

IFT1005 - Design et développement Web

Dr. Amal Ben Abdellah

Automne, 2022

Travail Pratique 3

Ce travail compte pour 15% de la note finale du cours.

Travail à remettre le **Dimanche 18 Décembre 2022 avant minuit**, sur Studium

Les travaux pratiques doivent être faits individuellement ou en équipe de deux ou trois

Aucun retard ne sera accepté

1 Objectifs

Le but de ce travail pratique est de vous familiariser avec le langage JavaScript et avec la bibliothèque [jQuery](#).

Plus particulièrement, vous aurez à mettre en place plusieurs interactions sur différentes pages du site web afin que celui-ci devienne dynamique et fonctionnel. À la fin de ce travail, le site web sera utilisable, mais aucune mesure de sécurité ne sera présente puisque l'ensemble des opérations seront réalisées du côté client.

2 Introduction

Lors des deux premiers travaux pratiques, vous avez mis en place la structure et la mise en forme du site web à réaliser. Cependant, comme vous l'avez sans doute remarqué, celui-ci était complètement statique. En effet, il n'était pas possible d'interagir avec les formulaires ou d'afficher une liste de produits correspondant au critère de tri et à la catégorie sélectionnés, par exemple.

Afin d'être en mesure d'ajouter des interactions du côté client, le langage JavaScript doit être utilisé. En ce sens, ce langage permet entre autres de manipuler les éléments du [DOM](#) (Document Object Model), d'avoir accès aux API HTML (p. ex. Geolocation, Local Storage, etc.) et de réaliser des requêtes asynchrones ([AJAX](#)) pour récupérer des ressources ou pour communiquer avec des services web.

3 Travail à réaliser

À partir du code que vous avez produit lors des travaux pratiques précédents, vous aurez à ajouter la couche logique du site web grâce au langage JavaScript. Dans le cas où votre travail pratique 2 n'aurait pas été bien réussi, vous pouvez utiliser le corrigé du TP2 qui se trouve sur Studium afin de partir du bon pied pour ce troisième travail.

Afin de promouvoir la compréhension du langage JavaScript, seules les bibliothèques jQuery et jQuery Validation sont permises pour ce travail pratique.

Avant de débiter, assurez-vous d'inclure la bibliothèque jQuery. En ce sens, vous devez inclure la ligne suivante sur toutes les pages du site web : `<script type="text/javascript" src="./assets/js/jquery-3.2.1.min.js"></script>`

```
<script type="text/javascript" src="./assets/js/jquery-3.2.1.min.js"></script>
```

L'utilisation de jQuery est fortement recommandée pour ce travail pratique. En effet, cette bibliothèque simplifie la manipulation des éléments du DOM tout en fournissant des méthodes utiles pour la réalisation de requêtes AJAX.

Vous aurez besoin de sauvegarder les données du panier d'achats et celles des commandes afin de pouvoir les utiliser de page en page. Pour ce faire, vous devez utiliser l'API [Local Storage](#) de HTML5. Plusieurs sites offrent des tutoriels simples pour l'utilisation du Local Storage, dont [celui-ci](#). Vous pouvez visualiser l'état courant du Local Storage avec l'outil de débogage de votre navigateur web (raccourci F12) sous l'onglet "Application" (pour Google Chrome).

Les sous-sections qui suivent décrivent les éléments à réaliser pour les différentes parties du site web. Il est à noter que le ou les fichiers qui contiendront votre code JavaScript doivent se trouver dans le dossier assets/scripts.

3.1 Entête

1. Lorsqu'aucun élément est présent dans le panier d'achats, le badge indiquant le nombre de produits se trouvant dans le panier (`.shopping-cart > .count`) ne doit pas être visible. La figure 1a se trouvant en annexe illustre ce cas.
2. Lorsqu'un ou plusieurs produits sont présents dans le panier, le badge indiquant le nombre d'items qui se trouvent dans le panier doit être visible et celui-ci doit indiquer le bon nombre de produits en tout temps, et ce, sur toutes les pages du site web. La figure 1b (voir annexe) montre l'affichage attendu lorsqu'un item se trouve dans le panier.

