Webinaire JavaScript

Débuggez votre code JavaScript avec Google Chrome et VisualStudio Code

Description du webinaire : Dans cet exercice, vous allez voir comment utiliser les outils de développeur Google Chrome pour analyser et débugger du code JavaScript, mais aussi les outils de VisualStudio Code pour le code JavaScript back-end.

Exercice 1

🎯**Compétences visées : Débugger du code JavaScript front-end grâce aux outils de développeurs.**
Projets et paths concernés :

- Développeur Web P5, P6, P7
- Développeur Front-End P4-P7

Énoncé

L'entreprise ToyLand SARL, fabricant et vendeur de jouets depuis 1924 souhaite construire sa présence en ligne avec un site de e-commerce. Elle a fait appel à un développeur freelance pour réaliser son site. Celui-ci a, dans un premier temps, présenté une maquette qui a été validée par l'entreprise.

Une version dynamique a été développée pour pouvoir facilement mettre à jour les produits à afficher.

Cependant, le site ne fonctionne pas, et les produits ne s'affichent pas.
L'entreprise fait donc appel à vous, freelance en développement web, pour

finaliser le projet. Votre mission est donc d'identifier les problèmes sur l'application et les corriger.

Sources

- Retrouvez le code source du projet ici (branche "exercice-js").
- Retrouvez également, en annexe à la fin de ce document, la maquette du site. C'est à ça que devrait ressembler le site une fois vos corrections effectuées.

Conseils pour faire l'exercice

1. Installez le backend

Pour cet exercice, commencez par télécharger le code et lancer le serveur à partir du dossier backend :

- Au premier lancement, il faut installer les composants avec la commande : "npm install".
- Puis lancez le serveur avec la commande : "node server".

2. Débuggez le code

Ensuite, pour réussir les corrections, vous devez commencer par ouvrir les outils de développeur de votre navigateur afin d'analyser les problèmes et trouver des solutions.

Durant le webinaire, nous utiliserons Google Chrome, mais la plupart des outils sont également disponibles sur les autres navigateurs.

Le but de l'exercice est de trouver pourquoi les produits ne s'affichent pas. Ces produits proviennent du backend via la requête GET /api/products.

Voici les outils que vous pouvez utiliser dans Chrome pour identifier et régler les problèmes :

- Utilisez l'onglet Console pour identifier la premières erreur et la corriger.
- L'onglet Sources permet de voir le code JavaScript exécuté, et d'ajouter des

breakpoints pour voir l'exécution ligne par ligne.
    - Certains éléments de cet onglet peuvent être utiles, comme l'outil Watch et la Call Stack.
- L'onglet Network permet de voir les différentes requêtes vers le serveur et toutes les informations associées.
- Il y a en tout 3 erreurs à corriger, les deux premières seront affichés dans la console, la troisième est un problème de prix qui s'affiche en euros, alors que les deux derniers zéros devraient être des centimes
- Sur la page détail un problème se pose au chargement, il peut être identifié avec l'onglet network, mais il n'y a pas besoin de le corriger pour cet exercice.

Vous avez maintenant toutes les informations pour débugger cette application. A vous de jouer !

Ressources pour cet exercice

- Comment ouvrir les Chrome DevTools
- Utilisation de la Console sur Chrome
- Inspecter l'activité avec Network sur Chrome
- Débugger du JavaScript sur Chrome

Exercice 2

🎯**Compétences visées : Débuggez du code back-end grâce à Google Chrome et VisualStudio Code.**

Projets et paths concernés :

- Développeur Web P6, P7

Énoncé

Maintenant que la page d'accueil affiche bien les produits, il y a un problème lorsque l'on clique sur l'un d'entre eux, il ne s'affiche pas. Le JavaScript côté front a pourtant l'air bon, il faut aller voir côté back-end ce qui ne fonctionne pas.

Vous pouvez télécharger le code pour cet exercice sur ce repo.

Conseils pour faire l'exercice

Pour réussir ces corrections, vous devez commencer par ouvrir le projet backend avec VisualStudio Code et analyser le code ainsi que son architecture.

Ensuite, utilisez le debug de VisualStudio Code pour analyser le comportement du code à l'exécution. Vous pourrez trouver les différents problèmes qui empêchent la requête /:id de fonctionner correctement et ainsi les corriger.

