



**COLLÈGE STEE**  
SCIENCES ET TECHNOLOGIES  
POUR L'ÉNERGIE ET L'ENVIRONNEMENT

## Techniques de programmation

# Projet Final : Système de Gestion d'un Petit Commerce

Charaf Eddine Dridi <[charaf.eddine-dridi@univ-pau.fr](mailto:charaf.eddine-dridi@univ-pau.fr)>

25/03/2023

Ce projet a pour but de mettre en pratique les compétences acquises en programmation en C. Les étudiants apprendront à construire un système de gestion basique mais fonctionnel pour un petit commerce, développant ainsi des compétences en structuration de données, logique de programmation, et conception d'interface utilisateur en console.

### Acteurs du Système

- 1. Gestionnaire du Commerce** : Principal utilisateur du système, responsable de la gestion des produits, des ventes, des fournisseurs, et des clients.
- 2. Client** : Acheteur des produits, dont les informations et l'historique d'achat sont enregistrés dans le système.
- 3. Fournisseur** : Entreprise ou individu fournissant les produits au commerce, géré dans le système pour le suivi des commandes et des stocks.

### Fonctionnalités Obligatoires

- **Gestion des Produits** : Permettre au gestionnaire et aux employés d'ajouter, de supprimer, et de modifier des informations sur les produits.
- **Gestion des Ventes** : Enregistrer les transactions de vente, mettre à jour automatiquement les stocks, et permettre la consultation de l'historique des ventes.
- **Gestion des Clients** : Enregistrer de nouveaux clients dans le système, et consulter/modifier les informations et l'historique d'achat des clients existants.
- **Gestion des Fournisseurs** : Ajouter et gérer les informations sur les fournisseurs, suivre les commandes passées, et la mise à jour des stocks à la réception des commandes.

### Idéallement

Le programme assurera une gestion avancée et intégrée du petit commerce, offrant non seulement les fonctionnalités de base mais aussi des capacités étendues pour améliorer l'expérience utilisateur, la sécurité des données, et l'efficacité opérationnelle. Voici les extensions idéales du système :

- 1. Programme de Fidélité pour les Clients** : Implémentation d'un système de récompenses pour encourager et récompenser les achats répétés des clients.
- 2. Notifications Automatisées** : Envoyer des alertes aux fournisseurs pour les réassorts et informer les gestionnaires des mises à jour importantes.
- 4. Sécurité Renforcée** : Protéger les informations sensibles à travers le chiffrement et la gestion

sécurisée des mots de passe pour l'accès au système.

### Exigences et Modalités de Soumission

- **Persistence des Données** : Utiliser la manipulation de fichiers pour enregistrer des données entre les sessions d'utilisation. Pour une compréhension approfondie sur la manipulation de fichiers en C, consulter l'Annexe A.
- **Documentation** : Fournir une documentation du code et un rapport de projet (2 pages max) détaillant la structure du système, les choix de conception, et les instructions d'utilisation.
- **Évaluation** : Basée sur la fonctionnalité, la qualité du code, l'originalité des fonctionnalités supplémentaires, et la qualité du rapport de projet. Le projet est à réaliser individuellement ou en binôme. Vous rendrez votre projet contenant toutes les sources nécessaires ainsi que le rapport dans une archive ZIP avant le dimanche 16 avril.

Le zip devra être nommé en suivant le modèle suivant :

TechProg\_PROJET\_Prenom1Nom1\_Prenom2Nom2.zip

## Annexe A : Manipulation de Fichiers Binaires en C.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define MAX_PERSONS 2

struct Person {
    char name[50];
    int age;
};

int main() {
    FILE *file;
    struct Person people[MAX_PERSONS];

    // Écriture des données initiales dans un fichier binaire
    file = fopen("people.dat", "wb");
    if (file == NULL) {
        printf("Erreur lors de l'ouverture du fichier.");
        return 1;
    }

    for (int i = 0; i < MAX_PERSONS; i++) {
        printf("Entrez le nom de la personne %d : ", i + 1);
        scanf("%s", people[i].name);
        printf("Entrez l'âge de la personne %d : ", i + 1);
        scanf("%d", &people[i].age);
    }

    fwrite(people, sizeof(struct Person), MAX_PERSONS, file);
    fclose(file);

    // Modification des données dans le fichier binaire
    file = fopen("people.dat", "rb+");
    if (file == NULL) {
        printf("Erreur lors de l'ouverture du fichier.");
        return 1;
    }

    int personIndex;
    printf("Entrez l'index de la personne que vous voulez modifier (de 1 à %d) : ", MAX_PERSONS);
    scanf("%d", &personIndex);
    if (personIndex < 1 || personIndex > MAX_PERSONS) {
        printf("Index invalide.");
        return 1;
    }

    printf("Entrez le nouveau nom pour la personne %d : ", personIndex);
    scanf("%s", people[personIndex - 1].name);
    printf("Entrez le nouvel âge pour la personne %d : ", personIndex);
    scanf("%d", &people[personIndex - 1].age);

    fseek(file, (personIndex - 1) * sizeof(struct Person), SEEK_SET);
    fwrite(&people[personIndex - 1], sizeof(struct Person), 1, file);
    fclose(file);

    // Lecture des données modifiées à partir du fichier binaire
    file = fopen("people.dat", "rb");
    if (file == NULL) {
        printf("Erreur lors de l'ouverture du fichier.");
        return 1;
    }

    fread(people, sizeof(struct Person), MAX_PERSONS, file);

    printf("\nDonnées lues à partir du fichier après modification :\n");
    for (int i = 0; i < MAX_PERSONS; i++) {
        printf("Personne %d :\n", i + 1);
        printf("Nom : %s\n", people[i].name);
        printf("Âge : %d\n", people[i].age);
    }

    fclose(file);

    return 0;
}
```

Ce code en langage C permet de saisir les informations sur deux personnes (nom et âge), de les enregistrer dans un fichier binaire, de modifier les informations d'une des personnes enregistrées et enfin d'afficher les informations mises à jour à partir du fichier binaire.