

# Cahier des Charges

## Système de Gestion des Inscriptions Scolaires

### École Prestige - Module d'Inscription

**Version:** 1.0

**Date:** 30 décembre 2025

**Statut:** Document de référence technique

---

## 1. PRÉSENTATION GÉNÉRALE DU PROJET

### 1.1 Contexte

Ce document constitue le cahier des charges technique du module de gestion des inscriptions du projet "Système Intégré de Gestion Scolaire" pour l'École Prestige. Ce module représente la première phase du projet global de digitalisation administrative.

### 1.2 Objectifs du Module

- Digitaliser complètement le processus d'inscription des élèves
- Permettre aux parents de soumettre des demandes d'inscription en ligne
- Offrir à l'administration une application desktop robuste fonctionnant en mode connecté et déconnecté
- Assurer une synchronisation fiable des données entre le système local et le cloud
- Réduire de 70% le temps de traitement des inscriptions

### 1.3 Architecture Globale du Système

Le système comprend **trois composantes principales** :

#### A. Application Web Parents (Frontend Web)

- Interface publique pour les parents
- Formulaire d'inscription en ligne
- Suivi du statut de la demande
- Technologies : React.js / Next.js, TailwindCSS

#### B. Application Desktop Administration (Application Locale)

- Application lourde pour le personnel administratif
- Fonctionnement en mode online et offline
- Gestion complète des demandes d'inscription
- Technologies : Electron + React, SQLite local

## C. Backend Centralisé

- Supabase comme backend cloud
  - Gestion de l'authentification
  - Stockage des données et fichiers
  - API REST pour la synchronisation
- 

## 2. SPÉCIFICATIONS FONCTIONNELLES

### 2.1 Application Web Parents

#### 2.1.1 Accès et Authentification

##### F-WEB-01 : Page d'accueil publique

- Description : Présentation du processus d'inscription
- Bouton "Nouvelle inscription" visible sans authentification
- Informations sur les documents requis
- FAQ sur le processus d'inscription

##### F-WEB-02 : Création de compte parent

- Formulaire d'inscription avec : nom, prénom, email, téléphone, mot de passe
- Validation email obligatoire (lien de confirmation)
- Gestion des mots de passe sécurisée (minimum 8 caractères)
- Possibilité de connexion via email/mot de passe

##### F-WEB-03 : Connexion sécurisée

- Authentification par email et mot de passe

- Option "Se souvenir de moi"
- Fonctionnalité "Mot de passe oublié" avec réinitialisation par email
- Déconnexion sécurisée

## 2.1.2 Formulaire d'Inscription

### F-WEB-04 : Formulaire en plusieurs étapes

#### Étape 1 : Informations sur l'enfant

- Nom et prénom(s)
- Date de naissance (avec calcul automatique de l'âge)
- Lieu de naissance
- Sexe
- Classe souhaitée (sélection dans une liste dynamique)
- Photo de l'enfant (upload, formats JPG/PNG, max 5 MB)

#### Étape 2 : Informations des parents

- Parent 1 (père) : nom, prénom, profession, téléphone, email
- Parent 2 (mère) : nom, prénom, profession, téléphone, email
- Tuteur légal (si différent) : nom, prénom, lien de parenté, téléphone
- Adresse complète du domicile
- Personne à contacter en cas d'urgence

#### Étape 3 : Informations médicales

- Groupe sanguin
- Allergies connues (champ texte)
- Maladies chroniques (champ texte)
- Traitements en cours (champ texte)
- Nom et contact du médecin traitant

#### Étape 4 : Documents justificatifs

- Acte de naissance (PDF, max 2 MB)

- Certificat de vaccination (PDF, max 2 MB)
- Derniers bulletins scolaires si applicable (PDF, max 5 MB)
- Justificatif de domicile (PDF, max 2 MB)
- Barre de progression d'upload pour chaque fichier