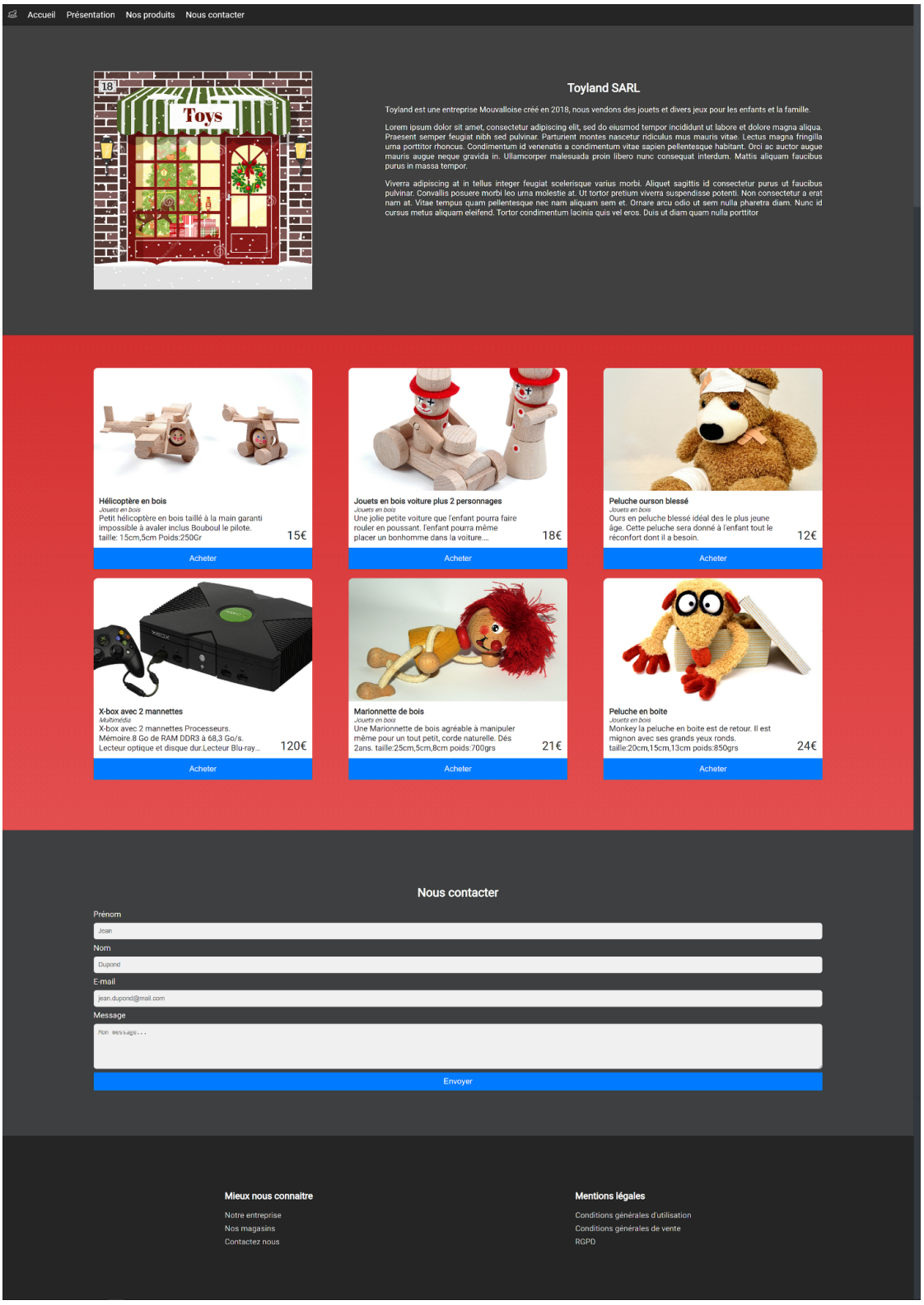
A vous de jouer !

Ressources pour cet exercice

- Débugger une application Node.js avec VisualStudio Code

Annexe : maquettes du site

Page d'accueil



Page détail produit