3.2 Page des produits (products.html)

1. La liste des produits (`id="products-list"`) doit correspondre aux produits listés dans le fichier "products.json". Pour ce faire, vous devez effectuer une requête AJAX

pour récupérer le fichier JSON et vous devez par la suite ajouter les informations récupérées à la liste. Assurez-vous que la liste est triée par prix croissant, c'est-à-dire le critère "Prix (bas-haut)" (critère par défaut). L'affichage attendu est illustré à la figure 2 (voir annexe).

2. Le groupe de boutons associé aux catégories des produits (id="product-categories") doit être fonctionnel. Ainsi, lorsqu'un bouton du groupe est cliqué, la catégorie associée à ce bouton doit être appliquée à la liste de produits. En ce sens, seuls les produits liés à la catégorie sélectionnée devront être visibles dans la liste des produits. Également, lorsqu'un bouton est cliqué, ce même bouton doit devenir sélectionné (class="selected"). Toutes ces opérations doivent être réalisées sans rafraîchissement de page. La figure 3 se trouvant en annexe illustre d'ailleurs l'affichage souhaité lorsque la catégorie "Appareil photo" est sélectionnée.
3. Le groupe de boutons liés au classement des produits (id="product-criteria") devrait avoir un comportement similaire au groupe de boutons relié aux catégories de produits. En effet, lorsqu'un bouton du groupe est cliqué, la liste de produits affichée devra être triée selon le critère de tri sélectionné. De plus, lorsqu'un bouton est cliqué, celui-ci doit devenir sélectionné (class="selected"). Toutes ces opérations doivent se faire sans rafraîchissement de page. La figure 4 (voir annexe) montre l'affichage attendu lorsque la catégorie "Consoles" et le critère de tri "Nom (A-Z)" sont sélectionnés.
4. Le texte associé au nombre de produits qui sont actuellement affichés dans la liste de produits (id="products-count") doit être mis à jour à chaque fois qu'une nouvelle catégorie est sélectionnée. En ce sens, ce texte doit indiquer en tout temps le bon compte de produits.
5. Lorsqu'un produit de la liste est cliqué, le lien pour accéder à ce produit devrait être de la forme suivante : `"/product.html?id=#productId"`. Cette forme correspond à un paramètre qui est passé avec l'URL. Dans ce cas-ci, le paramètre est "id", et celui-ci est égal à l'identifiant du produit (#productId).

3.3 Page d'un produit (product.html)

1. À partir du paramètre spécifié par l'URL nommé "id", la page d'un produit devrait afficher les informations associées au produit ayant le même numéro d'identifiant qui a été spécifié. Ainsi, le nom du produit (id="product-name"), l'image (id="product-image"), la description (id="product-desc"), les caractéristiques (id="product-features") ainsi que le prix (id="product-price") devront être mis à jour afin d'afficher les bonnes informations. Pour en savoir plus concernant la récupération d'un paramètre d'URL avec JavaScript, veuillez consulter ce [lien](#). La figure 5 (voir annexe) montre l'affichage attendu pour le produit "Console Wii".

2. Dans le cas où un identifiant invalide a été spécifié via le paramètre "id" de l'URL, le contenu principal de la page doit être vide. Uniquement un titre doit être présent et ce dernier doit indiquer "Page non trouvée". La figure 6 se trouvant en annexe illustre l'affichage souhaité lorsqu'un identifiant invalide est spécifié.
3. Le formulaire permettant d'ajouter le produit au panier (id="add-to-cart-form") doit être fonctionnel. Ainsi, il doit être possible de spécifier la quantité à ajouter au panier pour le produit courant. De plus, lorsque le bouton "Ajouter" est cliqué, les informations du formulaire doivent être utilisées pour ajouter le produit au panier. Cela doit se faire sans aucun rafraîchissement de page.
4. Lorsque le produit est ajouté au panier, une boîte de dialogue (id="dialog") doit faire son apparition en bas de la page afin de confirmer que le produit a été ajouté au panier. Cette notification doit être affichée pendant cinq secondes. Également, il faut que la quantité associée aux nombres d'items dans le panier d'achats (.shopping-cart > .count) soit mise à jour dès qu'un produit est ajouté. La figure 7 (voir annexe) montre l'affichage souhaité lorsqu'un produit est ajouté au panier.