### **Étape 5 : Récapitulatif et validation**

- Affichage récapitulatif de toutes les informations saisies
- Possibilité de revenir en arrière pour modifier
- Case à cocher : "J'atteste que les informations fournies sont exactes"
- Bouton "Soumettre la demande"

### **F-WEB-05 : Validation et sauvegarde automatique**

- Validation en temps réel des champs obligatoires
- Messages d'erreur contextuels et explicites
- Sauvegarde automatique en brouillon toutes les 30 secondes
- Possibilité de quitter et reprendre plus tard

### **F-WEB-06 : Confirmation de soumission**

- Message de confirmation avec numéro de demande unique
- Email automatique de confirmation au parent
- Instructions sur les prochaines étapes
- Délai estimé de traitement

#### **2.1.3 Suivi de la Demande**

### **F-WEB-07 : Tableau de bord parent**

- Liste de toutes les demandes d'inscription du parent
- Statut de chaque demande (En attente, En cours d'examen, Acceptée, Refusée, Documents manquants)
- Date de soumission et dernière mise à jour
- Numéro de référence de la demande

### **F-WEB-08 : Détail d'une demande**

- Affichage complet des informations soumises
- Historique des changements de statut avec dates
- Messages/commentaires de l'administration
- Documents téléchargés consultables

#### **F-WEB-09 : Notifications**

- Notification email lors du changement de statut
- Badge de notification dans l'interface web
- Liste des notifications consultables

#### **F-WEB-10 : Ajout de documents complémentaires**

- Possibilité d'ajouter des documents après soumission si demandés par l'administration
  - Upload avec confirmation
- 

## **2.2 Application Desktop Administration**

### **2.2.1 Installation et Configuration Initiale**

#### **F-DESK-01 : Installation de l'application**

- Programme d'installation Windows (.exe) et MacOS (.dmg)
- Configuration initiale au premier lancement
- Paramétrage de la connexion Supabase (URL du projet, clés API)
- Test de connexion à la base de données cloud
- Création automatique de la base SQLite locale

#### **F-DESK-02 : Gestion des modes de fonctionnement**

- Détection automatique de la connexion internet
- Indicateur visuel du mode actuel (Online/Offline) dans l'interface
- Basculement automatique entre les modes
- Message d'information lors du passage en mode offline

## **2.2.2 Authentification et Gestion des Utilisateurs**

### **F-DESK-03 : Connexion du personnel administratif**

- Authentification via Supabase (email/mot de passe)
- Gestion des rôles : Administrateur, Gestionnaire, Consultation seule
- Session persistante avec déconnexion automatique après 30 minutes d'inactivité
- Possibilité de travailler hors ligne après une première connexion

### **F-DESK-04 : Gestion des comptes utilisateurs (Admin uniquement)**

- Création de nouveaux comptes pour le personnel
- Attribution des rôles et permissions
- Désactivation/réactivation de comptes
- Réinitialisation de mot de passe

## **2.2.3 Tableau de Bord et Vue d'Ensemble**

### **F-DESK-05 : Tableau de bord principal**

- Statistiques en temps réel :
  - Nombre total de demandes
  - Demandes en attente de traitement
  - Demandes traitées aujourd'hui
  - Demandes acceptées/refusées
- Graphiques d'évolution des inscriptions
- Liste des tâches prioritaires
- Alertes sur les demandes nécessitant une action urgente

## **2.2.4 Gestion des Demandes d'Inscription**

### **F-DESK-06 : Liste des demandes**

- Tableau avec toutes les demandes
- Colonnes : Numéro, Nom de l'enfant, Date de soumission, Classe souhaitée, Statut, Actions
- Filtres : par statut, par classe, par période, par recherche (nom, numéro)

- Tri par colonne (cliquable)
- Pagination (50 demandes par page)
- Export Excel/CSV de la liste filtrée

#### **F-DESK-07 : Consultation d'une demande**

- Affichage complet de toutes les informations de l'élève
- Visualisation des documents joints (lecteur PDF intégré)
- Téléchargement de documents individuels ou tous en ZIP
- Historique complet des actions effectuées sur la demande
- Commentaires internes entre membres du personnel

