

Guide d'étapes clés

Projet "Créez une page web dynamique avec JavaScript"

Comment utiliser ce document ?

Sur cette page, vous trouverez un exemple de découpage en étapes pour faire votre projet. Vous y trouverez :

- des recommandations pour réussir chaque étape ;
- les problèmes récurrents et points de vigilance à garder en tête ;
- une estimation de votre avancement sur l'ensemble du projet.

Ce découpage est simplement une suggestion pour vous organiser. Vous n'êtes pas obligé de compléter les étapes dans l'ordre.

Gardez en tête que votre progression sur les étapes n'est qu'une estimation, et sera différente selon votre vitesse de progression.

Recommandations générales

Dans ce projet, vous allez mettre en application les connaissances que vous avez acquises sur JavaScript.

Vous allez devoir réaliser plusieurs fonctionnalités et travailler avec des données dynamiques provenant d'une API.

Lorsque vous vous retrouvez confronté à ce genre de situation, le mieux est de découper l'application en petites tâches qui seront plus simples à réaliser successivement. Ce document est là pour vous aider ; cependant vous êtes libre de procéder à votre propre découpage.

Si vous faites votre propre découpage, pensez à le faire valider par votre mentor.

Étape 0 : Installez l'environnement de développement

5 % de progression

Avant de commencer, il est essentiel de prendre connaissance des différents éléments du projet mis à disposition, et d'installer tous les éléments nécessaires au bon démarrage du projet.

Avant de démarrer cette étape, je dois avoir :

- Terminé les cours JavaScript associés au projet.
- Installé Node.js et npm sur mon ordinateur.

Une fois cette étape terminée, je devrais avoir :

- Installé le back-end et le front-end du site.

Recommandations :

- Analysez en détail les différents éléments de l'énoncé : prenez connaissance du Kanban et du code mis à votre disposition, et n'hésitez pas à prendre des notes si nécessaire.
- Clonez le repo GitHub.
- Suivez le ReadMe pour l'installation des dépendances du dossier Backend pour l'installation des dépendances.
- Lancez le back-end du projet et découvrez la documentation Swagger de l'API.
- Faites un test de la route de récupération des travaux de l'architecte, via Swagger ou un outil comme Postman, afin de connaître des données existantes en base de données.

Points de vigilance :

- Afin de pouvoir faire fonctionner le back-end, vous devez avoir à disposition Node.js et npm sur votre ordinateur.

Ressources :

- [Site de Node.js](#)
- [Documenter une API avec Swagger](#) : Le tutoriel ici montre comment créer une documentation d'API avec Swagger. Cette ressource va vous aider à comprendre le principe de fonctionnement de Swagger pour voir comment bien utiliser le document fourni avec le back-end.

Étape 1 : Créez la page de présentation des travaux à partir du HTML existant

20 % de progression

Maintenant que tout est en place, vous pouvez commencer par ajouter le contenu dynamique du site.

Étape 1.1 : Récupération des travaux depuis le back-end

Utilisez fetch et récupérez les données provenant du back-end.

Avant de démarrer cette étape, je dois avoir :

- Installé l'environnement de développement.
- Testé que je récupère bien les informations du back-end (par exemple avec Postman ou Swagger).

Une fois cette étape terminée, je devrais avoir :

- La galerie fonctionnelle affichée avec la liste des travaux provenant du back-end.

Recommandations :

- Faites l'appel à l'API avec **fetch** afin de récupérer dynamiquement les projets de l'architecte.
- Utilisez JavaScript pour ajouter à la galerie les travaux de l'architecte que vous avez récupéré.
- Supprimez du HTML les travaux qui étaient présents. Il ne doit vous rester que le contenu que vous avez ajouté dynamiquement grâce à JavaScript.

Ressources :

- Le chapitre "[Envoyez une requête depuis le navigateur](#)" du cours "Créez des pages web dynamiques avec JavaScript" vous indiquera comment utiliser "fetch" pour faire vos appels API.
- Le chapitre "[Créez un nouvel élément dans une page web](#)" du cours "Apprenez à programmer avec JavaScript" vous aidera à ajouter les éléments dans la galerie.