3.4 Page du panier d'achats (shopping-cart.html)

1. Lorsqu'aucun élément ne se trouve dans le panier d'achats, seuls le titre de la page ainsi qu'un paragraphe indiquant "Aucun produit dans le panier." doivent être présents. La figure 8 qui se trouve en annexe montre l'affichage attendu lorsque le panier est vide.
2. Lorsqu'un ou plusieurs produits se trouvent dans le panier, ceux-ci doivent être listés en ordre alphabétique dans le tableau. Également, le prix de chacun des produits (class="price") ainsi que le total (id="total-amount") doivent être cohérents en fonction de la quantité de chacun des items. La figure 9 (voir annexe) montre l'affichage souhaité lorsque le panier contient des items.
3. Le bouton "X" (class="remove-item-button") qui est associé à un item particulier permet de supprimer l'item qui lui est associé dans le panier d'achats. Lorsqu'un bouton "X" est cliqué, une fenêtre de type **confirm** doit faire son apparition pour indiquer à l'utilisateur que l'item associé s'apprête à être supprimer (p. ex. "Voulez-vous supprimer le produit du panier ?"). Lorsque l'utilisateur confirme la suppression de l'item, ce dernier est supprimé du panier. Ainsi, la ligne du tableau correspondant à cet item doit être supprimée de la page et le total associé aux items dans le panier doit être mis à jour. De plus, dans le cas où cette suppression ferait en sorte que le panier devient vide, la page doit indiquer qu'il n'y a plus de produits. Toutes ces opérations doivent se faire sans rafraîchissement de page. La figure 10 qui se trouve en annexe illustre un exemple de boîte de dialogue qui doit être affichée avant la suppression d'un item dans le panier.

4. Le bouton "-" (class="remove-quantity-button") permet de diminuer d'un la quantité associée à un item particulier du panier. Lorsque la quantité d'un item est d'un, le bouton lié à ce même item doit être désactivé (voir la propriété [disabled](#)). Lorsque la quantité associée à un item est supérieure à un, le bouton de cet item doit être actif. Lorsque le bouton est cliqué, la quantité de l'item (class="quantity") doit être décrémentée d'un. De plus, le prix ainsi que le total doivent être mis à jour pour refléter les changements. Cela doit se faire sans rafraîchissement de page.
5. Le bouton "+" (class="add-quantity-button") sert à augmenter d'un la quantité pour item du panier d'achats. Lorsque le bouton est cliqué, la quantité de l'item (class="quantity") doit être ajustée. Également, le prix et le total doivent être actualisés. Le tout doit être fait sans rafraîchissement de page.
6. Lorsque le bouton qui permet de vider le panier (id="remove-all-items-button") est cliqué, une fenêtre de confirmation (confirm) doit faire son apparition pour indiquer à l'utilisateur que tous les items du panier s'apprêtent à être supprimés (p. ex. "Voulez vous supprimer tous les produits du panier ?"). Une fois que l'utilisateur a confirmé la suppression des éléments, les items doivent être supprimés et la page doit indiquer qu'aucun élément n'est présent dans le panier. Cela doit se faire sans rafraîchissement de page.