#### **F-DESK-08 : Traitement d'une demande**

- Changement de statut avec menu déroulant :
  - "En attente" → "En cours d'examen"
  - "En cours d'examen" → "Documents manquants" / "Acceptée" / "Refusée"
- Champ obligatoire pour justifier le changement de statut
- Possibilité d'envoyer un message au parent
- Demande de documents complémentaires avec liste des documents manquants

#### **F-DESK-09 : Validation et acceptation**

- Génération automatique d'un numéro de matricule élève
- Attribution automatique ou manuelle à une classe
- Impression de la lettre d'acceptation (template personnalisable)
- Envoi automatique d'email de confirmation au parent
- Création automatique du dossier élève complet dans le module élèves (phase ultérieure)

#### **F-DESK-10 : Refus d'une demande**

- Sélection du motif de refus (liste prédéfinie + champ libre)
- Message personnalisé au parent
- Envoi automatique d'email de notification
- Archivage de la demande

## **2.2.5 Gestion des Classes et Capacités**

### **F-DESK-11 : Configuration des classes**

- Liste des classes disponibles (Maternelle Petite Section, Moyenne Section, Grande Section, CP, CE1, CE2, CM1, CM2)
- Pour chaque classe : nom, capacité maximale, effectif actuel, places disponibles
- Activation/désactivation des inscriptions par classe
- Gestion de liste d'attente par classe

### **F-DESK-12 : Gestion de la liste d'attente**

- Activation automatique de la liste d'attente quand une classe est pleine
- Visualisation de la liste d'attente par classe
- Ordre chronologique de la liste
- Notification automatique au prochain parent en liste d'attente lors d'un désistement

## **2.2.6 Communication**

### **F-DESK-13 : Messagerie avec les parents**

- Envoi de messages individuels à un parent depuis le dossier
- Modèles de messages prédéfinis (demande de documents, convocation, etc.)
- Historique des communications
- Confirmation de lecture (si possible)

### **F-DESK-14 : Notifications groupées**

- Envoi d'emails groupés par statut, par classe ou sélection manuelle
- Templates personnalisables
- Prévisualisation avant envoi
- Journal des envois

## **2.2.7 Rapports et Statistiques**

### **F-DESK-15 : Génération de rapports**

- Rapport quotidien/hebdomadaire/mensuel des inscriptions
- Rapport par classe (nombre d'inscrits, places restantes)
- Statistiques démographiques (âge, provenance géographique)
- Temps moyen de traitement des demandes
- Export en PDF ou Excel

## **2.2.8 Fonctionnement Offline**

### **F-DESK-16 : Mode hors ligne**

- Accès complet en lecture à toutes les données synchronisées localement
- Possibilité de :
  - Consulter les demandes
  - Modifier les statuts
  - Ajouter des commentaires
  - Traiter les demandes
- Indicateur visuel sur chaque action effectuée en mode offline ("En attente de synchronisation")
- File d'attente des modifications en local

### **F-DESK-17 : Synchronisation automatique**

- Déclenchement automatique dès la détection d'une connexion internet
- Indicateur de progression de la synchronisation
- Gestion intelligente des conflits :
  - Priorité aux modifications serveur pour les données critiques
  - Notification à l'utilisateur en cas de conflit majeur
  - Proposition de résolution manuelle si nécessaire
- Journal de synchronisation consultable

### **F-DESK-18 : Synchronisation manuelle**

- Bouton "Synchroniser maintenant" accessible à tout moment
- Force la synchronisation bidirectionnelle
- Affichage du dernier timestamp de synchronisation réussie

---

## 2.3 Backend Supabase

### 2.3.1 Architecture de Base de Données

#### F-BACK-01 : Schéma de tables principales

##### Table : parents

- id (UUID, PK)
- user\_id (FK vers auth.users de Supabase)
- nom
- prenom
- telephone
- email
- created\_at, updated\_at

