



12/02/2025

Rapport des Test

Parcours CDA

LUCAS ALLAIN
CEFIM

Table des matières

Présentation	2
Présentation de l'application	2
Etapes de développement.....	3
Choix techniques et Stylistiques.....	4
Difficultés.....	4
Idées d'améliorations	5
Conclusion	6

Présentation

Tout d'abord, comme nous ne nous connaissons pas encore, je vais me présenter brièvement. Je me nomme Lucas ALLAIN (ALLAIN étant mon nom de famille) et j'ai candidaté à votre parcours de concepteur / développeur d'application en alternance. Suite à mon inscription, j'ai reçu les tests à effectuer, et aujourd'hui je vous présente le résultat de mon travail.

Présentation de l'application

Pour ces tests, il était demandé de réaliser une application web en HTML, CSS et JavaScript. Cette appli web permet la gestion de tâche, que l'utilisateur a créé et manipule. L'application doit être fonctionnelle, avec une interface utilisateur et le développeur doit explorer des techniques avancées de manipulation du DOM (Document Object Model) pour le JavaScript. L'idée étant de pouvoir fournir une expérience utilisateur fluide et agréable.

Cette application permet donc, de pouvoir créer une tâche en lui donnant un titre (obligatoire et limité à 100 caractères), une description (optionnelle et limité à 500 caractères), une date d'échéance et un niveau de priorité. Tout ceci se fait dans un formulaire avec un bouton pour pouvoir créer la tâche et l'ajouter à la liste.

Quant à la liste, elle permet de visualiser la liste des tâches existantes, on peut réaliser plusieurs actions sur cette liste. Il est possible d'appliquer un filtrage sur le statut de la tâche si l'on désire afficher uniquement les tâches « En cours » ou les tâches dites « Terminée ». On peut trier les tâches selon la date d'échéance par ordre croissant et décroissant ou bien les triées par niveaux de priorité croissant ou décroissant. Ces niveaux de priorités et les statut « Terminée » sont également visible par le fond de couleurs des tâches dans la liste pour pouvoir se repérer plus facilement. Enfin il est également possible de supprimer une tâche en cliquant sur le bouton de suppression à droite du tableau devant la tâche correspondante.

L'application sauvegarde également les données de la session, ce qui permet de retrouver ses tâches même si on a fermé la page.

Etapes de développement

Pour réaliser ce projet, j'ai d'abord commencé par faire la page HTML dans sa quasi-totalité. Puis j'y ai appliqué des styles CSS au fur et à mesure que j'avancais dans l'implémentation de nouvelles fonctionnalités avec JavaScript.

Donc, j'ai démarré par l'entête et le pied de page qui sont très simple, j'ai ensuite créé le formulaire de création de tâche, lui ai appliqué du style et l'ai rendu fonctionnel pour pouvoir plus tard remplir ma liste de tâches. Également, j'ai créé un jeu de données dans config.js pour ne pas être obligé de devoir créer des tâches à chaque fois que je rechargeais la page.

Par la suite, j'ai construit l'entête de mon tableau qui accueillera les futures tâches, j'y ai rajouté deux colonnes supplémentaires par rapport aux données que l'on rentre dans le formulaire : Statut et Action. Dans un premier temps, je me suis assuré que les tâches s'affichaient correctement en leur appliquant des styles CSS pour placer correctement chaque élément. Également, j'ai fait en sorte de gérer les comportements entre mon formulaire et ma liste pour leur alignement, leurs espacements, etc..., j'y ai même ajouté du responsive en dessous d'une certaine taille d'écran.

Comme ce n'était pas très pratique de créer des tâches sans arrêt à chaque fois que je voulais tester une nouvelle fonctionnalité, j'ai implémenté la sauvegarde des tâches dans le local Storage. Dès lors, j'ai rendu fonctionnel et persistant le fait de marquer une tâche comme « Terminée »

Puis vint le moment de rendre cette liste de tâches dynamique avec l'ajout d'un bouton de suppression pour chaque tâche. Puis avec la création d'un fond de couleur selon soit la priorité, soit son statut si la tâche est « Terminée ». On a aussi un filtre pour n'afficher que les tâches soit « En cours » soit « Terminée ». Pour finir, j'ai implémenté deux tris, un par date d'échéance croissant et décroissant, l'autre par priorité également croissant (« Elevée » → « Moyenne » → « Basse ») et décroissant (« Basse » → « Moyenne » → « Elevée »).

Choix techniques et Stylistiques

Pour réaliser cette appli web, j'ai utilisé Visual Studio Code, l'outil de versionnage GIT. Je n'ai pas utilisé cet GIT au début du développement du projet, car j'avais un sous-estimé l'importance du projet et j'avais estimé que je n'en aurais pas besoin. Or, je me suis rendu compte que l'investissement dans ces tests nécessitait l'utilisation de cet outil, d'autant qu'un dépôt Git en ligne et également une consigne de ces tests.

Enfin, je me suis servi de recherches internet pour me renseigner sur des éléments que je ne connaissais pas (comme localStorage) ou pour chercher des solutions à des problèmes que j'ai rencontrés. Également, j'ai utilisé chatGPT, qui m'a fourni de bonnes pistes à explorer en me fournissant des bouts de code que j'ai dû comprendre pour pouvoir les modifier et les implémenter selon mes besoins.

Concernant les choix de design, j'ai surtout suivi mon instinct en me demandant ce que moi en tant qu'utilisateur je souhaiterais pouvoir faire sur l'appli et ce à quoi je m'attendrais à y trouver. Ainsi on se retrouve avec un formulaire très épuré, un bouton de suppression de tâche bien visible et bien en face de la tâche qu'il va effectivement supprimer. Des options de filtrage et de tri sur les entêtes correspondants dans la liste de tâche. Des repères visuels pour les priorités des tâches avec leur fond de couleur. Et enfin le formulaire et la liste agencés de telle sorte que leurs proportions soient agréables pour la navigation.