Étape 1.2 : Réalisation du filtre des travaux

Ajoutez maintenant les filtres pour afficher les travaux par catégorie

Avant de démarrer cette étape, je dois avoir :

- Tous les projets s'affichant dans la galerie en provenance du back-end.

Une fois cette étape terminée, je devrais avoir :

- La possibilité de filtrer la galerie par catégorie de projet.

Recommandations :

- Il y a plusieurs façons ici de récupérer les catégories. En analysant les informations à votre disposition, tentez de répondre aux questions suivantes pour implémenter les catégories :
 - Ai-je suffisamment d'informations dans les données reçues ?
 - Dois-je faire un autre appel à l'API ?
- Au clic sur un élément du menu de catégories, filtrer les travaux selon le filtre sélectionné.

Points de vigilance :

- Il est important de garder une option de menu permettant d'afficher tous les travaux, comme par défaut.

Ressources :

- Découvrez [l'objet Set](#) et assurez-vous une liste de catégories sans doublons.

Étape 2 : Codez la page de connexion

40 % de progression

Maintenant que l'affichage du contenu est dynamique, il est temps de mettre en place les éléments nécessaires pour administrer le site. On commencera par la page de connexion.

Étape 2.1 : Intégration du design de la page de formulaire

Avant d'interagir avec le site, il faut pouvoir s'y connecter ; pour cela, intégrer la page de connexion en suivant le design de la maquette.

Avant de démarrer cette étape, je dois avoir :

- La page d'accueil avec la galerie fonctionnelle et filtrable par catégorie.

Une fois cette étape terminée, je devrais avoir :

- La page de login intégrée mais non fonctionnelle.

Recommandations :

- Avant de considérer comme conclu le travail d'intégration, demandez-vous :
 - Est-ce que le rendu est conforme à la maquette ?

Étape 2.2 : Authentification de l'utilisateur

Maintenant que la page de connexion est prête, nous allons pouvoir connecter notre utilisateur.

Avant de démarrer cette étape, je dois avoir :

- Le formulaire intégré à la page de connexion.

Une fois cette étape terminée, je devrais avoir :

- Le formulaire de connexion fonctionnel avec :
 - Redirection vers la page d'accueil quand la connexion est confirmée.
 - Un message d'erreur quand les informations utilisateur / mot de passe ne sont pas correctes.

Recommandations :

- Pour réaliser cette étape, demandez-vous :
 - Quel type de requête me permet d'envoyer les valeurs des entrées de mon formulaire ?
 - Si la combinaison utilisateur - mot de passe est correcte, comment rediriger vers la page d'accueil et s'assurer que la configuration est maintenue ?
 - Si la combinaison est fausse, comment prévenir l'utilisateur ?

Point de vigilance :

- Pensez à stocker le token d'authentification pour pouvoir réaliser les envois et suppressions de travaux.

Ressources :

- Le chapitre "[Sauvegarder les données grâce à une API HTTP](#)" du cours "Créez des pages web dynamiques avec JavaScript" vous aidera à envoyer une requête POST avec Fetch

Étape 3 : Ajoutez la modale

85 % de progression

Une fois que nous avons l'utilisateur connecté, il est temps de lui donner la possibilité de gérer ses travaux.

Étape 3.1 : Ajout de la fenêtre modale

Commençons donc par créer notre fenêtre modale et gérer son apparition et sa disparition.

Avant de démarrer cette étape, je dois avoir :

- La possibilité de me connecter comme administrateur du site.

Une fois cette étape terminée, je devrais avoir :

- La fenêtre modale fonctionnelle pour ajouter des médias dans la galerie. La modale doit pouvoir se déclencher au clic sur le bouton Modifier, et se refermer au clic sur la croix ou en dehors de la modale.

Recommandations :

- Pour bien construire la fenêtre modale, pensez à bien étudier les maquettes du site :
 - Que doit contenir ma modale ?
 - Quelles actions dois-je pouvoir entreprendre?