##### Table : demandes\_inscription

- id (UUID, PK)
- numero\_demande (VARCHAR unique, auto-généré)
- parent\_id (FK vers parents)
- statut (ENUM : 'en\_attente', 'en\_cours', 'documents\_manquants', 'acceptee', 'refusee')
- classe\_souhaitee
- date\_soumission
- date\_derniere\_modification
- traite\_par (FK vers users administratifs)
- created\_at, updated\_at

##### Table : eleves (dans demandes)

- id (UUID, PK)
- demande\_id (FK vers demandes\_inscription)
- nom
- prenom

- date\_naissance
- lieu\_naissance
- sexe
- photo\_url (lien vers Supabase Storage)
- created\_at, updated\_at

#### **Table : informations\_parents**

- id (UUID, PK)
- demande\_id (FK)
- pere\_nom, pere\_prenom, pere\_profession, pere\_telephone, pere\_email
- mere\_nom, mere\_prenom, mere\_profession, mere\_telephone, mere\_email
- tuteur\_nom, tuteur\_prenom, tuteur\_lien, tuteur\_telephone
- adresse\_complete
- urgence\_nom, urgence\_telephone, urgence\_lien

#### **Table : informations\_medicale**

- id (UUID, PK)
- demande\_id (FK)
- groupe\_sanguin
- allergies
- maladies\_chroniques
- traitements\_en\_cours
- medecin\_nom, medecin\_telephone

#### **Table : documents**

- id (UUID, PK)
- demande\_id (FK)
- type\_document (ENUM : 'acte\_naissance', 'certificat\_vaccination', 'bulletins', 'justificatif\_domicile', 'autre')
- nom\_fichier

- url\_storage (lien vers Supabase Storage)
- taille\_fichier
- date\_upload
- uploaded\_by (parent ou admin)

#### **Table : historique\_statuts**

- id (UUID, PK)
- demande\_id (FK)
- ancien\_statut
- nouveau\_statut
- commentaire
- modifie\_par (FK vers users)
- date\_modification

#### **Table : classes**

- id (UUID, PK)
- nom
- niveau
- capacite\_max
- effectif\_actuel
- inscriptions\_ouvertes (BOOLEAN)
- created\_at, updated\_at

#### **Table : messages**

- id (UUID, PK)
- demande\_id (FK)
- expediteur\_type (ENUM : 'parent', 'admin')
- expediteur\_id (FK)
- message
- lu (BOOLEAN)

- date\_envoi

#### **Table : sync\_log (pour tracking de synchronisation)**

- id (UUID, PK)
- device\_id (identifiant unique de l'app desktop)
- dernier\_sync
- statut\_sync
- erreurs (JSON)

#### **2.3.2 Supabase Storage**

##### **F-BACK-02 : Organisation du stockage**

- Bucket "inscriptions-documents" (privé)
- Structure : `/demandes/{demande_id}/{type_document}/{filename}`
- Bucket "photos-eleves" (privé)
- Structure : `/eleves/{demande_id}/photo.jpg`
- Politiques RLS (Row Level Security) pour sécuriser l'accès

#### **2.3.3 API et Authentification**

##### **F-BACK-03 : Authentification**

- Supabase Auth pour parents et personnel administratif
- JWT tokens pour sécuriser les requêtes API
- Gestion des rôles via métadonnées utilisateur
- Politiques RLS basées sur les rôles

##### **F-BACK-04 : API REST Endpoints** Les endpoints Supabase sont accessibles via la PostgREST API :

- GET `/demandes_inscription` - Liste des demandes (avec filtres)
- GET `/demandes_inscription?id=eq.{id}` - Détail d'une demande
- POST `/demandes_inscription` - Création nouvelle demande
- PATCH `/demandes_inscription?id=eq.{id}` - Mise à jour demande
- GET `/documents?demande_id=eq.{id}` - Documents d'une demande

- POST `/documents` - Upload référence document
- POST `/historique_statuts` - Ajouter entrée historique
- GET `/classes` - Liste des classes
- Etc.

