

Projet 3 : Gestion d'une Coopérative

Systeme de Gestion Intégré

GUEI Jean Michel	TOURE Awa	ADJOUMANI Koffi Wilfried
DJATCHI Gnahoua Junior		SAVANE Syndou

Décembre 2025

Table des Matières

1	RAPPORT DE PROJET	3
1.1	Projet 3 : Gestion d'une Coopérative	3
1.2	MEMBRES DU GROUPE	3
1.3	TABLE DES MATIÈRES	4
1.4	1. INTRODUCTION	4
1.4.1	1.1 Contexte du Projet	4
1.4.2	1.2 Objectifs	4
1.4.3	1.3 Technologies Utilisés	4
1.5	Choix du Moteur de Stockage de MySQL	5
1.5.1	Justification Technique	5
1.5.1.1	1. Support des Transactions ACID	5
1.5.1.2	2. Gestion des Clés Étrangères	5
1.5.1.3	3. Verrouillage au Niveau Ligne	5
1.5.1.4	4. Récupération Automatique après Crash	5
1.5.1.5	5. Performances Optimisées	5
1.5.2	Comparaison avec MyISAM	6
1.5.2.1	Par conséquent,	6
1.5.3	2.1 Tables de la Base de Données	6
1.5.4	2.2 Diagramme de Dépendances Fonctionnelles	7
1.6	3. CRÉATION DE LA BASE DE DONNÉES & Alimentation de la Base de Données	7
1.6.1	Méthodologie Adoptée	7
1.6.1.1	1. Analyse des Tables	7
1.6.1.2	2. Conception du Schéma	7
1.6.1.3	3. Nettoyage des Données	8
1.6.1.4	4. Insertion des Données	8
1.6.2	Script de Création	8
1.6.3	3.2 Insertion des Données	10
1.7	4. FONCTIONS MÉTIER IMPLÉMENTÉES	11
1.7.1	4.1 Fonctions pour les Fournisseurs	11
1.7.1.1	a) nomFournisseur(codeF)	11
1.7.1.2	b) montantLiv(codeF, refPr, dateLiv)	11
1.7.1.3	c) montantTotalLiv(codeF, dateLiv)	11
1.7.1.4	d) nomFourCher(refPr)	11
1.7.2	4.2 Fonctions pour les Adhérents	12
1.7.2.1	a) nomAdherent(codeAdh)	12
1.7.2.2	b) montantCom(codeAdh, refPr, dateCom)	12

	1.7.2.3	c) montantTotalCom(codeAdh, dateCom)	12
1.7.3	4.3	Fonctions de Gestion	12
	1.7.3.1	a) benefice()	12
	1.7.3.2	b) gererStockLiv(codeF, refPr, qteLiv)	13
	1.7.3.3	c) gererStockCom(codeF, refPr, qteLiv)	13
1.7.4	4.4	Fonctions de Facturation	13
	1.7.4.1	a) factureAdh(codeAdh, date)	13
	1.7.4.2	b) factureFour(codeF, date)	13
	1.7.4.3	c) genererNumeroCommande(code, date)	13
1.8	5.	INTERFACE WEB	14
1.8.1	5.1	Page d'Accueil (index.php)	14
1.8.2	5.2	Édition de Factures	14
	1.8.2.1	Fonctionnalités :	14
	1.8.2.2	Éléments de la Facture :	14
1.8.3	5.3	Design et Ergonomie	15
1.9	6.	TESTS ET RÉSULTATS	15
1.9.1	6.1	Tests des Fonctions	15
1.9.2	6.2	Exemple de Facture Générée	15
1.9.3	6.3	Vérification des Contraintes d'Intégrité	16
1.10	7.	Déploiement du Projet	16
	1.10.1	1. Versioning sur GitHub	16
	1.10.2	2. Hébergement sur InfinityFree	17
	1.10.2.1	Configuration de l'Environnement	17
	1.10.2.2	Étapes de Déploiement	17
	1.10.2.3	Adaptation Serveur	17
	1.10.2.4	Optimisations Appliquées	18
1.11	8.	CONCLUSION	18
	1.11.1	8.1 Objectifs Atteints	18
	1.11.2	8.2 Compétences Acquises	18
	1.11.3	8.3 Améliorations Possibles	19
	1.11.4	8.4 Difficultés Rencontrées	19
1.12		ANNEXES	19
	1.12.1	Annexe A : Structure de la Base de Données	19
	1.12.2	Annexe B : Code Source Complet	19
	1.12.2.1	Structure du projet	19
	1.12.3	Annexe C : Manuel Utilisateur	20

