

# PolyShare

## Plateforme de partage de photocopies offline-first

Projet étudiant - Sénégal

9 novembre 2025

*“Télécharge une fois, lis toute la semaine”*

### Table des matières

---

# 1 Vision du projet

---

## 1.1 Problématique

Les étudiants au Sénégal (UCAD, UGB, etc.) rencontrent plusieurs obstacles majeurs :

- ✖ Polycopiés hors de prix (2000-5000 FCFA par poly)
- ✖ Connexion internet faible et instable
- ✖ Bibliothèques surchargées et sous-équipées
- ✖ Diffusion anarchique via WhatsApp
- ✖ Pas de plateforme centralisée

## 1.2 Solution proposée

**PolyShare** : une PWA (Progressive Web App) permettant aux étudiants de :

- ✔ Partager gratuitement leurs polycopiés
- ✔ Télécharger des polys pour consultation offline
- ✔ Organiser leur bibliothèque par fac/filière/matière
- ✔ Fonctionner même sans connexion internet

## 1.3 Avantages compétitifs

1. **Offline-first** : fonctionne sans connexion constante
2. **Ultra-léger** : optimisé pour connexions faibles
3. **Gratuit** : pas de barrière financière
4. **PWA** : pas besoin du Play Store, installation directe
5. **Partage viral** : via QR code et Bluetooth (future)

## 2 Architecture technique

---

### 2.1 Stack technologique

#### 2.1.1 Frontend (PWA)

- **Framework** : React 18+ avec Vite
- **Styling** : TailwindCSS
- **Offline** : Service Worker + Workbox
- **Stockage local** : IndexedDB (via Dexie.js)
- **Lecture PDF** : PDF.js ou React-PDF
- **UI Components** : shadcn/ui ou Headless UI

#### 2.1.2 Backend (API REST)

- **Runtime** : Node.js 20+ avec Express.js
- **Base de données** : PostgreSQL (métadonnées)
- **Stockage fichiers** : AWS S3 ou Cloudinary
- **Cache** : Redis (optionnel phase 1)
- **Authentification** : JWT

#### 2.1.3 Hébergement

- **Frontend** : Vercel ou Netlify (gratuit)
- **Backend** : Railway, Render ou Fly.io (gratuit au début)
- **Base de données** : Supabase ou Neon (PostgreSQL gratuit)
- **CDN** : Cloudflare (gratuit)

### 2.2 Fonctionnalités principales

#### 2.2.1 Phase MVP (Minimum Viable Product)

1. Inscription/connexion simple (email + mot de passe)
2. Navigation par fac/filière/matière/semestre
3. Upload de photocopies (PDF uniquement)
4. Téléchargement pour lecture offline
5. Bibliothèque personnelle (stockage local)
6. Recherche basique
7. Système de karma (upload = points pour télécharger)

#### 2.2.2 Phase 2 (post-MVP)

- Annotations sur PDF
- Partage Bluetooth/Wi-Fi Direct
- Mode "Pack de la semaine"
- Signalement de contenu inapproprié
- Statistiques personnelles
- Notifications push

### **2.2.3 Phase 3 (monétisation)**

- Intégration Orange Money / Wave
- Abonnement Premium
- Anciens sujets d'examens
- Stockage cloud illimité

## 3 Planification détaillée

---

### 3.1 Phase 0 : Préparation (Semaine 1-2)

#### 3.1.1 Validation du besoin

- Interviewer 15-20 étudiants sénégalais
- Analyser la concurrence existante
- Définir les personas utilisateurs
- Valider les hypothèses clés
- Créer un document de spécifications fonctionnelles

#### 3.1.2 Design UX/UI

- Wireframes basse fidélité (papier/Excalidraw)
- Maquettes haute fidélité (Figma)
- Prototype cliquable
- Tests utilisateurs (5 personnes minimum)
- Itération sur le design

#### 3.1.3 Modélisation

- Diagramme de cas d'utilisation (UML)
- Diagramme de classes
- Modèle de données (MCD/MLD)
- Diagramme de séquence (actions principales)
- Architecture système

#### 3.1.4 Setup environnement

- Créer le repo GitHub
- Initialiser le projet frontend (Vite + React)
- Initialiser le projet backend (Express)
- Setup ESLint + Prettier
- Configurer le CI/CD de base