#### **F-BACK-05 : Webhooks et notifications**

- Trigger base de données pour créer entrées d'historique automatiquement
- Edge Functions Supabase pour envoi emails (via service comme Resend ou SendGrid)
- Notifications temps réel via Supabase Realtime (optionnel)

#### **2.3.4 Sécurité**

#### **F-BACK-06 : Politiques de sécurité**

- Row Level Security (RLS) activé sur toutes les tables sensibles
  - Parents ne voient que leurs propres demandes
  - Personnel administratif voit toutes les demandes selon leurs permissions
  - Chiffrement HTTPS obligatoire
  - Validation côté serveur de tous les inputs
  - Rate limiting sur les endpoints publics
- 

### **3. SPÉCIFICATIONS TECHNIQUES**

#### **3.1 Technologies et Stack Technique**

##### **Application Web Parents**

- **Frontend** : React 18+ avec Next.js 14+ (App Router)
- **Styling** : TailwindCSS 3+
- **Gestion d'état** : Zustand ou Context API
- **Formulaires** : React Hook Form + Zod pour validation
- **Upload fichiers** : React Dropzone
- **Client API** : Supabase JavaScript Client

- **Authentification** : Supabase Auth

## Application Desktop Administration

- **Framework** : Electron 28+
- **Interface** : React 18+
- **Styling** : TailwindCSS 3+
- **Base locale** : SQLite3 (via better-sqlite3)
- **ORM** : Drizzle ORM ou SQL brut
- **Synchronisation** : Supabase JavaScript Client
- **Gestion offline** : Custom sync engine
- **Packaging** : Electron Builder

## Backend

- **BaaS** : Supabase (PostgreSQL + Storage + Auth + Edge Functions)
- **Base de données** : PostgreSQL 15+
- **Storage** : Supabase Storage
- **Auth** : Supabase Auth
- **Email** : Resend ou SendGrid (via Edge Functions)

## 3.2 Architecture de Synchronisation

### 3.2.1 Stratégie de Synchronisation

**Principe** : Bidirectionnelle avec résolution de conflits

#### Mode Online :

- Toutes les opérations sont effectuées directement sur Supabase
- RéPLICATION automatique vers SQLite local en arrière-plan
- Cache local pour performances

#### Mode Offline :

- Toutes les opérations sont enregistrées dans SQLite
- Marquées avec un flag `sync_pending = true`

- Timestamp local enregistré
- Mise en file d'attente pour synchronisation

#### **Retour Online :**

1. Détection de la connexion
2. Pull des changements distants (depuis dernier sync)
3. Résolution des conflits
4. Push des changements locaux
5. Vérification d'intégrité
6. Mise à jour du timestamp de sync

#### **3.2.2 Gestion des Conflits**

##### **Règles de résolution :**

- **Modification serveur + modification locale sur même demande :**
  - Comparer timestamps
  - Priorité à la modification la plus récente
  - Si delta < 5 secondes : alerte utilisateur pour résolution manuelle
- **Suppression distante + modification locale :**
  - Alerter utilisateur
  - Proposition : annuler modification locale OU restaurer élément distant
- **Création de demande offline déjà existante :**
  - Peu probable (UUID), mais si détecté : merger les informations

#### **3.2.3 Tables de Synchronisation SQLite**

Structure miroir des tables Supabase avec ajouts :

- `sync_status` : 'synced', 'pending', 'conflict'
- `last_modified_local` : timestamp
- `last_modified_remote` : timestamp
- `conflict_data` : JSON (données en conflit)

### **3.3 Sécurité et Protection des Données**

#### **3.3.1 Sécurité Web**

- HTTPS obligatoire (Let's Encrypt)
- Content Security Policy (CSP) stricte
- Protection CSRF avec tokens
- Sanitization de tous les inputs
- Upload : validation type MIME + extension + taille
- Rate limiting : max 100 requêtes/minute par IP