Chapitre 1

RAPPORT DE PROJET

1.1 Projet 3 : Gestion d'une Coopérative

Formation : Data Analyst Industriel

Année Universitaire : 2025-2026

Date de soumission : Décembre 2025

Encadreur : K. M. BROU

1.2 MEMBRES DU GROUPE

1. GUEI JEAN MICHEL
 2. TOURE AWA
 3. ADJOUMANI KOFFI WILFRIED
 4. DJATCHI GNAHOUA JUNIOR
 5. SAVANE SYNDOU
-

1.3 TABLE DES MATIÈRES

1. Introduction
 2. Schéma Relationnel
 3. Création de la Base de Données
 4. Fonctions Métier Implémentées
 5. Interface Web
 6. Tests et Résultats
 7. Déploiement du Projet
 8. Conclusion
-

1.4 1. INTRODUCTION

1.4.1 1.1 Contexte du Projet

Ce projet consiste à développer un système de gestion pour une coopérative qui gère les commandes des adhérents et les livraisons des fournisseurs. Le système permet de suivre les stocks, calculer les bénéfices et éditer des factures.

1.4.2 1.2 Objectifs

- Créer une base de données relationnelle pour gérer les adhérents, fournisseurs, commandes et produits
- Développer des fonctions PHP pour les opérations métier
- Créer une interface web pour la gestion et l'édition de factures
- Assurer le suivi des stocks et des transactions

1.4.3 1.3 Technologies Utilisés

- **SGBD** : MySQL V8.0.43
- **Langage** : PHP V8.2.28
- **Interface** : PDO (PHP Data Object)
- **Serveur Web** : Apache/WAMPSEVER

- **Front-end** : HTML5, CSS3, JavaScript
- **IDE** : VS CODE

1.5 Choix du Moteur de Stockage de MySQL

1.5.1 Justification Technique

Le moteur de stockage **InnoDB** a été choisi pour ce projet pour les raisons suivantes :

1.5.1.1 1. Support des Transactions ACID

InnoDB garantit l'intégrité des données grâce aux propriétés ACID (Atomicité, Cohérence, Isolation, Durabilité). Ceci est crucial pour notre système de gestion de coopérative où les opérations de commandes et livraisons doivent être fiables et cohérentes.

1.5.1.2 2. Gestion des Clés Étrangères

InnoDB supporte nativement les contraintes de clés étrangères (**FOREIGN KEY**), essentielles pour maintenir l'intégrité référentielle entre nos tables (adhérents, fournisseurs, produits, commander, detailsliv). Cela empêche les insertions ou suppressions incohérentes.

1.5.1.3 3. Verrouillage au Niveau Ligne

Contrairement à MyISAM qui verrouille toute la table, InnoDB utilise un verrouillage au niveau des lignes. Cela améliore les performances lors d'accès concurrents, permettant à plusieurs utilisateurs de travailler simultanément sur différentes lignes de la même table.

1.5.1.4 4. Récupération Automatique après Crash

InnoDB dispose d'un mécanisme de journalisation (transaction logs) qui permet une récupération automatique des données en cas de panne du serveur, garantissant ainsi la pérennité des informations de la coopérative.

1.5.1.5 5. Performances Optimisées

InnoDB utilise un système de cache sophistiqué (Buffer Pool) qui améliore significativement les performances pour les opérations de lecture/écriture fréquentes, typiques dans

un système de gestion de coopérative.

1.5.2 Comparaison avec MyISAM

Critère	InnoDB	MyISAM
Transactions	Oui	Non
Clés étrangères	Oui	Non
Verrouillage	Niveau ligne	Niveau table
Récupération crash	Automatique	Manuelle
Intégrité données	Forte	Faible

1.5.2.1 Par conséquent,

Le choix d'InnoDB assure la **fiabilité**, la **cohérence** et les **performances** nécessaires pour un système de gestion de coopérative professionnel, tout en garantissant l'intégrité des données financières et transactionnelles critiques. ## 2. SCHÉMA RELATIONNEL

1.5.3 2.1 Tables de la Base de Données

ADHERENTS (numAdh, nomAdh, villeAdh, telAdh, emailAdh)