### 3.2 Phase 1 : Développement MVP (Semaine 3-8)

#### 3.2.1 Semaine 3-4 : Backend fondations

- Setup base de données PostgreSQL
- Modèles : User, Document, Category, University, Faculty
- API Auth : register, login, logout, refresh token
- API Documents : upload, list, get by ID, delete
- Middleware : auth, upload fichiers, validation
- Tests unitaires (Jest)

### 3.2.2 Semaine 5-6 : Frontend core

- Setup PWA (manifest.json, service worker)
- Pages : Home, Login, Register, Browse, Document Detail
- Composants réutilisables : Card, Button, Input, Modal
- Intégration API auth
- Intégration API documents (liste, détail)
- Système de routing (React Router)

### 3.2.3 Semaine 7 : Fonctionnalités offline

- Service Worker avec Workbox
- Cache stratégies (Cache First pour PDFs)
- IndexedDB setup avec Dexie.js
- Download manager (téléchargement + stockage local)
- Bibliothèque offline (lecture depuis IndexedDB)
- Sync queue pour uploads en attente

### 3.2.4 Semaine 8 : Upload et recherche

- Formulaire upload avec validation
- Compression PDF côté client (optionnel)
- Upload avec progression
- Système de recherche (fac/filière/matière/keyword)
- Filtres avancés
- Système de karma (upload = +10 points, download = -1 point)

## 3.3 Phase 2 : Tests et optimisation (Semaine 9-10)

### 3.3.1 Tests

- Tests end-to-end (Playwright ou Cypress)
- Tests de charge (combien d'utilisateurs simultanés?)
- Tests offline complets
- Tests sur connexion lente (throttling)
- Tests multi-devices (Android, iOS, Desktop)

### 3.3.2 Optimisations

- Optimisation taille bundle (code splitting)
- Lazy loading des composants
- Compression images et assets
- Optimisation requêtes SQL (indexes)
- Mise en cache API (Redis ou in-memory)

### 3.3.3 Sécurité

- Rate limiting (brute force protection)
- Validation robuste des uploads (type MIME, taille)
- Sanitization des inputs
- Protection CSRF
- Headers sécurisés (CORS, CSP)
- Scan vulnérabilités (npm audit)

## 3.4 Phase 3 : Déploiement et lancement (Semaine 11-12)

### 3.4.1 Déploiement

- Deploy backend sur Railway/Render
- Deploy frontend sur Vercel/Netlify
- Setup base de données production (Supabase)
- Configuration CDN Cloudflare
- Setup monitoring (Sentry pour errors)
- Setup analytics (Plausible ou Umami)

### 3.4.2 Bêta test

- Recruter 30 étudiants testeurs (UCAD priorité)
- Créer groupe WhatsApp support
- Onboarding des testeurs
- Collecte feedback quotidien
- Itérations rapides (hotfix en 24h)

### 3.4.3 Marketing pré-lancement

- Page landing explicative
- Tutoriel vidéo (3 min max)
- Affiches à imprimer (QR code)
- Posts réseaux sociaux (templates)
- Contacter délégués de classe
- Préparer pitch pour associations étudiantes

## 3.5 Phase 4 : Lancement public (Semaine 13+)

### 3.5.1 Semaine 13 : Lancement soft

- Annonce dans 5 groupes Facebook étudiants
- Partage dans 10 groupes WhatsApp
- Affiches dans 3 facultés UCAD
- Post LinkedIn ciblé Sénégal
- Objectif : 100 utilisateurs semaine 1

### **3.5.2 Semaine 14-16 : Croissance**

- Partenariat avec 2-3 associations étudiantes
- Campagne "Partage ton poly, gagne un abonnement"
- Premier rapport de métriques
- Identifier les power users (ambassadeurs)
- Expansion vers UGB si succès UCAD

## 4 Modélisation UML

---

### 4.1 Cas d'utilisation principaux

#### 4.1.1 Acteurs

- **Étudiant non connecté** : peut naviguer, voir les polys
- **Étudiant connecté** : peut télécharger, uploader
- **Modérateur** (future) : valide les uploads
- **Système** : gestion offline, sync

#### 4.1.2 Cas d'utilisation

1. S'inscrire / Se connecter
2. Naviguer par fac/filière/matière
3. Rechercher un polycopié
4. Télécharger un poly (pour offline)
5. Uploader un polycopié
6. Consulter sa bibliothèque offline
7. Gagner/dépenser des points karma
8. Signaler un contenu

