**Guide d’utilisateur**

1. **Introduction**

Ce guide explique comment installer, configurer et utiliser le système de gestion d’inventaire.  
On y apprend à gérer les produits, suivre les ventes, analyser les stocks et générer des rapports exportables (CSV/JSON).  
Il couvre également l’utilisation de Git et GitHub pour gérer le code et VS Code pour exécuter et modifier le projet.

2. **Prérequis**

Avant de commencer, il faut :

* Python 3.10 ou + installé sur l’ordinateur.
* Git installé pour cloner et gérer le projet depuis GitHub.
* Un compte GitHub pour récupérer le projet ou sauvegarder les modifications.
* Visual Studio Code (VS Code) pour ouvrir et exécuter le projet.

3. **Installation des outils**

3.1 Installer Python

1. Aller sur [python.org](https://www.python.org/downloads/) et télécharger la dernière version 3.
2. Pendant l’installation, cocher “Add Python to PATH”.
3. Vérifier l’installation dans un terminal :
4. Python --version

3.2 Installer Git

1. Télécharger Git sur [git-scm.com](https://git-scm.com/downloads).
2. Installer Git et vérifier :
3. git --version

3.3 Créer un compte GitHub

1. Aller sur [github.com](https://github.com) et créer un compte gratuit.
2. Vérifier la connexion avec :
3. git config --global user.name "NomUtilisateur"
4. git config --global user.email "email@example.com"

3.4 Installer VS Code

1. Télécharger VS Code sur code.visualstudio.com.
2. Installer l’extension Python (Microsoft).

4. Récupération du projet

4.1 Trouver l’URL HTTPS du dépôt

1. Aller sur la page du dépôt GitHub.
2. Cliquer sur Code (vert) → HTTPS → copier l’URL, par exemple :
3. https://github.com/malini147/Gestionnaire\_inventaire.git

4.2 Cloner le projet

1. Ouvrir un terminal (ou terminal intégré VS Code).
2. Taper :
3. git clone https://github.com/nom-utilisateur/gestion-inventaire.git
4. cd gestion-inventaire

4.3 Installer les dépendances Python

Dans le dossier du projet :

pip install -r requirements.txt

5. Lancer l’application

1. Ouvrir VS Code et sélectionner File > Open Folder → dossier gestion-inventaire.
2. Dans le terminal intégré :
3. python main.py
4. Le menu principal s’affiche : Ajouter produit, Lister produits, Rapports, etc.

6. Utilisation complète

6.1 Gestion des produits

* Ajouter un produit : remplir nom, catégorie, fournisseur, prix, id, coût, description, SKU , catégorie, quantité en stock, date de création , date de modification, quantité.
* Modifier un produit : choisir le produit et mettre à jour les informations.
* Supprimer un produit : sélectionner le produit à retirer.
* Lister les produits : afficher tous les produits avec leurs détails (quantité, stock valorisé, marge…)

6.2 Gestion des catégories

* Ajouter une catégorie : remplir nom, description, id, date de création.
* Modifier une catégorie: choisir la catégorie et mettre à jour les informations.
* Supprimer un catégorie : sélectionner la catégorie à retirer.
* Lister les catégories: afficher tous les catégories avec leurs détails .

6.3 Gestion

* Ajouter un fournisseur : remplir nom, description, id, date de création.
* Modifier un fournisseur: choisir le fournisseur et mettre à jour les informations.
* Supprimer un fournisseur : sélectionner le fournisseur à retirer.
* Lister les fournisseurs: afficher tous les fournisseurs avec leurs détails .

6.2 Analyse et rapports

* Produits les plus ou moins vendus : le système calcule automatiquement la quantité vendue.
* Analyse par catégorie : regroupement des produits pour identifier les catégories les plus rentables.
* Rapports financiers : calcul du stock valorisé et de la marge bénéficiaire.

6.3 Export des rapports

* Les rapports peuvent être exportés en CSV ou JSON :
  + Menu “Rapports” → Choisir le format.
  + Les fichiers sont générés dans le dossier exports/.
  + Ouvrables dans Excel, LibreOffice ou tout logiciel compatible JSON.

7. Gestion du projet avec Git et GitHub

7.1 Faire un commit

Après avoir modifié ou ajouté des fichiers :

git add .

git commit -m "Message décrivant les modifications"

git push

7.2 Récupérer un projet existant

1. Copier l’URL HTTPS du dépôt GitHub.
2. Dans le terminal :

git clone URL

cd nom-dossier

1. Ouvrir le dossier dans VS Code.

7.3 Astuces Git

* git status → voir les fichiers modifiés.
* Faire des commits fréquents avec des messages clairs.
* Toujours pousser (git push) pour sauvegarder les modifications sur GitHub.

8. Bonnes pratiques

* Vérifier que Python et Git sont à jour.
* Sauvegarder régulièrement les fichiers CSV/JSON des rapports.
* Organiser les produits par catégories et fournisseurs pour des analyses plus efficaces.
* Utiliser les commits Git pour garder un historique clair des changements.

9. Dépannage rapide

| Problème | Solution |
| --- | --- |
| python n’est pas reconnu | Vérifier que Python est ajouté au PATH. |
| Module manquant | Refaire pip install -r requirements.txt |
| Export CSV introuvable | Vérifier le dossier exports/ |
| Git demande mot de passe | Configurer un token GitHub |

10. Support : Email

* [zaramalini147@gmail.com](mailto:zaramalini147@gmail.com)