Clé primaire : numAdh

FOURNISSEURS (codeFour, nomFour, villeFour, telFour, emailFour)

Clé primaire : codeFour

PRODUITS (refProd, nomProd, prixVente)

Clé primaire : refProd

DATES (date)

Clé primaire : date

COMMANDER (numAdh, refProd, dateCom, qteCom, numCom)

Clés primaires : numAdh, refProd, dateCom

Clés étrangères :

- numAdh → ADHERENTS(numAdh)
- refProd → PRODUITS(refProd)
- dateCom → DATES(date)

DETAILSLIV (codeFour, refProd, dateLiv, qteLiv, prixAchat, numLiv)

Clés primaires : codeFour, refProd, dateLiv

Clés étrangères :

- codeFour → FOURNISSEURS(codeFour)
- refProd → PRODUITS(refProd)
- dateLiv → DATES(date)

1.5.4 2.2 Diagramme de Dépendances Fonctionnelles

- numAdh → nomAdh, villeAdh, telAdh, emailAdh
 - codeFour → nomFour, villeFour, telFour, emailFour
 - refProd → nomProd, prixVente
 - (numAdh, refProd, dateCom) → qteCom, numCom
 - (codeFour, refProd, dateLiv) → qteLiv, prixAchat, numLiv
-

1.6 3. CRÉATION DE LA BASE DE DONNÉES & Alimentation de la Base de Données

1.6.1 Méthodologie Adoptée

1.6.1.1 1. Analyse des Tables

Étude approfondie de la structure des tables (adherents, fournisseurs, produits, commander, detailsliv) pour comprendre les relations et les contraintes d'intégrité.

1.6.1.2 2. Conception du Schéma

Création du schéma relationnel avec définition des clés primaires, clés étrangères et contraintes d'intégrité référentielle. Aussi, nous avons défini des index afin de faciliter et

optimiser les requêtes dans notre BD.

1.6.1.3 3. Nettoyage des Données

- Correction des valeurs aberrantes (quantités invalides)
- Normalisation des formats (dates)
- Vérification de la cohérence inter-tables

1.6.1.4 4. Insertion des Données

Utilisation de requêtes SQL INSERT INTO en respectant l'ordre des dépendances :

1. **Tables** indépendantes : adherents, fournisseurs, produits, dates
2. **Tables** dépendantes : commander, detailsliv

Cette approche méthodique a garanti l'intégrité et la cohérence des données dès l'initialisation du système.

1.6.2 Script de Création

```
DROP DATABASE IF EXISTS Gestion_cooperative;
```

```
CREATE DATABASE Gestion_cooperative;
```

```
USE Gestion_cooperative;
```

```
--Table ADHERENTS
```

```
CREATE TABLE IF NOT EXISTS Adherents(
```

```
    numAdh VARCHAR(10) PRIMARY KEY NOT NULL,
```

```
    nomAdh VARCHAR(50) NOT NULL,
```

```
    villeAdh VARCHAR(50) NOT NULL,
```

```
    telAdh INT(10) NULL,
```

```
    emailAdh VARCHAR(50) NULL
```

```
)ENGINE=INNODB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE UNIQUE INDEX index_numAdh ON Adherents(numAdh, nomAdh, villeAdh, telAdh, emailAdh);
```

-- Table FOURNISSEURS

```
CREATE TABLE IF NOT EXISTS Fournisseurs(  
    codeFour VARCHAR(10) PRIMARY KEY NOT NULL,  
    nomFour VARCHAR(50) NOT NULL,  
    villeFour VARCHAR(50) NOT NULL,  
    telFour INT(10) NULL,  
    emailFour VARCHAR(50) NULL  
    )ENGINE=INNODB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;  
CREATE UNIQUE INDEX index_codeFour ON Fournisseurs(codeFour, nomFour, villeFour, telFour);
```

-- Table PRODUITS

```
CREATE TABLE IF NOT EXISTS Produits(  
    refProd VARCHAR(10) PRIMARY KEY NOT NULL,  
    nomProd VARCHAR(50) NOT NULL,  
    prixVente INT(10) NOT NULL  
    )ENGINE=INNODB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;  
  
CREATE UNIQUE INDEX index_refProd ON Produits(refProd, nomProd, prixVente);
```

-- Table DATES

```
CREATE TABLE IF NOT EXISTS Dates(  
    dates TIMESTAMP PRIMARY KEY  
    )ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

-- Table COMMANDER