### 4.2 Modèle de données

#### 4.2.1 Entités principales

```
User {  
  id: UUID (PK)  
  email: String (unique)  
  password_hash: String  
  username: String  
  karma_points: Integer (default: 10)  
  role: Enum ['student', 'moderator', 'admin']  
  university_id: UUID (FK)  
  created_at: Timestamp  
  updated_at: Timestamp  
}
```

```
University {  
  id: UUID (PK)  
  name: String (ex: "UCAD")  
  country: String (default: "Sénégal")  
  city: String  
}
```

```
Faculty {  
  id: UUID (PK)  
  name: String (ex: "Sciences et Techniques")  
  university_id: UUID (FK)  
}
```

```

Major {
  id: UUID (PK)
  name: String (ex: "Informatique")
  faculty_id: UUID (FK)
}

Course {
  id: UUID (PK)
  name: String (ex: "Algorithmique")
  code: String (ex: "INF0301")
  major_id: UUID (FK)
  semester: Integer
}

Document {
  id: UUID (PK)
  title: String
  description: Text
  course_id: UUID (FK)
  uploaded_by: UUID (FK -> User)
  file_url: String (S3/Cloudinary)
  file_size: Integer (bytes)
  file_hash: String (SHA256, évite doublons)
  downloads_count: Integer (default: 0)
  status: Enum ['pending', 'approved', 'rejected']
  created_at: Timestamp
  updated_at: Timestamp
}

Download {
  id: UUID (PK)
  user_id: UUID (FK)
  document_id: UUID (FK)
  downloaded_at: Timestamp
}

```

#### 4.2.2 Relations

- Un User appartient à une University (N-1)
- Une University a plusieurs Faculties (1-N)
- Une Faculty a plusieurs Majors (1-N)
- Un Major a plusieurs Courses (1-N)
- Un Course a plusieurs Documents (1-N)
- Un User peut uploader plusieurs Documents (1-N)
- Un User peut télécharger plusieurs Documents (N-N via Download)

## 5 Métriques de succès

---

### 5.1 Objectifs MVP (3 mois)

- **500 utilisateurs inscrits** (UCAD uniquement)
- **100 polycopiés uploadés**
- **2000 téléchargements** au total
- **40% de rétention** semaine 2
- **20% d'utilisateurs actifs hebdomadaires**

### 5.2 KPIs à suivre

1. **Acquisition** : nouveaux inscrits / semaine
2. **Activation** : % qui téléchargent au moins 1 poly
3. **Rétention** : utilisateurs actifs semaine N / semaine N-1
4. **Engagement** : temps moyen passé offline sur l'app
5. **Virilité** : % d'utilisateurs qui invitent d'autres
6. **Contribution** : ratio uploads / downloads

## 6 Budget et ressources

---

### 6.1 Coûts estimés (6 premiers mois)

- **Hébergement** : 0-20\$ / mois (gratuit au début)
- **Stockage S3** : ~10\$ / mois (500 polys)
- **Nom de domaine** : 12\$ / an (.sn ou .com)
- **Outils** : Figma (gratuit), VS Code (gratuit)
- **Marketing** : 50\$ (impression affiches)
- **TOTAL** : ~100\$ les 6 premiers mois

### 6.2 Temps requis

- **Développement MVP** : 10-15h / semaine × 10 semaines = 100-150h
- **Tests et déploiement** : 20h
- **Marketing et support** : 5h / semaine continu

## 7 Risques et mitigations

---

### 7.1 Risques techniques

1. **Stockage local limité sur téléphones**
  - Mitigation : compression PDFs, auto-nettoyage
2. **Service Worker complexe**
  - Mitigation : utiliser Workbox (simplifie)
3. **Upload de contenu inapproprié**
  - Mitigation : modération + signalement

## 7.2 Risques business

1. **Faible adoption initiale**  
→ Mitigation : focus sur 1 fac, ambassadeurs
2. **Problèmes de droits d'auteur**  
→ Mitigation : CGU claires, disclaimer
3. **Concurrence émergente**  
→ Mitigation : vitesse d'exécution, communauté

## 8 Prochaines étapes immédiates

---

### 8.1 Cette semaine

1. Valider l'idée : interviewer 10 étudiants
2. Setup repos : GitHub frontend + backend
3. Wireframes : esquisser les 5 écrans principaux
4. Modélisation : diagramme de classes basique

### 8.2 Semaine prochaine

1. Maquettes Figma : design système complet
2. Setup projet : Vite + React + Express
3. Premier endpoint : API register/login
4. Première page : Home avec navigation

**Let's build something that matters !**

Contact : [ton email]  
GitHub : [ton username]  
Localisation : France → Sénégal