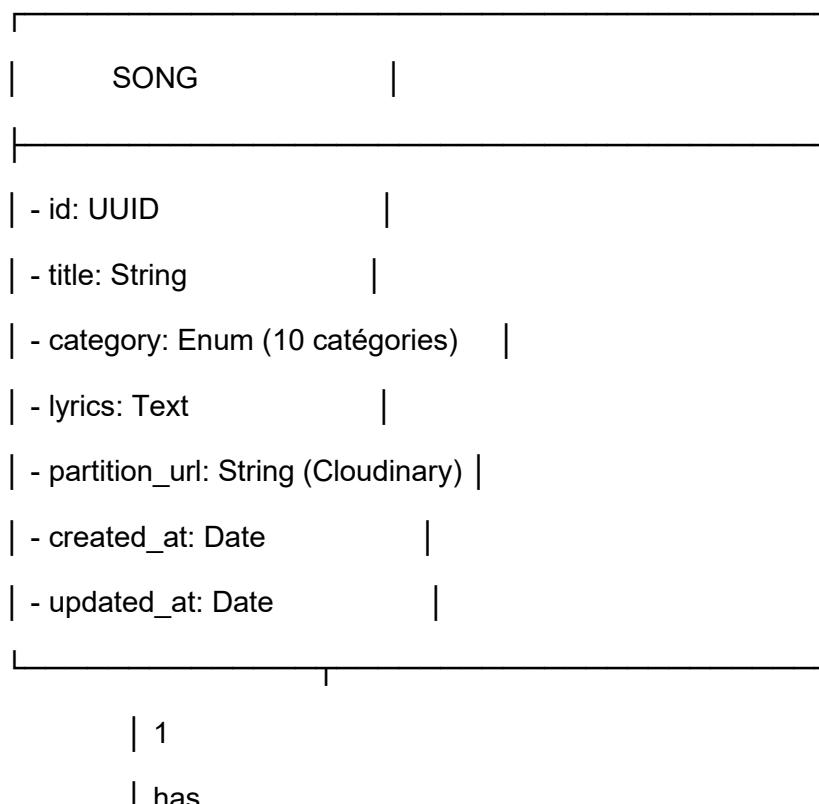
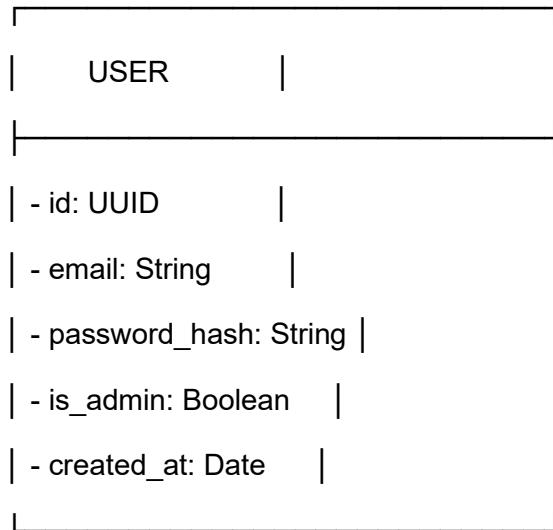
- 9.1. Collecte des paroles des 15-20 chants
- 9.2. Enregistrements audio (4 voix × 15-20 chants)
- 9.3. Scan/photo des partitions
- 9.4. Import dans la base de données