```
CREATE TABLE IF NOT EXISTS Commander(  
    numAdh VARCHAR(10) NOT NULL,  
    refProd VARCHAR(10) NOT NULL,  
    dateCom TIMESTAMP NOT NULL,  
    qteCom INT NOT NULL,
```

```

CONSTRAINT contraintnumAdh
    FOREIGN KEY (numAdh) REFERENCES Adherents(numAdh)
    ON UPDATE CASCADE
    ON DELETE CASCADE,

CONSTRAINT contraintrefProd
    FOREIGN KEY (refProd) REFERENCES Produits(refProd)
    ON UPDATE CASCADE
    ON DELETE CASCADE

)ENGINE=INNODB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- Table DETAILSLIV
CREATE TABLE IF NOT EXISTS DetailsLiv(
    codeFour VARCHAR(10) NOT NULL,
    refProd VARCHAR(10) NOT NULL,
    dateLiv TIMESTAMP NOT NULL,
    qteLiv INT(10) NOT NULL,
    prixAchat INT NOT NULL,
    CONSTRAINT contraintcodeFour FOREIGN KEY (codeFour) REFERENCES Fournisseurs(codeF
    ON UPDATE CASCADE
    ON DELETE CASCADE,

    CONSTRAINT contrainterefProd FOREIGN KEY (refProd) REFERENCES Produits(refProd)
    ON UPDATE CASCADE
    ON DELETE CASCADE

)ENGINE=INNODB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

1.6.3 3.2 Insertion des Données

Les données ont été insérées conformément aux tables fournies dans le cahier des charges (voir annexe).

1.7 4. FONCTIONS MÉTIER IMPLÉMENTÉES

1.7.1 4.1 Fonctions pour les Fournisseurs

1.7.1.1 a) nomFournisseur(codeF)

Description : Retourne le nom d'un fournisseur connaissant son code

Code :

```
function nomFournisseur($codeF){  
    global $DB;  
    $sql = "SELECT nomFour FROM Fournisseurs WHERE codeFour=:codeF";  
    $query = $DB->prepare($sql);  
    $query->bindParam(':codeF', $codeF, PDO::PARAM_STR);  
    $query->execute();  
    $resultat = $query->fetch(PDO::FETCH_ASSOC);  
    return $resultat['nomFour'];  
}
```

1.7.1.2 b) montantLiv(codeF, refPr, dateLiv)

Description : Calcule le montant d'une livraison à une date donnée

Formule : $\text{Montant} = \text{qteLiv} \times \text{prixAchat}$

1.7.1.3 c) montantTotalLiv(codeF, dateLiv)

Description : Calcule le montant total des livraisons d'un fournisseur à une date donnée

Logique : Somme de tous les montants de livraison pour un fournisseur à une date

1.7.1.4 d) nomFourCher(refPr)

Description : Retourne le code du fournisseur le plus cher pour un produit donné

Résultat : [prixMax, date, codeFour]

1.7.2 4.2 Fonctions pour les Adhérents

1.7.2.1 a) nomAdherent(codeAdh)

Description : Retourne le nom d'un adhérent connaissant son code

1.7.2.2 b) montantCom(codeAdh, refPr, dateCom)

Description : Calcule le montant d'une commande à une date donnée

Formule : $\text{Montant} = \text{qteCom} \times \text{prixVente}$

1.7.2.3 c) montantTotalCom(codeAdh, dateCom)

Description : Calcule le montant total des commandes d'un adhérent à une date donnée

1.7.3 4.3 Fonctions de Gestion

1.7.3.1 a) benefice()

Description : Calcule le bénéfice global de l'entreprise

Formule : $\text{Bénéfice} = \text{Total Ventes} - \text{Total Achats}$

Code :

```
function benefice() {  
    global $DB;  
    $sql = "SELECT produits.prixVente, detailsliv.prixAchat, detailsliv.qtativ  
           FROM produits JOIN detailsliv  
           ON produits.refProd = detailsliv.refProd";  
    $query = $DB->prepare($sql);  
    $query->execute();  
    $resultat = $query->fetchAll(PDO::FETCH_ASSOC);  
  
    $Vente = 0;  
    $Achat = 0;  
  
    foreach($resultat as $row) {
```

```

        $Vente += $row["qteliv"] * $row["prixVente"];
        $Achat += $row["qteliv"] * $row["prixAchat"];
    }

    return $Vente - $Achat;
}