3.5 Page de commande (order.html)

Il est obligatoire d'utiliser la bibliothèque jQuery.validate pour effectuer la validation du formulaire. En ce sens, assurez-vous d'inclure les lignes de code suivantes sur la page de commande afin que la bibliothèque soit fonctionnelle :

```
<script type="text/javascript" src="./assets/js/jquery.validate.min.js"></script>
<script type="text/javascript" src="./assets/js/additional-methods.min.js"></script>
<script type="text/javascript" src="./assets/js/messages_fr.js"></script>
```

1. Le formulaire de commande (id="order-form") peut être envoyé uniquement lorsque tous les champs sont complets et validés.
2. Les champs du formulaire doivent avoir les contraintes suivantes :
 - Le prénom (id="first-name") et le nom (id="last-name") sont obligatoires et doivent avoir une longueur minimale de deux caractères ;
 - L'adresse courriel (id="email") est obligatoire et doit être valide (par exemple: amal.ben.abdellah@umonteral.ca) ;
 - Le numéro de téléphone (id="phone") est obligatoire et doit être un numéro de téléphone canadien valide ([phoneUs](#)) ;
 - Le numéro de carte de crédit (id="credit-card") est obligatoire et doit être valide (p. ex. 4111 1111 1111 1111 [numéro de carte Visa]) ;

- La date d'expiration de la carte de crédit (id="credit-card-expiry") est obligatoire et doit respecter le format mm/aa. Les valeurs possibles pour le mois (mm) sont 01 à 12. Celles possibles pour l'année (aa) sont 00 à 99. Puisqu'aucune méthode n'existe pour effectuer une validation d'une carte de crédit dans le format demandé avec jQuery.validate, vous devez écrire une [méthode de validation personnalisée](#) pour valider cette date. Il est conseillé d'utiliser une [expression régulière](#) (RegExp) pour réaliser cette validation. Le message d'erreur à inscrire si la date est invalide est : "La date d'expiration de votre carte de crédit est invalide."
3. Lorsque le formulaire est soumis et qu'un ou plusieurs champs ne sont pas valides, un message d'erreur doit être affiché en dessous de chacun des champs invalides. Ces messages d'erreurs doivent provenir de la bibliothèque jQuery.validate. De plus, cela doit se faire sans rafraîchissement de page. La figure 11 qui se trouve en annexe montre un formulaire soumis contenant plusieurs erreurs.
 4. Lorsque le formulaire est soumis et que toutes les informations des champs sont valides, les items présents dans le panier d'achats doivent être retirés et une commande doit être créée. Pour les besoins de ce travail pratique, assurez-vous de conserver uniquement le nom (prénom et nom) du client ainsi que le numéro de commande (numéro unique correspondant au nombre de commandes déjà effectuées [p. ex. la deuxième commande qui a été effectuée sera la commande #2]).

3.6 Page de confirmation (confirmation.html)

Le numéro de commande (id="confirmation-number") ainsi que le nom du client (id="name") doivent être affichés sur la page. La figure 12 (voir annexe) montre un exemple de page de confirmation qui est attendue.

4 Remise

- N'oubliez pas d'ajouter votre/vos nom(s) en commentaires au début de chacun de vos fichiers.
- Vous devez placer votre code HTML et le dossier "assets" dans un dossier compressé au format ZIP
- Nommez votre fichier archive avec votre nom, par exemple, BenAbdellah_TP3.zip.
- Vous devrez remettre votre travail directement sur Studium.
- Vous pouvez remettre le travail autant de fois que vous voulez jusqu'à la date demandée.
- Le navigateur web Google Chrome sera utilisé pour tester votre site web.