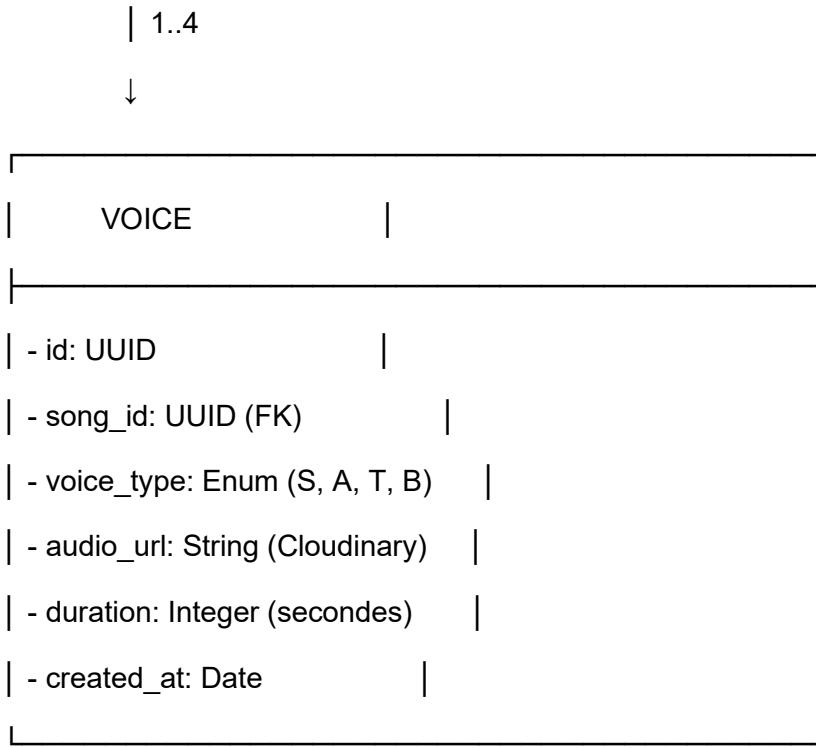
## 10. DÉPLOIEMENT ET LIVRAISON

- 10.1. Tests finaux en production
- 10.2. Documentation utilisateur
- 10.3. Formation des administrateurs
- 10.4. Lancement officiel

### 3.3. Modélisation UML - Diagramme de Classes

Modèle Conceptuel de Données (simplifié) :





### Diagramme de Cas d'Utilisation

(Version textuelle simplifiée)

ACTEUR : Utilisateur (Choriste)

- → Consulter le catalogue de chants
- → Voir les détails d'un chant
- → Écouter les 4 voix
- → Télécharger la partition

