

**Titre :** *Création d'une boutique en ligne pour la vente de produits en utilisant C# et la technologie ASP.NET.*

<b>Cours</b>	Technologies du commerce électronique (INF27523)
<b>Session</b>	Hiver 2025
<b>Contact</b>	<a href="mailto:yacine_yaddaden@uqar.ca">yacine_yaddaden@uqar.ca</a>

## Table des matières

1. Objectif du travail.....	1
2. Technologies à utiliser.....	1
3. Description détaillée.....	1
3.1. Interface Web .....	1
3.2. Fonctionnalités à implémenter .....	2
3.2.1. Inscription, authentification et profil utilisateur .....	2
3.2.2. Affichage, parcours et filtrage des produits .....	2
3.2.3. Gestion des paniers .....	2
3.2.4. Paiement électronique .....	2
3.2.5. Génération et accès aux factures .....	2
3.3. Récupération des données.....	2
3.4. Stockage des données.....	2
3.5. Paiement électronique.....	2
4. Structure du projet.....	3
5. Modalité d'évaluation.....	3
6. Date de remise.....	3
7. Assistance.....	3
8. Points importants.....	3
9. Documentation .....	3

## 1. Objectif du travail

L'objectif principal de ce premier travail pratique est de concevoir et développer une boutique en ligne en utilisant le langage C# et le Framework web ASP.NET. Ce projet vise à mettre en application les connaissances acquises en classe, notamment en matière de conception de sites web, de gestion de la persistance des données et d'intégration des solutions de paiement électronique.

## 2. Technologies à utiliser

Pour mener à bien ce projet, les technologies suivantes seront utilisées :

- ☑ **Environnement de développement et outils :**
  - Microsoft Visual Studio 2022 Community (programmation).
  - Git pour la gestion de version.
  - GitHub pour héberger le dépôt du projet.
- ☑ **Langage de programmation :**
  - C# comme langage de programmation principal.
  - HTML/CSS et JavaScript pour l'interface Web.
- ☑ **Framework et Bibliothèques :**
  - **ASP.NET** pour la création du site web.
  - **Entity Framework Core** pour la persistance des données.
  - **Stripe** pour le paiement électronique.
  - **Twitter Bootstrap 5** pour l'interface Web.
- ☑ **Logiciel :**
  - **Postman** pour tester les API REST.

Il est recommandé de faire de la *gestion de version* avec l'outil [Git](#). Il est recommandé également de créer un dépôt sur GitHub.

## 3. Description détaillée

Afin de réaliser le projet, il est nécessaire de suivre les étapes suivantes :

### 3.1. Interface Web

En utilisant le Framework CSS **Twitter Bootstrap 5**, il est demandé de créer l'interface graphique avec les différentes pages (système de grille, composants, couleurs prédéfinies, etc.). De plus, il est demandé d'utiliser la technologie **RAZOR** comme moteur de *templates* d'ASP.NET pour faciliter la création des

différentes pages : page d'accueil, **inscription**, **authentification**, parcours des produits, panier, **paiement**, affichage des factures, **page d'erreur 404**, **profil utilisateur**, historique des commandes.

## 3.2. Fonctionnalités à implémenter

Dans le cadre de la réalisation de ce projet, il est demandé d'implémenter les fonctionnalités ci-dessous :

### 3.2.1. Inscription, authentification et profil utilisateur

Il y a deux profils utilisateur : l'un pour le **client** et l'autre pour le **vendeur**. Le client a la possibilité de passer commande pour un ou plusieurs produits, tandis que le vendeur peut mettre en ligne des produits. Il est nécessaire de prévoir une page d'inscription pour les nouveaux utilisateurs, une page d'authentification (avec gestion des sessions) et une page pour la gestion des informations des utilisateurs.

### 3.2.2. Affichage, parcours et filtrage des produits

Les produits ajoutés (via les API REST fournies) pourront être affichés pour les clients sur la page d'accueil. Pour chaque produit, il est nécessaire d'afficher les informations pertinentes, telles que le titre, la description, le prix, la catégorie, l'image, etc. Lorsqu'un client clique sur un produit, celui-ci est affiché sur une autre page. Il est également nécessaire d'ajouter une barre de recherche et de mettre en place un système de filtrage pour faciliter la navigation et la recherche de produits.

### 3.2.3. Gestion des paniers

Le client a la possibilité d'ajouter plusieurs produits dans son panier avant de procéder au paiement. Dans le panier, le client peut visualiser l'ensemble des produits ajoutés et avoir un aperçu du montant total à payer.

### 3.2.4. Paiement électronique

Le paiement s'effectue en utilisant **Stripe**, en fournissant les informations de la carte de crédit (nom du détenteur, numéro de la carte, date d'expiration, et CCV), ainsi que le code postal du détenteur. Il est important de s'assurer que le paiement a bien été validé. Il est également possible de visualiser les transactions sur le tableau de bord de Stripe.

