

# Projet C# de Gestion d'Inventaire et de Stock

---

## Introduction

---

Bienvenue dans ce projet C# de gestion d'inventaire et de stock qui permettra de mettre en lien votre projet SQL, votre projet C# et votre prochain projet en PHP.

Vous travaillerez en groupe de trois pour développer une application C# qui permettra de gérer un inventaire de produits, d'effectuer des opérations CRUD (Create, Read, Update, Delete) sur les articles et les utilisateurs, de mettre en place un serveur web, d'explorer des notions avancées telles que l'asynchrone, les requêtes HTTP, l'interaction avec une base de données MySQL, la gestion de versions avec GitHub, la gestion de projet avec Trello, et la compartimentalisation des serveurs sous Debian.

## Ressources

---

- [Cours Initiation C#](#)
- [Installation Visual Code](#)
- [Développer sur Visual Code](#)

## Objectifs

---

- Mettre en place une API C# pour la gestion d'inventaire/stock.
- Réaliser des opérations CRUD pour les articles et les utilisateurs.
- Configurer un serveur web pour héberger l'API.
- Configurer un serveur pour héberger la base de données.
- Explorer les notions d'asynchronisme et de requêtes HTTP à l'aide de Postman.
- Interagir avec une base de données.
- Utiliser GitHub pour la gestion de versions.
- Organiser le projet et l'équipe avec Trello.
- Sécuriser le serveur de base de données en utilisant des contrôles d'accès (ACL).
- Gérer les utilisateurs et leurs mots de passe.

## Tâches principales

---

### 1. Mise en place de l'API C#

- Créez une API C# qui permet de gérer les articles et les utilisateurs.

- Assurez-vous que l'API est accessible via HTTP.
- Assurez-vous que l'API est bien documentée (fichier sous format PDF) **NB : Utilisez du natif, pas de framework autorisé (ASP.NET, etc.)**

## 2. Opérations CRUD

- Implémentez les opérations CRUD pour les articles (Créer, Lire, Mettre à jour, Supprimer).
- Fournissez des API endpoints pour ces opérations.

## 3. Serveur Web

- Configurez un serveur web pour héberger votre API.
- Assurez-vous que le serveur web est accessible via HTTP.
- Assurez-vous que l'API est accessible via HTTP.

## 4. Notions à aborder

- Explorez les notions d'asynchrone en C# pour rendre l'API plus performante.
  - [Lien vers les ressources sur l'asynchrone en C#](#)
- Familiarisez-vous avec les requêtes HTTP en utilisant Postman pour tester votre API.
  - [Lien installation Postman](#)
  - [Lien de comment utiliser Postman](#)
- Intégrez la gestion de la base de données MySQL dans votre application.
  - [Lien vers MySQL documentation](#)

## 5. GitHub

- Utilisez GitHub pour gérer la version de votre code.
- Assurez-vous de collaborer efficacement en utilisant des branches, des pull requests et des commentaires.

## 6. Gestion de projet avec Trello

- Utilisez Trello pour organiser les tâches de votre équipe (les checklists pour suivre le progrès des tâches, les dates d'échéance pour vous assurer que le projet est livré à temps, les étiquettes pour organiser les tâches par catégorie, etc.)
- Créez des tableaux, des listes, et des cartes pour suivre le progrès du projet.
- Assurez-vous que chaque membre de l'équipe ait un rôle clair et des tâches assignées.

## 7. Compartimentalisation des serveurs

- Configurez votre serveur web et votre serveur de base de données sur des machines distinctes sous Debian.
- Assurez-vous que les deux serveurs communiquent correctement entre eux.

## 8. Architecture

- Assurez-vous que votre code est bien structuré et organisé (interfaces, méthodes, classes, plusieurs fichiers, etc.)
- Assurez-vous que votre code est commenté.

## Options

---

En plus des tâches principales, vous pouvez explorer les options suivantes pour enrichir votre projet :

- Implémentez un système de passage de commandes et suivi de commandes.
- Ajoutez une fonctionnalité de recherche d'articles par nom et catégorie.
- Sécurisez davantage le serveur de base de données en utilisant des contrôles d'accès (ACL).
- Développez un système de gestion des utilisateurs avec une authentification sécurisée par mot de passe.
- Effectuer une interface graphique pour l'API.

Pour aller encore plus loin, vous pouvez réaliser des tests unitaires sur l'API.

- [Concept des test unitaires](#)
- [Procédure pas à pas](#)

## Rendu

---

- Le projet doit être rendu sous forme d'un lien vers un dépôt GitHub.
- Fichier PDF contenant la documentation de l'API.
- Rapport expliquant comment ont été gérées et configurées les machines virtuelles.

## Barème

---

- Mise en place de l'API C# : 8 points
- Serveur : 6 points
- GitHub et gestion de projet : 4 points
- Architecture : 2 points
- Options : +1.5 points par option
  - si réalisation de l'interface graphique avec toutes les fonctionnalités de base : +4 points