



Mini-projet : Système de Gestion Simplifiée des Commandes pour un Café



Par Robert DIASSÉ

Description :

Ce projet consiste à développer une application Python permettant à un café de gérer efficacement les commandes des clients. L'application offrira les fonctionnalités suivantes :

1. **Affichage du menu** des articles disponibles avec leurs prix.
2. **Gestion des commandes** : les clients peuvent sélectionner plusieurs articles de manière interactive.
3. **Calcul du total** pour chaque commande client avec un récapitulatif des articles commandés.
4. **Suivi des ventes** : affichage du nombre total d'articles vendus et des revenus générés au cours de la journée.

Ce projet permet de résoudre un problème concret de gestion quotidienne dans un environnement de petite entreprise, tout en mettant en pratique les bases de Python, les structures de contrôle et les collections.

Marche à suivre pour le Système de Gestion Simplifiée des Commandes pour un Café

1. Structure du projet

Voici une suggestion pour l'organisation des fichiers :

```
CafeGestion/  
├── app.py           # Fichier principal pour exécuter l'application  
├── menu.py         # Gestion du menu (données des articles)  
└── commandes.py    # Gestion des commandes et des ventes
```

2. Explication des fichiers

- **app.py**
 - C'est le point d'entrée de l'application.
 - Ce fichier gère l'interaction utilisateur via des menus simples en ligne de commande.
 - Il fait appel à **menu.py** pour afficher les articles disponibles et à **commandes.py** pour gérer les commandes et les ventes.
- **menu.py**
 - Contient une liste des articles disponibles au café, avec leur nom et leur prix.

- Cette liste est une collection de dictionnaires, par exemple :

```
menu = [  
    {"article": "Café", "prix": 250.0},  
    {"article": "Thé", "prix": 200.0},  
    {"article": "Croissant", "prix": 300.0},  
    {"article": "Jus d'orange", "prix": 500.0}  
]
```

- **commandes.py**

- Contient les fonctions pour :
 - Enregistrer une commande.
 - Afficher un récapitulatif de la commande.
 - Calculer le total de la commande.
 - Suivre les ventes cumulées (articles vendus et revenus générés).
-

3. Fonctionnalités à implémenter

1. Affichage du menu

- Charger le menu depuis **menu.py**.
- Afficher chaque article avec son prix de manière lisible.

2. Prise de commande interactive

- L'utilisateur (serveur) choisit des articles en saisissant leur nom ou un identifiant.
- Permettre plusieurs sélections jusqu'à ce que la commande soit terminée.
- Ajouter chaque article sélectionné à une liste de commande.

3. Calcul du total de la commande

- Parcourir la liste des articles commandés.
- Calculer et afficher le total à payer.

4. Suivi des ventes

- Stocker le nombre total d'articles vendus et les revenus générés.
 - Afficher ces informations à la demande de l'utilisateur.
-

4. Exemple d'utilisation

```
=== Café Gestion ===  
1. Voir le menu  
2. Prendre une commande  
3. Suivi des ventes  
4. Quitter
```

Choisissez une option : 2

Entrez le nom de l'article (ou 'terminer' pour finir) : Café

Entrez le nom de l'article (ou 'terminer' pour finir) : Croissant

Entrez le nom de l'article (ou 'terminer' pour finir) : terminer

Commande enregistrée !

Récapitulatif :

- Café : 250.0 Fcfa

- Croissant : 300.0 Fcfa

Total : 550.0 Fcfa

Appuyez sur Entrée pour continuer...