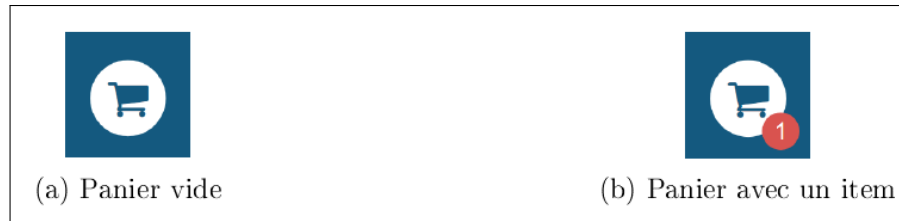


Figure 1: Badge indiquant le nombre de produits dans le panier

5 Barème

Globalement, vous serez évalué sur la qualité de votre mise en forme en CSS. Plus précisément, le barème de correction est le suivant :

- Respect des exigences pour la mise en forme
 - Entête (5 points)
 - Page des produits (17.5 points)
 - Page d'un produit (12,5 points)
 - Page du panier d'achats (20 points)
 - Page de commande (17.5 points)
 - Page de confirmation (2.5 points)
- Code JavaScript
 - Structure du code (15 points)
 - Qualité et clarté du code (10 points)

6 Annexe

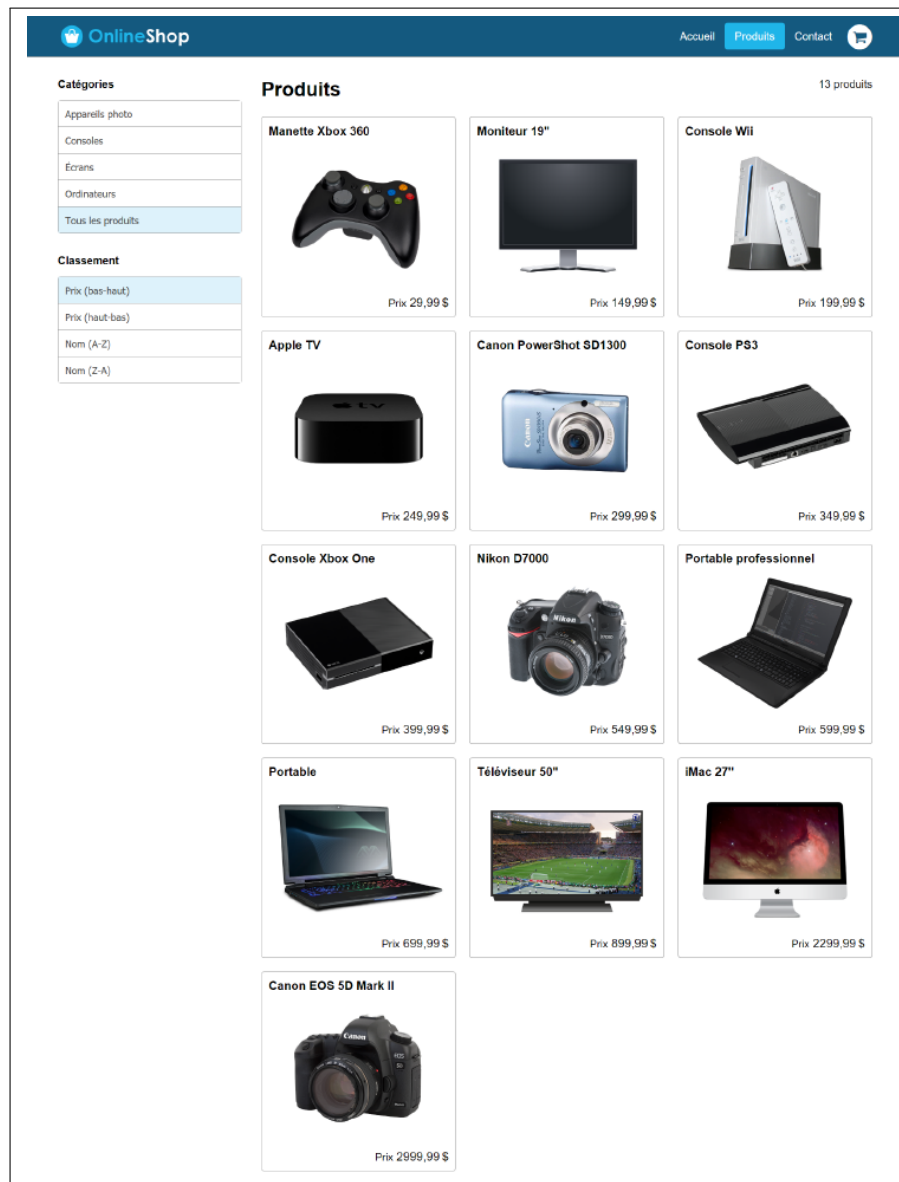


Figure 2: Affichage par défaut de la page des produits

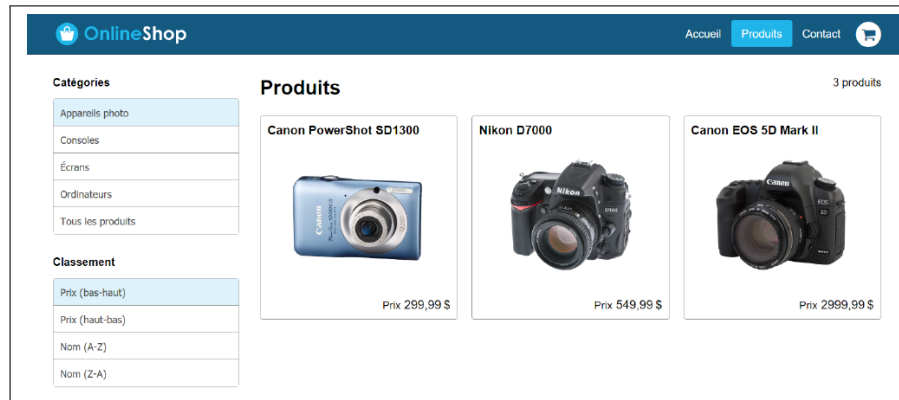


Figure 3: Affichage de la catégorie "Appareil photo"

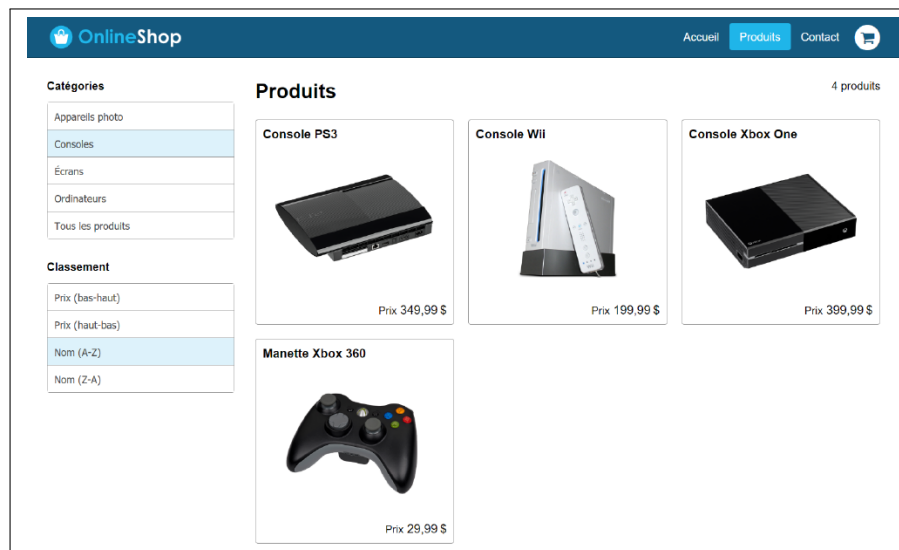


Figure 4: Affichage de la catégorie "Consoles" avec le critère de tri "Nom (A-Z) "



Figure 5: Affichage attendu pour la page d'un produit



Figure 6: Affichage désiré lorsqu'un identifiant de produit invalide est spécifié