```

1.7.3.2 b) gererStockLiv(codeF, refPr, qteLiv)

Description : Permet de gérer le stock à chaque livraison

Logique : Calcule la somme des quantités livrées pour un produit

1.7.3.3 c) gererStockCom(codeF, refPr, qteLiv)

Description : Permet de gérer le stock à chaque commande

Logique : Stock disponible = qteLiv - somme(qteCom)

1.7.4 4.4 Fonctions de Facturation

1.7.4.1 a) factureAdh(codeAdh, date)

Description : Récupère les informations nécessaires pour éditer une facture adhérent

Données retournées : - Informations adhérent (nom, ville) - Liste des produits commandés - Quantités et prix

1.7.4.2 b) factureFour(codeF, date)

Description : Récupère les informations nécessaires pour éditer une facture fournisseur

1.7.4.3 c) genererNumeroCommande(code, date)

Description : Génère automatiquement un numéro de commande/livraison

Format : - Adhérent : CodeC{increment} (ex : A01C1) - Fournisseur : CodeL{increment} (ex : F01L1)

1.8 5. INTERFACE WEB

1.8.1 5.1 Page d'Accueil (index.php)

L'interface principale propose un menu avec les fonctionnalités suivantes :

- Affichage des tables
- Gestion des stocks (commandes et livraisons)
- Calcul des montants
- Recherche d'informations
- Édition de factures

1.8.2 5.2 Édition de Factures

1.8.2.1 Fonctionnalités :

- Saisie du code adhérent/fournisseur
- Sélection de la date
- Génération automatique de la facture

1.8.2.2 Éléments de la Facture :

1. **En-tête** : Logo et nom de l'entreprise
2. **Informations client/fournisseur** :
 - Numéro
 - Nom
 - Adresse
 - Numéro de commande/livraison
 - Date
3. **Tableau des produits** :
 - Produit acheté
 - Prix unitaire
 - Quantité
 - Montant HT
4. **Totaux** :
 - Total HT

— Total TTC (TVA 18%)

5. Montant en lettres

1.8.3 5.3 Design et Ergonomie

- Interface responsive et moderne
- Navigation intuitive
- Affichage clair des informations
- Impression facilitée des factures
- Formulaires validés

1.9 6. TESTS ET RÉSULTATS

1.9.1 6.1 Tests des Fonctions

Fonction	Paramètres Testés	Résultat Attendu	Résultat Obtenu	Statut
nomFournisseurF01		Toto	Toto	
montantLivF01, P01,	2021-01-01	30000	30000	
montantTotalF01, 2021-01-01		342500	342500	
benefice	-	Calcul correct	Calcul correct	
nomAdherentA01		Koffi	Koffi	
montantComA01, P01,	2021-01-01	36000	36000	
montantTotalComA01, 2021-01-01		104525	104525	
gererStockLivF01, P01, 120		Stock calculé	Stock calculé	
gererStockComF01, P01, 120		Stock restant	Stock restant	

1.9.2 6.2 Exemple de Facture Générée

Société Coopérative Union

Type : Fournisseur

Adresse : Abidjan

Nom Fournisseur: Froto

Numero Four : F02

Num Livraison : F02L4

Date : 01/01/2021

Produit acheté	Prix unit.	Quantité	Montant HT
CD	275	210	57 750
Sac ordi	15 000	870	13 050 000
Tablette PC	350 000	430	150 500 000
USB 8 Go	3 000	560	1 680 000

Total HT : 165 287 750 FCFA

Total TTC : 195 039 545 FCFA

Arrêtée la présente facture à la somme de :

Cent quatre-vingt-quinze millions trente-neuf mille cinq cent quarante-cinq francs CF

1.9.3 6.3 Vérification des Contraintes d'Intégrité

- Clés primaires uniques respectées
 - Clés étrangères validées
 - Contraintes de non-nullité appliquées
 - Intégrité référentielle maintenue
-

1.10 7. Déploiement du Projet

1.10.1 1. Versioning sur GitHub

Le projet a été versionné sur GitHub pour assurer la traçabilité et la collaboration :

Initialisation du dépôt Git

```
git init
```

```
git add .
```

```
git commit -m "Initial commit: Projet Gestion Coopérative"
```