#### **3.3.2 Sécurité Desktop**

- Stockage chiffré des credentials Supabase (via electron-store avec encryption)
- Base SQLite chiffrée (SQLCipher si données très sensibles)
- Pas de stockage de mot de passe en clair
- Auto-update sécurisé de l'application

#### **3.3.3 Protection des Données (RGPD-like)**

- Consentement explicite des parents pour traitement des données
- Droit d'accès : parents peuvent télécharger leurs données
- Droit de rectification : parents peuvent modifier leurs informations
- Droit à l'oubli : suppression complète des données sur demande (après validation)
- Journalisation des accès aux données sensibles

### **3.4 Performance et Optimisation**

#### **3.4.1 Application Web**

- Code splitting et lazy loading
- Images optimisées (Next.js Image)
- Caching agressif des assets statiques
- Compression Gzip/Brotli
- Temps de chargement initial < 3 secondes

- Lighthouse score > 90

### 3.4.2 Application Desktop

- Lazy loading des composants lourds
- Virtualisation des listes longues (react-window)
- Pagination des requêtes (50 items)
- Indexation SQLite sur colonnes recherchées fréquemment
- Démarrage app < 2 secondes

### 3.4.3 Base de Données

- Index sur colonnes fréquemment filtrées (statut, classe, date)
  - Requêtes optimisées avec EXPLAIN ANALYZE
  - Connexions poolées
  - Vacuum automatique SQLite
- 

## 4. EXIGENCES NON FONCTIONNELLES

### 4.1 Performance

- **Temps de réponse API** : < 500ms pour 95% des requêtes
- **Temps de chargement page web** : < 2 secondes
- **Upload fichier** : avec barre de progression, confirmation
- **Synchronisation** : < 30 secondes pour 100 demandes

### 4.2 Disponibilité

- **Application web** : 99% uptime (dépend Supabase)
- **Application desktop** : 100% en mode offline pour consultation
- **Maintenance planifiée** : notifications 48h à l'avance

### 4.3 Compatibilité

**Application Web :**

- Navigateurs : Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
- Responsive : desktop, tablette, mobile
- Pas d'application mobile native (PWA possible en phase 2)

#### **Application Desktop :**

- Windows 10/11 (64-bit)
- macOS 11+ (Big Sur et supérieurs)
- Linux Ubuntu 20.04+ (optionnel)

#### **4.4 Accessibilité**

- Conformité WCAG 2.1 niveau AA
- Navigation au clavier complète
- Labels ARIA appropriés
- Contrastes de couleurs suffisants
- Tailles de police ajustables

#### **4.5 Scalabilité**

- Support jusqu'à 500 demandes d'inscription par année scolaire
- 50 utilisateurs administratifs simultanés
- Stockage : 10 GB de documents (extensible)

#### **4.6 Sauvegarde et Récupération**

- **Sauvegardes automatiques** : quotidiennes (Supabase)
  - **Rétention** : 30 jours minimum
  - **RPO (Recovery Point Objective)** : 24h maximum
  - **RTO (Recovery Time Objective)** : 4h maximum
  - **Export régulier** : possibilité d'export complet mensuel
-

## 5. INTERFACES ET DESIGN

### 5.1 Charte Graphique

- **Couleurs principales** : à définir avec l'école (ex: bleu (#1E40AF), vert (#059669))
- **Typographie** : Inter ou Roboto (Google Fonts)
- **Logo** : fourni par l'école (formats SVG, PNG)
- **Style** : moderne, épuré, professionnel

### 5.2 Principes UX

- Interface intuitive nécessitant formation minimale
- Feedback visuel immédiat sur toutes les actions
- Messages d'erreur clairs et constructifs
- Workflow d'inscription linéaire et guidé
- Indicateurs de progression visibles