Figure 7: Affichage souhaité lorsqu'un produit est ajouté dans le panier

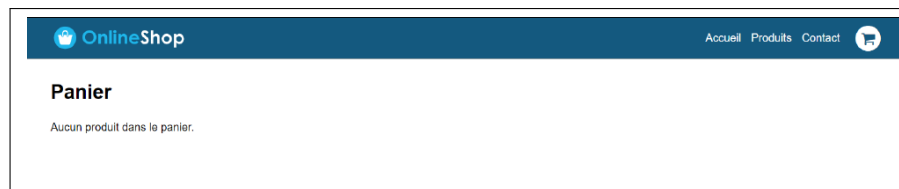


Figure 8: Affichage attendu lorsque le panier est vide

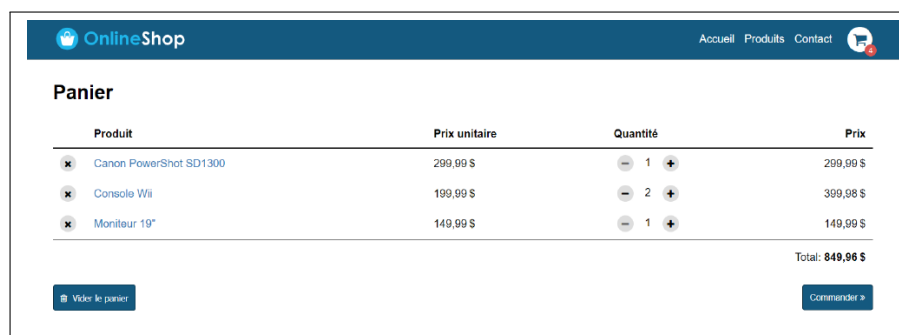


Figure 9: Affichage souhaité lorsque des produits se trouvent dans le panier

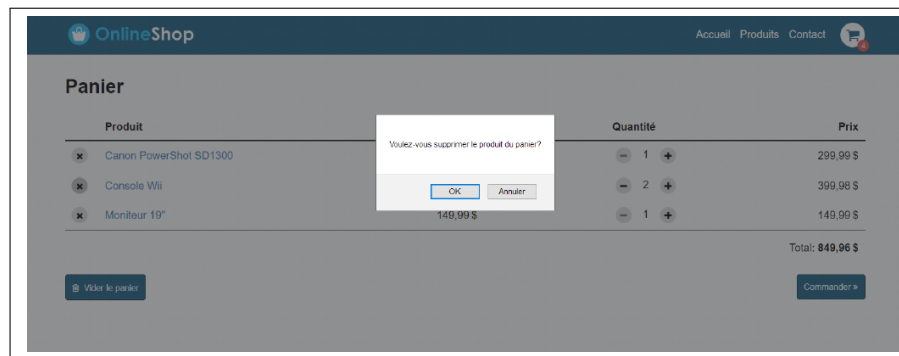


Figure 10: Exemple de boîte de dialogue avant la suppression d'un item dans le panier

The screenshot shows the 'Commande' (Checkout) page of the OnlineShop. The form contains several fields with validation errors indicated by red text:

- Contact**
 - Prénom**: Amal
 - Nom**: B (Error: Veuillez fournir au moins 2 caractères.)
 - Adresse courriel**: Adresse courriel (Error: Ce champ est obligatoire.)
 - Téléphone**: ###-###-#### (Error: Ce champ est obligatoire.)
- Paiement**
 - Numéro de carte de crédit**: **** * (Error: Ce champ est obligatoire.)
 - Expiration (mm/aa)**: 12 (Error: La date d'expiration de votre carte de credit est invalide.)

A 'Payer' (Pay) button is located at the bottom right of the form.

Figure 11: Formulaire de la page de commande contenant des erreurs

The screenshot shows the confirmation page of the OnlineShop. It displays the message: 'Votre commande est confirmée Amal Ben Abdellah!' (Your order is confirmed Amal Ben Abdellah!). Below this, it states: 'Votre numéro de confirmation est le 00001.' (Your confirmation number is 00001.).

Figure 12: Page de confirmation lorsque le formulaire soumis est valide