

# Construisez un site e-commerce en JavaScript

MISSION

COURS

RESSOURCES

ÉVALUATION



90 heures

Mis à jour le vendredi 6 mai 2022

Ce projet a récemment été mis à jour pour l'améliorer ! Si vous avez commencé le projet avant le 29/09/2021, vous pouvez décider de continuer sur **l'ancienne version**. Dans ce cas, vous pouvez préciser "version avant 29-09-21" sur vos livrables.

Le nommage des livrables à déposer sur la plateforme a été changé et des indications sur les temps de soutenance ont été ajoutées le 22/03/2022.

## Avant de démarrer votre travail sur ce projet, nous vous conseillons de:

1. lire le scénario en entier, chaque section du projet ainsi que les documents fournis ;
2. consulter **le guide des étapes clés** avec des recommandations et des ressources pour organiser votre travail ;
3. préparer une liste de questions pour votre première session de mentorat.

## Scénario

Vous êtes en poste dans une agence de développement web depuis quelques semaines maintenant. Après avoir réalisé avec succès l'intégration de quelques sites web (HTML/CSS), on vous confie une nouvelle mission.

Votre client est Kanap, une marque de canapés qui vend ses produits depuis sa boutique exclusivement. Aujourd'hui, celle-ci souhaiterait avoir une plateforme de e-commerce en plus de sa boutique physique pour vendre ses produits sur Internet.



Logo de Kanap

Dans le cadre de cette mission, vous travaillez avec une équipe constituée de :

- Corinne, le CTO de l'agence ;
- Frank, le développeur front-end qui s'est chargé d'intégrer la maquette statique du site ;
- Bilal, le développeur back-end qui implémente l'API à laquelle est connecté le front-end.

Corinne vous envoie un e-mail pour vous briefer sur la mission :

**De :** Corinne

**À :** Vous

**Objet :** Site e-commerce Kanap

Hello !

Comme on en a discuté hier, voici les informations pour que tu puisses démarrer l'implémentation du site de Kanap de manière dynamique.

Voici les différentes tâches que tu vas devoir mener à bien :

- Unifier les travaux déjà réalisés par l'équipe en intégrant dynamiquement les éléments de l'API dans les différentes pages web avec JavaScript. Le code du front-end et de l'API est disponible sur [ce repo](#).

- Mettre en place un plan de test d'acceptation à partir de ce [template](#) que nous avons pour habitude d'utiliser.

Pour plus de précisions, voici les [spécifications techniques et fonctionnelles](#) du projet. Tu pourras y trouver tous les détails de celui-ci, les attentes pour chaque page du site web et les détails de l'API.

N'hésite pas à venir me voir si tu as la moindre question, ma porte est toujours ouverte.

Bonne journée,

**Corinne**

Un peu plus tard, Frank vous envoie un e-mail pour vous apporter quelques précisions complémentaires sur son travail :

**De :** Frank

**À :** Vous

**Objet** Maquettes statiques du site de Kanap

Salut,

Visiblement c'est le moment pour toi de rejoindre le projet ! Je viens donc te donner quelques informations sur la partie que j'ai pu réaliser, pour t'aider lors de ton développement.

4 pages ont été mises en place : page d'accueil, page Produit, page Panier et la page Confirmation. Sur l'ensemble des pages, toutes les parties statiques sont en place, elles sont donc prêtes à recevoir le contenu dynamique.

Aussi, sur chaque page, un exemple de la partie dynamique est systématiquement donné ; de cette façon, tu n'as pas à t'occuper de la mise en place de la structure HTML ni du style CSS, tout est déjà fait. Tu n'as plus qu'à t'occuper d'intégrer ces éléments dynamiquement grâce à JS et l'API.

Enfin, dans le code HTML j'ai intégré des "id" dans différentes balises, cela devrait t'aider à intégrer les éléments dynamiques. Avec tout ça, normalement tu n'auras pas besoin de toucher au code HTML/CSS.

Bon développement !

**Frank**

Ça y est, vous avez toutes les informations pour démarrer votre projet. Bon courage !

Pour ce projet, **vous ne pouvez utiliser que du code JavaScript pur**. L'utilisation de tout framework ou librairie JavaScript (React, Angular, Vue ou jQuery, par exemple) est interdite pour ce projet.

Pour vous aider à réaliser ce projet, [voici un exemple de découpage](#) des étapes à suivre. Vous y trouverez des conseils pour chaque étape, ainsi que sur l'utilisation des ressources pour ce projet.

## Livrables

- Un fichier ZIP contenant le **code fonctionnel du site web**.
- Un document PDF du **plan de test**.

Pour faciliter votre passage devant le jury, déposez sur la plateforme, dans un dossier zip nommé **“Titre\_du\_projet\_nom\_prénom”**, avec tous les livrables du projet comme suit : **Nom\_Prénom\_n° du livrable\_nom du livrable\_\_date de démarrage du projet**. Cela donnera :

- *Nom\_Prénom\_1\_code\_mmaaaa ;*
- *Nom\_Prénom\_2\_plan\_test\_mmaaaa.*

Par exemple, le premier livrable peut être nommé comme suit : *Dupont\_Jean\_1\_code\_012022*.

## Soutenance

Durant la présentation orale, l'évaluateur interprétera le rôle de Corinne. La soutenance est structurée de la manière suivante :

- **Présentation des livrables (15 minutes)**
  - Pendant 10 à 12 minutes, vous allez devoir présenter le site web dans lequel vous avez intégré les éléments dynamiques, ainsi que le code JavaScript réalisé.
  - Puis, pendant 3 à 5 minutes, vous devrez présenter le plan de test mis en place.
- **Discussion (10 minutes)**
  - L'évaluateur, qui pour rappel jouera le rôle de Corinne, vous challengera sur les différents points techniques du projet. Par exemple (mais pas uniquement) :
    - l'utilisation de l'API (GET, POST) ;
    - le DOM (ajout, suppression et modification d'éléments) ;
    - l'utilisation de URLSearchParams ;
    - l'utilisation de localStorage.
- **Débriefing (5 minutes)**
  - À la fin de la soutenance, l'évaluateur arrêtera de jouer le rôle de Corinne pour vous permettre de débriefer ensemble.

Votre présentation devrait durer 15 minutes (+/- 5 minutes). Puisque le respect de la durée des présentations est important en milieu professionnel, les présentations en dessous de 10 minutes ou au-dessus de 20 minutes peuvent être refusées.

## Compétences évaluées



Valider des données issues de sources externes



Gérer des événements JavaScript



Créer un plan de test pour une application



Interagir avec un web service avec JavaScript