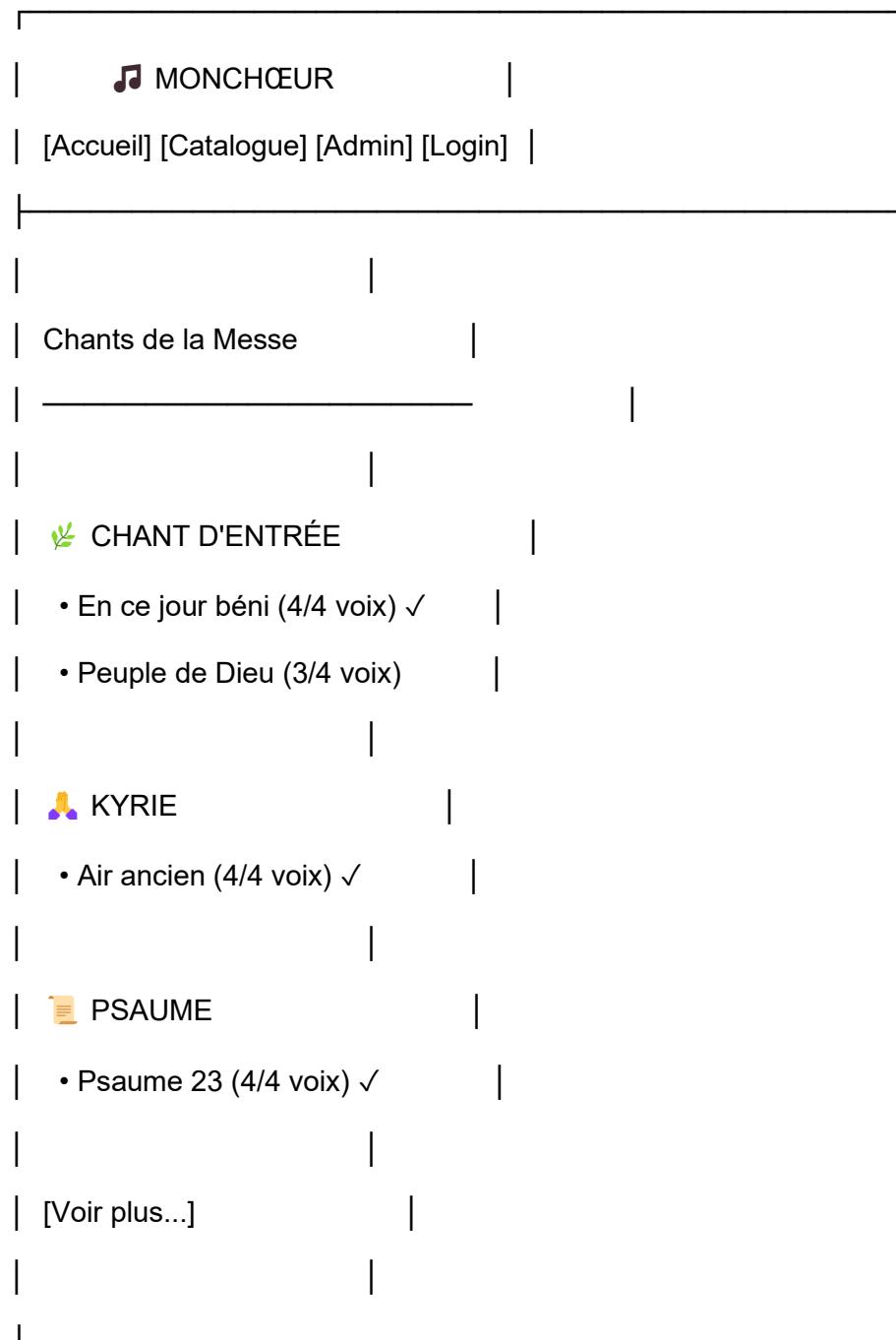
ACTEUR : Administrateur

- → Se connecter (authentification)
- → Ajouter un nouveau chant
- → Modifier un chant existant
- → Supprimer un chant
- → Uploader fichiers audio et partition

### 3.4. Maquettage et Prototypage

#### Wireframes Textuels des Pages Principales

##### 1. PAGE CATALOGUE (Liste des chants)



## 2. PAGE DÉTAILS D'UN CHANT

<a href="#">← Retour au catalogue</a>	
EN CE JOUR BÉNI	
Catégorie : Chant d'Entrée	
— PAROLES —	
En ce jour béni, le Seigneur...	
[Texte complet des paroles]	
— ÉCOUTER —	
Soprano [▶] 2:34	
Alto [▶] 2:34	
Ténor [▶] 2:34	
Basse [▶] 2:34	
— PARTITION —	
[ Affichage PDF/Image]	
[ Télécharger]	

## 3. PAGE ADMIN - AJOUT DE CHANT

Administration - Ajouter un Chant	
<hr/>	
Titre : [ ]	
<hr/>	
Catégorie : [▼ Chant d'Entrée]	
<hr/>	
Paroles :	
[ ]	
[ ]	
<hr/>	
— Fichiers Audio —	
Soprano : [Choisir fichier]	✓
Alto : [Choisir fichier]	✓
Ténor : [Choisir fichier]	⚠ Manquant
Basse : [Choisir fichier]	✓
<hr/>	
Partition : [Choisir PDF/Image]	✓
<hr/>	
[Annuler]	[Enregistrer le chant]
<hr/>	

#### Outils de Prototypage Recommandés

Outil	Usage
<b>Figma</b>	Gratuit, collaboratif, prototypes cliquables → Recommandé pour maquettes haute fidélité
<b>Balsamiq</b>	Wireframes rapides, style "sketch" → Bon pour maquettes basse fidélité

**Papier + Crayon**

Rapide pour esquisses initiales  
→ Utilisé pour premières itérations

### 3.5. Planning Agile - Sprints de 2 Semaines

Approche Agile avec 4 Sprints de 2 semaines pour respecter la deadline début avril.

