DOSSIER TECHNIQUE – PGI Gestion de Vente de Vélos

1. Vue d'ensemble

1.1 Contexte

Le PGI a été développé pour remplacer les outils fragmentés utilisés par l'entreprise (Excel, applications indépendantes).
Objectif : **centraliser la gestion complète d'un magasin de vélos**, incluant plusieurs modèles, et variations (tailles, couleurs).

1.2 Modules couverts

- **Ventes** : gestion clients, nouvelle commande.
- **Achats** : gestion fournisseurs, commandes d'approvisionnement.
- **Stocks** : vélos, mouvements.
- **Facturation** : factures PDF.
- **Ressources humaines** : employés, paie.
- **Dashboard** : rapports, graphiques.

2. Architecture technique

2.1 Architecture 3 tiers

1. **Présentation** : HTML5 – CSS3 – JavaScript ES6 – Bootstrap 5
2. **Métier** : PHP 8 – Apache 2.4 (AMPPS / WAMP / XAMPP)
3. **Données** : MySQL 8.0

2.2 Pattern MVC

- **Model** : classes PHP pour l'accès aux tables (produits, ventes, etc.)
- **View** : pages HTML/PHP + composants Bootstrap
- **Controller** : scripts orchestrant la logique et les routes

2.3 Stack technique

Composant	Technologie
Langage	PHP 8.x
Base de données	MySQL 8
Frontend	HTML/CSS/JS + Bootstrap
IDE	Visual Studio Code
Versionnement	GitHub

3. Base de données

3.1 Structure principale

Ventes

ID, Client, Date, Total (€), Statut, Actions

Achats

ID, Fournisseur, Date, Montant total, Statut, Actions

Produits

Référence, Nom, Marque, Prix vente, Stock, Actions

Facturation

ID, Client, Commande, Date, Montant, Actions

RH

ID, Prénom, Nom, Email, Poste, Salaire, Date d’embauche, Actions

3.2 Optimisations

- **Vues SQL** : v_stock_disponible, v_ventes_mois, v_alertes_stock
- **Triggers** : mise à jour du stock lors d'une vente ou réception fournisseur
- **Procédures stockées** : creer_vente(), maj_stock(), add_mouvement_stock()
- **Indexation** sur produits, codes articles, emails, dates.

4. Développement Backend

4.1 Exemple de connexion PDO

```
// include/db.php
define('DB_HOST', 'localhost');
define('DB_NAME', 'pgi_velos');
define('DB_USER', 'root');
define('DB_PASS', '');
$pdo = new PDO(
  "mysql:host=".DB_HOST.";dbname=".DB_NAME.";charset=utf8mb4",
  DB_USER, DB_PASS,
  [
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
  ]
);
```

5. Fonctionnalités par module

5.1 Module Ventes

- Gestion des clients (particulier / pro)
- Vérification de stock automatique

5.2 Module Achats

- Bons de commande fournisseur
- Statuts : brouillon → envoyé → reçu → clôturé
- Réception partielle ou totale
- Mise à jour stock automatique

5.3 Module Produits

- Gestion des vélos
- Multi-tailles
- Alertes stock bas
- Inventaire annuel ou mensuel

5.4 Module Facturation

- Factures PDF

5.5 Module RH

- Gestion employés
- Calcul paie automatique
- Export fiche de paie PDF

6. Sécurité

Authentification

- Hachage bcrypt
- Sessions sécurisées
- Rôles : admin, manager, vendeur, employé, fournisseur

Protection

- Requêtes préparées
- Vérification rôle → accès

7. Tests et performances

7.1 Tests fonctionnels

Module	Test	Résultat
Ventes	création client + vente	OK
Achats	Nouvel achat	OK
Produits	inventaire	OK
Factures	génération PDF	OK
RH	Nouvel employé	OK

7.2 Performance

Opération	Objectif	Résultat
Dashboard	< 600 ms	OK
Liste produits	< 300 ms	OK
Création vente	< 1 s	OK

8. Maintenance

8.1 Sauvegardes automatiques

- Dump MySQL quotidien
- Conservation : 7 jours + 4 hebdomadaires
- Restauration testée chaque semaine

8.2 Nettoyage mensuel

```
DELETE FROM sessions
WHERE last_activity < DATE_SUB(NOW(), INTERVAL 30 DAY);
OPTIMIZE TABLE velos, stock, ventes;
```

9. Évolutions prévues

Court terme

- Inventaire mobile (QR code)
- Module comptabilité simplifiée

Moyen terme

- Synchronisation e-commerce (API REST)
- Outil d'analyse ventes (IA)

Long terme

- Migration vers framework Laravel / Symfony
- Application mobile vendeur
- Multi-boutiques

10. Ressources

Accès système

- URL locale : https://pgivelo.site/