Points de vigilance :

- Lorsque vous créez la modale, assurez-vous que quel que soit le nombre de fois que vous ouvrez / fermez la modale, une seule modale est présente dans le code source.

Ressources :

- [Tutoriel Fenêtre modale Grafikart](#)

Étape 3.2 : Suppression de travaux existants

Dans cette étape, nous rendrons fonctionnelle la suppression des travaux, en communiquant avec l'API et en actualisant le DOM.

Avant de démarrer cette étape, je dois avoir :

- La fenêtre modale qui s'ouvre quand on clique sur le bouton pour ajouter un projet. Et qui se referme lorsque l'on clique en dehors de la modale.

Une fois cette étape terminée, je devrais avoir :

- La possibilité de supprimer un des travaux de l'architecte.

Recommandations :

- Faites bien attention à la requête. Regardez le document Swagger et demandez-vous : comment construire ma requête fetch pour supprimer un élément ?

Points de vigilance :

- On ne devrait pas avoir besoin de recharger la page pour voir que le projet a été supprimé. Assurez-vous donc de répondre à la question suivante :
 - Comment retirer des éléments du DOM après avoir reçu la confirmation de la suppression de l'entrée en base de données ?

Étape 3.3 : Envoi d'un nouveau projet au back-end via le formulaire de la modale

Nous allons maintenant interagir avec les formulaires, et ajouter de nouveaux travaux pour notre architecte.

Avant de démarrer cette étape, je dois avoir :

- La fenêtre modale qui s'ouvre quand on clique sur le bouton pour ajouter un projet. Et qui se referme lorsque l'on clique en dehors de la modale.

Une fois cette étape terminée, je devrais avoir :

- Un message d'erreur si le formulaire n'est pas correctement rempli.
- Une réponse de l'API si le formulaire est correctement envoyé.
- Si je recharge la page, le nouveau projet qui doit s'afficher dans la galerie.

Recommandations :

- Avant d'envoyer les informations via le fetch, pensez à vérifier :
 - Est-ce que j'ai bien toutes les informations nécessaires à l'envoi d'une nouvelle entrée ?

Ressources :

Voir cette information sur [l'utilisation des objets FormData](#) pour l'envoi des données du formulaire. *

Étape 3.4 : Traitement de la réponse de l'API pour afficher dynamiquement la nouvelle image de la modale.

Finalisons l'ajout de projet en actualisant de manière dynamique le DOM, afin d'ajouter nos travaux sans recharger la page.

Avant de démarrer cette étape, je dois avoir :

- Une réponse correcte de l'API quand j'envoie un nouveau projet depuis le formulaire à l'API.

Une fois cette étape terminée, je devrais avoir :

- L'ajout dynamique du projet dans la galerie après l'envoi du formulaire.

Recommandations :

- De la même manière que vous avez ajouté les projets existants du back-end au début du projet, analysez où et comment actualiser le DOM, afin d'afficher les nouveaux projets sans recharger la page.

Points de vigilance :

- N'oubliez pas d'ajouter la nouvelle image, non seulement dans le portfolio, mais également dans la liste des images se trouvant dans la modale.

Ressources :

- Le chapitre "[Créez un nouvel élément dans une page web](#)" du cours "Apprenez à programmer avec JavaScript" vous aidera à ajouter des éléments dans le DOM.

Étape 4 : Vérifiez l'ensemble du projet et s'assurer du traitement des erreurs dans les formulaires

100 % de progression

Avant de démarrer cette étape, je dois avoir :

- L'image qui s'ajoute au DOM.
- Toutes les fonctionnalités demandées implémentées.

Une fois cette étape terminée, je devrais avoir :

- Un projet testé et validé prêt pour la soutenance.

Recommandations :

- Testez votre code en vous posant les questions suivantes :
 - Comment se comportent mes formulaires si j'entre des données erronées ?
 - Est-ce que le visuel correspond aux attentes de la maquette ?
 - Comment la mise à jour de l'interface est-elle gérée quand je dois ajouter ou supprimer des éléments du DOM ?

Projet terminé !