Sprint	Focus	Tâches
<b>SPRINT 1 (3-16 fév)</b>	Setup + Base de Données	<ul style="list-style-type: none"> <li>• Setup Google Antigravity + Next.js</li> <li>• Configuration Vercel</li> <li>• Setup Supabase (DB)</li> <li>• Création du schéma de données</li> <li>• Tests connexion backend ↔ DB</li> </ul> <p>Livrable : Infrastructure fonctionnelle</p>
<b>SPRINT 2 (17 fév - 2 mars)</b>	Catalogue + Frontend	<ul style="list-style-type: none"> <li>• Page d'accueil</li> <li>• Page catalogue (liste chants)</li> <li>• Page détails chant</li> <li>• Composant lecteur audio</li> <li>• Design responsive</li> </ul> <p>Livrable : Interface utilisateur navigable</p>
<b>SPRINT 3 (3-16 mars)</b>	Admin + Upload	<ul style="list-style-type: none"> <li>• Authentification (NextAuth)</li> <li>• Page admin (ajout chant)</li> <li>• Formulaire d'upload</li> <li>• Intégration Cloudinary</li> <li>• API CRUD complète</li> </ul> <p>Livrable : Système d'administration fonctionnel</p>
<b>SPRINT 4 (17-30 mars)</b>	Contenu + Tests + Déploiement	<ul style="list-style-type: none"> <li>• Import des 15-20 chants</li> <li>• Enregistrements audio (bénévoles)</li> <li>• Tests fonctionnels complets</li> <li>• Corrections bugs</li> <li>• Tests responsive</li> <li>• Documentation</li> </ul> <p>Livrable : MVP complet prêt pour lancement</p>

⚠ **BUFFER** : 1-5 avril = tampon pour finitions et imprévus avant la deadline.



## PHASE 4 : RÉALISATION ET PILOTAGE

### 4.1. Méthodologie Agile - Pratiques Recommandées

Pratique	Description
<b>Tableau Kanban</b>	Utiliser Trello ou GitHub Projects  Colonnes : À Faire   En Cours   En Test   Terminé  Un ticket = une tâche (ex: "Créer page catalogue")
<b>Daily Scrum (si équipe)</b>	Réunion courte (15 min) chaque jour  3 questions : <ul style="list-style-type: none"><li>• Qu'ai-je fait hier ?</li><li>• Que vais-je faire aujourd'hui ?</li><li>• Quels obstacles ?</li></ul>
<b>Sprint Review (fin de sprint)</b>	Démonstration des fonctionnalités terminées  Validation avec utilisateurs test  Ajustement du backlog si nécessaire
<b>Vélocité</b>	Mesurer le nombre de tâches terminées par sprint  Aide à estimer la capacité réelle  Ajuster les prévisions si sous-estimation

### 4.2. Backlog et User Stories

Les User Stories décrivent les fonctionnalités du point de vue de l'utilisateur.

**Format : En tant que [RÔLE], je veux [ACTION] afin de [BÉNÉFICE]**

#### Epic 1 : Catalogue de Chants

- US1.1 : En tant que choriste, je veux voir la liste des chants organisés par catégorie liturgique afin de trouver rapidement le chant qui m'intéresse.
- US1.2 : En tant que choriste, je veux voir les détails d'un chant (titre, paroles, audio, partition) afin d'apprendre le chant.
- US1.3 : En tant que choriste basse, je veux écouter spécifiquement la voix de basse afin de m'entraîner sur ma partie.
- US1.4 : En tant que chef de chœur, je veux voir combien de chants sont disponibles par catégorie afin de planifier la messe.

### **Epic 2 : Administration et Gestion**

- US2.1 : En tant qu'administrateur, je veux me connecter de manière sécurisée afin d'accéder à l'interface d'administration.
- US2.2 : En tant qu'administrateur, je veux ajouter un nouveau chant avec ses fichiers audio afin d'enrichir le catalogue.
- US2.3 : En tant qu'administrateur, je veux être alerté si des voix manquent afin de compléter le chant avant publication.
- US2.4 : En tant qu'administrateur, je veux modifier ou supprimer un chant afin de corriger des erreurs ou mettre à jour le contenu.

### **Epic 3 : Apprentissage (Futur - Hors MVP)**

- US3.1 : En tant que débutant, je veux voir les paroles défiler pendant l'audio afin de mieux suivre le chant.
- US3.2 : En tant que choriste, je veux ralentir l'audio afin de mieux apprendre les passages difficiles.
- US3.3 : En tant que choriste, je veux marquer un chant comme "apris" afin de suivre ma progression.