Difficultés

En ce qui concerne les difficultés que j'ai rencontré, cela faisait un moment que je ne m'étais pas lancé dans un projet de cette envergure, certaines compétences en développements et certains réflexes (comme utiliser GIT) étaient un peu rouillés. Or, m'y replonger dedans, m'a permis de raviver ces compétences et réflexes.

Également, il y avait des choses que je ne connais pas (comme localStorage) sur lesquelles j'ai dû me renseigner, certaines fonctionnalités sur lesquelles je butais à les rendre fonctionnelles (par exemple la fonction `sort()`) et certains bugs qui m'ont fait perdre du temps. J'ai réussi à surmonter ces difficultés grâce aux ressources que j'ai pu trouver sur internet et à chatGPT (notamment pour déboguer mon code). Trouver la petite erreur d'inattention dans plusieurs dizaines de lignes, c'est bien pratique.

Idées d'améliorations

Bien que j'aie implémenté une partie des fonctionnalités de la partie bonus (localStorage, tri par date et priorité). Il en reste plusieurs que je n'ai implémenté dans l'appli web : la barre de recherche par titre, la barre de progression du pourcentage de tâches « Terminée » et des notifications pour les tâches proches de leur échéance.

A ces pistes d'amélioration, j'ai eu des idées supplémentaires de fonctionnalités ou modifications à apporter pour une éventuelle reprise du projet. Certaines plutôt modestes, d'autres plus conséquentes et enfin d'autres encore sur un aspect plus technique qui n'est pas forcément visible par l'utilisateur.

Pour les premières, on pourrait modifier le format dans lequel la date est affiché pour un format français, voire même localisé ou customisable (mois en toutes lettres ou non) selon les goûts de l'utilisateur. Ajouter une scroll bar pour la liste des tâches si cette dernière devient trop longue (la cantonnée à la taille de la page ?) ou faire en sorte que le formulaire d'ajout de tâche suive la navigation de l'utilisateur sur la page (le formulaire restant affiché en haut de l'écran). Une refonte visuelle du tableau avec les tâches. On pourrait également rendre le bouton d'ajout plus parlant, par exemple en y ajouter un petit symbole « + » devant le texte « Ajouter » pour qu'il saute plus aux yeux.

Dans la même lignée, des flèches ou autres éléments visuels pour indiquer le type de tri ou de filtrage sur la colonne. J'avais commencé à implémenter ces derniers mais je suis tombé sur un problème, en effet, les filtrages et tris se réinitialisent si on en sélectionne d'autres mais pas les éléments visuels. De plus, j'ai eu le sentiment que ce que j'avais commencé à produire n'était pas très parlant, j'ai donc décidé de commenter cette partie dans mon code pour pouvoir rendre ce livrable.

Ce qui m'amène à des choses plus conséquentes, un système de tri et de filtrage plus puissant et persistant que celui en place, ce qui permettrait de pouvoir appliquer plusieurs filtrages et tris en même temps, ce qui ressouderait le problème que j'ai rencontré pour l'ajout d'éléments visuels sur l'entête du tableau.

Plus ambitieux encore, intégrer un calendrier (sur une autre page ?) pour avoir un second affichage des tâches. Et enfin, la possibilité de créer un compte avec login et mot de passe, ce qui amènerait sûrement une autre manière d'enregistrer les données (avec une base de données MySQL ?).

Et pour conclure, sur un plan plus technique, une meilleure séparation des fichiers CSS et JavaScript si l'application venait à avoir plusieurs pages. Avec par exemple, un fichier style.css qui aurait des styles communs pour toutes les pages de l'application, accompagnée de fichiers plus spécifiques à certaines pages, comme Index.css qui ne gèrerait que des styles spécifiques pour la page index.html. Et même chose pour les fichiers JavaScript, un main.js pour lancer des appels à fonctions et eventlistener communs à toutes les pages, index.js qui gère ces appels uniquement pour index.js et toutes les fonctions stockées dans scripts.js (ou bien faire également des fichiers spécifiques ?)

Enfin, j'ai ajouté un bloc try/catch dans la fonction de création de tâches, on pourrait ajouter cette sécurité à d'autres endroits dans le code et le coupler à un système de notifications similaires aux tâches proches de leur échéance.

Conclusion

En conclusion, j'ai eu plaisir à faire ce projet, ça m'a motivé à réaliser d'autres projets personnels de développement informatique. J'ai retrouvé la satisfaction de voir un projet qui avance au fur et à mesure et qui soit fonctionnel et qui n'est pas un simple projet pour voir ce qui est possible de faire avec une certaine technologie, mais quelque chose qui peut réellement être utile. Également, j'ai pu consolider mes connaissances et compétences, notamment en JavaScript, car j'ai dû utiliser des notions que je ne connaissais pas et que j'ai dû apprendre à utiliser.

Concernant les critères d'évaluation, je pense avoir suivi les consignes comme demandé. L'appli web est fonctionnelle, les fonctionnalités obligatoires ont été implémentées et fonctionnent. Mon code a une structure logique et est commenté. L'expérience utilisateur est fluide et intuitive, du moins, je sais que j'ai créé une UX que j'aimerais avoir. Concernant le design (UI), il sert surtout l'UX, l'application n'est pas moche ou repoussante, mais il est évident qu'elle pourrait être plus attrayante (notamment en ce qui concerne le tableau des tâches) .

Enfin, j'espère que mon travail vous conviendra et qu'il me permettra d'être admis dans votre formation.