Liaison avec GitHub

```
git remote add origin https://github.com/Adjoum/DataAnalytic_projet3_DB_gestion-coope
```

```
git push -u origin main
```

Dépôt GitHub : https://github.com/Adjoum/DataAnalytic_projet3_DB_gestion-cooperative

1.10.2 2. Hébergement sur InfinityFree

1.10.2.1 Configuration de l'Environnement

- **Plateforme** : InfinityFree (hébergement gratuit PHP/MySQL)
- **URL de production** : <https://gestion-cooperative.great-site.net/>
- **Base de données** : MySQL via phpMyAdmin

1.10.2.2 Étapes de Déploiement

1. **Création du compte** et configuration du site sur InfinityFree
2. **Création de la base MySQL** et récupération des credentials (hostname, username, password)
3. **Import de la base** via phpMyAdmin (fichiers schema.sql et data.sql)
4. **Upload des fichiers PHP** via FileZilla (FTP) dans le dossier /htdocs/
5. **Configuration de la connexion** : adaptation du fichier connexion_to_DB.php avec les paramètres InfinityFree
6. **Tests fonctionnels** : vérification de toutes les fonctionnalités en production

1.10.2.3 Adaptation Serveur

// Configuration InfinityFree

```
$DB_Name = '....';
```

```
$ServerName = 'sql303.infinityfree.com';
```

```
$User_Name = '.....';
```

```
$DB_Password = '....';
```

1.10.2.4 Optimisations Appliquées

- Utilisation de `DATE()` dans les requêtes SQL pour la compatibilité des dates
- Gestion des erreurs PDO pour le débogage en production
- Configuration du fichier `.htaccess` pour la performance et la sécurité

Le projet est désormais accessible publiquement et prêt pour la démonstration dans trois jours.

1.11 8. CONCLUSION

1.11.1 8.1 Objectifs Atteints

Le projet de gestion de coopérative a été réalisé avec succès. Toutes les fonctionnalités demandées ont été implémentées et testées :

Base de données créée avec toutes les tables et contraintes

Toutes les fonctions métier développées et fonctionnelles

Interface web intuitive et professionnelle

Édition de factures complète avec numérotation automatique

Gestion des stocks opérationnelle

Calcul du bénéfice global de l'entreprise

1.11.2 8.2 Compétences Acquises

- Conception de bases de données relationnelles
- Utilisation de PDO pour la connexion à la BD
- Utilisation des requêtes préparées de PHP pour la sécurité des requêtes
- Développement d'interfaces web professionnelles basique
- Utilisation parfaite des JOINTURES SQL, des contraintes d'intégrité
- Utilisation parfaite de la classe : `$f = new NumberFormatter("fr", NumberFormatter::SPELLING);` pour convertir les nombres en lettres
- Gestion de projets en équipe
- Tests et validation de fonctionnalités

- Révision en temps record pour certaines technologies : PHP, HTML, CSS, JS

1.11.3 8.3 Améliorations Possibles

- Ajout d'un système d'authentification
- Tableau de bord avec statistiques graphiques
- Export des factures en PDF
- Historique des modifications
- Gestion des utilisateurs avec rôles
- Notifications par email

1.11.4 8.4 Difficultés Rencontrées

- Gestion de la numérotation automatique des commandes
 - Calcul du stock disponible avec multiples livraisons
 - Mise en forme professionnelle des factures
 - Conversion des montants en lettres en français
-

1.12 ANNEXES

1.12.1 Annexe A : Structure de la Base de Données

Tables détaillées avec types de données et contraintes

1.12.2 Annexe B : Code Source Complet

1.12.2.1 Structure du projet

Fichiers PHP disponibles :

Projet_Groupe_3/

index.php

connexion_to_DB.php

close_DB.php

fonctions.php

afficherTable.php
benefice.php
gererStockCom.php
gererStockLiv.php
montantCom.php
montantLiv.php
montantTotalCom.php
montantTotalLiv.php
nomAdherent.php
nomFournisseur.php
nomFourCher.php
styles.css
sql/
 schema.sql
 data.sql
docs/
 rapport.pdf
 screenshots/
.gitignore
README.md
LICENSE

1.12.3 Annexe C : Manuel Utilisateur

Instructions pour l'utilisation du système

Fin du Rapport

*Ce rapport a été rédigé dans le cadre du Projet 3 : Gestion d'une Coopérative
Formation Data Analyst Industriel -INPHB - Année 2025-2026*