

Optiflow – Résumé Complet de l'Application (MVP)

1. Vision & Objectif

Offrir aux PME un mini-ERP de gestion de stock **simple, abordable et "AI-powered"** qui réduit les ruptures, diminue l'immobilisation de trésorerie et apporte des recommandations exploitables en quelques clics.

2. Utilisateurs Cibles

Rôle	Besoin principal	Valeur ajoutée IA
Gestionnaire Stock	Suivre niveaux, programmer réappro	Alertes rupture & quantités optimales
Responsable Achats	Décider commandes fournisseurs	Prévisions demande & ROI produit
Dirigeant PME	Vue macro rentabilité	Scoring produit & rapports automatisés
Employé magasin	Encodage rapide mouvements	UI CRUD simplifiée

3. Fonctionnalités Clés (MVP)

1. **CRUD Standard** : produits, mouvements stock, ventes, fournisseurs.
2. **Prévision Demande (7-90 j)** : modèle Prophet/ARIMA → graphique confiance.
3. **Alertes IA** : rupture imminente, sur-stockage + suggestions déstockage.
4. **Recommandations Quantités** : calcul stock sécurité + forecast – stock actuel.
5. **Chat IA** : Q&A, rapports PDF on-demand, explications + score de confiance.
6. **Scoring Performance** : top/bad sellers, marge, rotation.
7. **XAI** : justification de chaque conseil (facteurs + historique).

4. Architecture Technique (Vue 10 000 ft)

```
[React/Next.js] -(tRPC)-> [Supabase Edge Functions] — PostgreSQL
                        |                               L Storage: modèles IA
                        |                               L LangChain RAG – LLMs (GPT-4o, Claude, Gemini)
                        |                               L Cron jobs & REST endpoints
```

- **Cursor Agent** génère/maintient code front & back.
- **Claude Code notebooks** prototypent, entraînent et exportent modèles ML.
- **Supabase** centralise Auth, Base de données, Functions, Cron et Storage.

5. Flux de Données IA

1. **Collecte** : ventes & mouvements historisés dans PostgreSQL.
2. **Training** : Notebook Claude → modèle `forecast_model.pkl` → Storage.
3. **Inference** : Edge Function `/forecast` charge modèle, renvoie JSON.
4. **Décision** : fonctions TypeScript calculent alertes + suggestions.
5. **Explication** : LangChain combine contexte + règles XAI, renvoie via Chat.

6. Sécurité & Gouvernance

- Auth Supabase (JWT) + RLS (Row Level Security) sur tables sensibles.
- RGPD : pas de donnée personnelle sensible ; logs IA anonymisés.
- Validations Zod côté API + front.

7. Hypothèses & Limites (MVP)

- **Données historiques** ≥ 3 mois pour une première prévision fiable.
- Monomarché, mono-devise (FCFA) pour simplifier.
- Pas d'intégration temps réel IoT (scanners, balances) en phase MVP.

8. Périmètre Hors-MVP (Future Work)

- Optimisation cash-flow, RL thresholds, simulateur prix.
- Multi-entrepôt, multi-devise, GED factures.
- API publique partenaires (marketplaces, e-commerce).

9. Glossaire Express

- **Forecast** : prédiction volumes vente futurs.
- **XAI** : eXplainable AI → expliquer décisions modèles.
- **RAG** : Retrieval-Augmented Generation → moteur Q&A combinant base vectorielle + LLM.

Document à valider avant démarrage phase 0.