### 4.3. Gestion des Versions avec Git

#### Workflow Git Recommandé

1. 1. Créer un dépôt GitHub pour le projet
2. 2. Branches principales :
  - main : code en production (stable)
  - develop : intégration des fonctionnalités
  - feature/\* : branches de fonctionnalités
3. 3. Pour chaque nouvelle fonctionnalité :
  - Créer une branche : git checkout -b feature/nom-feature
  - Développer et commiter régulièrement
  - Merger dans develop quand terminé
4. 4. Commits clairs et fréquents (ex: "feat: ajout page catalogue")
5. 5. Déploiement automatique Vercel depuis main

#### Bonnes Pratiques de Commit

Préfixe	Usage
feat:	Nouvelle fonctionnalité
fix:	Correction de bug
docs:	Documentation
style:	Formatage, pas de changement de code
refactor:	Refactorisation du code
test:	Ajout de tests

Exemples :

- feat: création de la page catalogue avec liste des chants
- fix: correction du lecteur audio qui ne démarrait pas sur mobile
- docs: ajout du README avec instructions d'installation

## 5. LIVRABLES ET DOCUMENTATION

### 5.1. Livrables Attendus du Projet

Livrable	Description
<b>Produit Fonctionnel</b>	Site web MonChœur déployé sur Vercel <ul style="list-style-type: none"><li>MVP avec minimum 15-20 chants</li><li>Interface responsive</li><li>Administration fonctionnelle</li></ul>
<b>Base de Données</b>	Base Supabase structurée et remplie <ul style="list-style-type: none"><li>Tables : users, songs, voices</li><li>15-20 chants avec métadonnées complètes</li></ul>
<b>Fichiers Multimédias</b>	Stockage Cloudinary <ul style="list-style-type: none"><li>60-80 fichiers audio (4 voix × 15-20 chants)</li><li>15-20 partitions (PDF ou images)</li><li>Audio optimisé (MP3, 128-192 kbps)</li></ul>
<b>Code Source</b>	Dépôt GitHub complet <ul style="list-style-type: none"><li>Code commenté et structuré</li><li>README avec instructions</li><li>Historique Git propre</li></ul>
<b>Documentation Technique</b>	<ul style="list-style-type: none"><li>Architecture du système</li><li>Schéma de base de données</li><li>Guide d'installation en local</li><li>Documentation API</li></ul>
<b>Documentation Utilisateur</b>	<ul style="list-style-type: none"><li>Guide d'utilisation pour choristes</li><li>Guide d'administration (ajout chants)</li><li>FAQ</li></ul>
<b>Rapport de Projet</b>	Document récapitulatif : <ul style="list-style-type: none"><li>Contexte et objectifs</li><li>Méthodologie suivie</li><li>Difficultés rencontrées</li><li>Résultats obtenus</li><li>Perspectives d'évolution</li></ul>

### 5.2. Plan de Documentation

Document	Contenu
<b>README.md (GitHub)</b>	<ul style="list-style-type: none"><li>Description du projet</li><li>Technologies utilisées</li><li>Instructions d'installation</li><li>Commandes de développement</li><li>Contribution</li></ul>
<b>ARCHITECTURE.md</b>	<ul style="list-style-type: none"><li>Diagramme d'architecture</li><li>Stack technique détaillée</li><li>Structure des dossiers</li><li>Flux de données</li></ul>
<b>API.md</b>	<ul style="list-style-type: none"><li>Documentation des endpoints</li><li>Exemples de requêtes/réponses</li><li>Codes d'erreur</li></ul>

	<ul style="list-style-type: none"><li>• Authentification</li></ul>
<b>GUIDE_UTILISATEUR.pdf</b>	<ul style="list-style-type: none"><li>• Navigation sur le site</li><li>• Recherche de chants</li><li>• Écoute des voix</li><li>• Téléchargement partitions</li></ul>
<b>GUIDE_ADMIN.pdf</b>	<ul style="list-style-type: none"><li>• Connexion admin</li><li>• Ajout d'un chant (étapes)</li><li>• Modification/suppression</li><li>• Bonnes pratiques upload</li></ul>