### 3.2.5. Génération et accès aux factures

Une fois le paiement effectué, une facture est générée et liée au client, ainsi qu'aux vendeurs des produits. Le client pourra accéder à son historique d'achats ou à ses factures, tout comme le vendeur pourra accéder à ses propres informations. De plus, le client et le vendeur pourront consulter le montant total dépensé pour le client, ainsi que le montant total gagné pour le vendeur.

## 3.3. Récupération des données

Afin de faciliter le développement du site web, il est recommandé d'utiliser des données factices récupérées à partir d'**API REST publiques** et accessibles. Ces données incluent des informations sur les utilisateurs ainsi que sur les produits (descriptions, images, etc.), qui peuvent être récupérées automatiquement. Voici un exemple de deux API REST que vous pouvez utiliser :

☑ **Fake Store API** → <https://fakestoreapi.com/>

☑ **Dummy JSON** → <https://dummyjson.com/>

## 3.4. Stockage des données

Afin d'implémenter la persistance des données, il est demandé d'utiliser **Entity Framework Core** de Microsoft comme ORM. Pour cela, il est nécessaire d'installer les packages requis. De plus, il faudra créer les entités et la configuration (avec les tables) nécessaires pour assurer le stockage des données. Il est recommandé d'utiliser le serveur de gestion de base de données **SQL Server** installé avec Microsoft Visual Studio, mais il est également possible d'utiliser **MySQL Server**.

## 3.5. Paiement électronique

Afin de pouvoir effectuer un paiement en ligne avec **Stripe**, il est demandé de suivre les étapes définies dans le cours consacré aux passerelles de paiement en ligne, afin d'implémenter la fonctionnalité de paiement. Il est recommandé d'utiliser l'environnement de bac à sable (*sandbox*) de **Stripe**, qui permet de simuler et tester les transactions sans effectuer de paiements réels. Cette approche permet de visualiser les différentes transactions et de s'assurer que l'intégration fonctionne correctement avant de passer en mode production. Il est également important de configurer correctement les clés API, de sécuriser les transactions et de gérer les erreurs pour offrir une expérience utilisateur fluide et sécurisée.

## 4. Structure du projet

Lors de la création d'un nouveau projet dans **Microsoft Visual Studio 2022 Community**, une structure de base est automatiquement générée. Il est recommandé de :

- ✓ Organiser les fichiers en créant un dossier dédié aux **classes** du projet.
- ✓ Utiliser **Git** pour la gestion des versions et **GitHub** pour le partage.

## 5. Modalité d'évaluation

Le travail pratique comptera pour **25%** de la note du cours. Ces points sont répartis de la manière suivante :

Partie	Points
Interface Web	3,5%
Récupération des données	2,5%
Stockage des données avec Entity Framework Core	2,5%
Paiement électronique avec Stripe	3,0%
<b>Fonctionnalités à implémenter</b>	<b>10,0%</b>
→ Inscription et authentification	2,0%
→ Gestion du profil utilisateur	1,0%
→ Parcourir et filtrer les produits	2,5%
→ Gestion du panier	1,5%
→ Commande et paiement	1,5%
→ Gestion des factures	1,5%
Qualité du code source	1,0%
Qualité du rapport final	1,5%
La vidéo de présentation du projet	1,0%
<b>Total</b>	<b>25%</b>

## 6. Date de remise

- La date limite de remise est le **3 avril 2025 avant 23h00**
- Fichiers à remettre :
  - ✓ Un fichier compressé (.zip) contenant le code source,
  - ✓ Le rapport (un modèle **LaTeX** est fourni),
  - ✓ Une courte vidéo de démonstration faite avec [OBS Studio](#).

Tous les fichiers doivent être remis sur Moodle avant la date limite.

## 7. Assistance

Si vous avez besoin d'assistance, vous pouvez contacter l'auxiliaire d'enseignement, **Madame Dorra Lamouchi**, directement par courriel à l'adresse [Dorra.Lamouchi@uqar.ca](mailto:Dorra.Lamouchi@uqar.ca) afin d'organiser une rencontre sur ZOOM.

Des questions peuvent également être posées au professeur à la fin des séances de cours.

## 8. Points importants

### Note I :

- Le travail est en équipe de deux (voir la liste des équipes),
- Le professeur peut poser des questions liées au travail pratique,
- Le non-respect de l'énoncé entraînera une perte de points,
- Le plagiat sera sanctionné selon la politique de l'université,
- Le retard dans la remise du projet entraînera des pénalités.

**Note II :** Dans le cas où il y a des aspects qui ne sont pas clairs, n'hésitez pas à m'en faire part afin que je puisse apporter des éclaircissements et éventuellement mettre à jour l'énoncé du travail pratique.

## 9. Documentation

Il est recommandé d'utiliser la documentation officielle des différentes bibliothèques indiquées plus haut. Il est également possible de consulter d'autres documentations sur internet.