### 5.3 Maquettes (à produire)

- Wireframes basse fidélité
  - Maquettes haute fidélité (Figma)
  - Prototype cliquable pour validation
  - Guide de style (style guide)
- 

## 6. TESTS ET QUALITÉ

### 6.1 Stratégie de Tests

#### Tests Unitaires

- Couverture minimale : 70%
- Framework : Jest + React Testing Library
- Tests des fonctions métier critiques
- Tests des composants React

## Tests d'Intégration

- Tests des flux complets (inscription, validation, etc.)
- Tests de synchronisation online/offline
- Tests des appels API

## Tests End-to-End (E2E)

- Framework : Playwright ou Cypress
- Scénarios utilisateurs complets
- Tests sur navigateurs multiples
- Tests desktop avec Spectron

## Tests de Charge

- Simulation 100 inscriptions simultanées
- Test de synchronisation massive
- Monitoring des performances

## Tests de Sécurité

- Scan de vulnérabilités (npm audit, Snyk)
- Tests de pénétration basiques
- Validation des politiques RLS Supabase

## 6.2 Tests Utilisateurs

- Tests avec 3-5 parents (formulaire web)
- Tests avec 2-3 membres personnel administratif (app desktop)
- Recueil de feedback
- Itérations basées sur retours

## 6.3 Recette

- **Recette fonctionnelle** : validation de toutes les fonctionnalités du cahier des charges
- **Recette technique** : validation performances, sécurité, compatibilité

- **Recette utilisateur** : validation par utilisateurs finaux
  - **Critères d'acceptation** : définis pour chaque fonctionnalité
- 

## 7. DÉPLOIEMENT ET MIGRATION

### 7.1 Environnements

#### Développement

- Base Supabase dédiée (dev)
- Domaine : dev-inscription.ecoleprestige.cm

#### Staging/Préproduction

- Base Supabase dédiée (staging)
- Domaine : staging-inscription.ecoleprestige.cm
- Tests finaux avant production

#### Production

- Base Supabase production
- Domaine : inscription.ecoleprestige.cm
- SSL Let's Encrypt
- Monitoring activé

### 7.2 Déploiement Application Web

- **Hébergement** : Vercel (gratuit) ou Netlify, ou VPS
- **CI/CD** : GitHub Actions
- Pipeline automatisé : commit → tests → build → deploy
- Déploiement progressif (staging puis prod)

### 7.3 Distribution Application Desktop

- **Build** : Electron Builder
- **Formats** :

- Windows : installateur .exe (NSIS)
- macOS : .dmg + .app
- Linux : .AppImage (optionnel)
- **Distribution** : téléchargement depuis site web sécurisé
- **Auto-update** : electron-updater (updates silencieux)

## 7.4 Migration de Données Existantes

Si l'école a des données d'inscriptions existantes (Excel, papier) :

1. **Audit des données** : inventaire et nettoyage
  2. **Mapping** : correspondance ancien format → nouveau schéma
  3. **Scripts de migration** : Python ou Node.js
  4. **Import en masse** : via interface admin dédiée
  5. **Vérification** : contrôle d'intégrité post-import
  6. **Validation** : validation par école
- 

## 8. FORMATION ET DOCUMENTATION

### 8.1 Documentation Technique

- **README.md** : installation, configuration, lancement
- **Architecture** : diagrammes (base de données, flux, composants)
- **API Documentation** : endpoints, paramètres, réponses
- **Guide de déploiement** : étapes détaillées
- **Guide de contribution** : pour développeurs futurs

### 8.2 Documentation Utilisateur

#### Manuel Parents

- Création de compte
- Processus d'inscription pas à pas
- Suivi de demande

- FAQ

## **Manuel Administrateurs**

- Installation application desktop
- Utilisation quotidienne
- Gestion des demandes
- Rapports et statistiques
- Résolution de problèmes courants
- Mode offline

## **8.3 Formation du Personnel**

- **Session 1** : Introduction générale (1h)
- \*\*Session