## 6. ANNEXES

### 6.1. Structure Liturgique des Chants de la Messe

Catégorie	Description
Chant d'Entrée	Ouverture de la célébration
Kyrie	Demande de pardon
Psaume	Méditation après la 1ère lecture
Acclamation de l'Évangile	Accueil de la Parole (Alléluia)
Prière Universelle	Prière des fidèles
Chant d'Offertoire	Préparation des dons
Sanctus	"Saint, Saint, Saint"
Agnus Dei	"Agneau de Dieu"
Communion	Pendant la communion
Chant de Sortie	Envoi en mission

### 6.2. Exemples de Chants Populaires par Catégorie

(Liste non exhaustive - à compléter lors de la collecte)

Catégorie	Exemples
Entrée	<ul style="list-style-type: none"> <li>• En ce jour béni</li> <li>• Peuple de Dieu</li> <li>• Louez le Seigneur</li> </ul>
Kyrie	<ul style="list-style-type: none"> <li>• Air ancien</li> <li>• Kyrie Eleison</li> </ul>
Psaume	<ul style="list-style-type: none"> <li>• Psaume 23</li> <li>• Le Seigneur est mon berger</li> </ul>
Alléluia	<ul style="list-style-type: none"> <li>• Magnificat</li> <li>• Alléluia, gloire à Dieu</li> </ul>
Prière Universelle	<ul style="list-style-type: none"> <li>• Ô Seigneur, tends l'oreille</li> </ul>
Offertoire	<ul style="list-style-type: none"> <li>• Celui qui sème en pleurant</li> </ul>
Sanctus	<ul style="list-style-type: none"> <li>• Air ancien</li> <li>• Saint, Saint, Saint</li> </ul>
Agnus Dei	<ul style="list-style-type: none"> <li>• Air ancien</li> <li>• Agneau de Dieu</li> </ul>
Communion	<ul style="list-style-type: none"> <li>• Demeurez en mon amour</li> <li>• Pain de vie</li> </ul>
Sortie	<ul style="list-style-type: none"> <li>• Aime ton prochain</li> <li>• Allez dans le monde</li> </ul>

### 6.3. Ressources et Outils Recommandés

Catégorie	Ressources
Développement	<ul style="list-style-type: none"> <li>• Google Antigravity (IDE IA)</li> <li>• Next.js Documentation</li> <li>• MDN Web Docs</li> </ul>

<b>Design</b>	<ul style="list-style-type: none"> <li>• Stack Overflow</li> <li>• Figma (prototypage)</li> <li>• Coolors.co (palettes de couleurs)</li> <li>• Google Fonts</li> </ul>
<b>Gestion de Projet</b>	<ul style="list-style-type: none"> <li>• GitHub Projects (Kanban)</li> <li>• Trello</li> <li>• Notion</li> </ul>
<b>Apprentissage</b>	<ul style="list-style-type: none"> <li>• freeCodeCamp</li> <li>• Next.js Learn</li> <li>• React Documentation</li> <li>• YouTube tutorials</li> </ul>
<b>Audio</b>	<ul style="list-style-type: none"> <li>• Audacity (édition audio gratuite)</li> <li>• Online Convert (conversion formats)</li> <li>• TwistedWave (en ligne)</li> </ul>
<b>Communauté</b>	<ul style="list-style-type: none"> <li>• Discord Next.js</li> <li>• Reddit r/webdev</li> <li>• Stack Overflow</li> </ul>

#### **6.4. Checklist de Lancement MVP**

- 15-20 chants complets uploadés (titre, paroles, 4 voix, partition)
- Toutes les catégories liturgiques représentées
- Site responsive testé sur mobile, tablette, desktop
- Authentification admin fonctionnelle et sécurisée
- Lecteurs audio fonctionnels sur tous navigateurs
- Formulaire d'ajout de chant testé et validé
- Performance : temps de chargement < 3 secondes
- Déploiement Vercel réussi et stable
- Documentation utilisateur complète
- Documentation admin créée
- Tests utilisateur réalisés (3-5 personnes)
- Corrections des bugs majeurs effectuées
- README GitHub à jour
- Sauvegarde de la base de données configurée
- Formation des administrateurs effectuée



---

## **FIN DU DOCUMENT**

---

*Bonne chance pour la réalisation de MonChœur ! ♪*