Test technique - Melody

Contexte

Tu as décidé de plonger dans le monde de la technologie, en débutant ton apprentissage du développement web. En tant que junior, tu es à la recherche d'une première expérience qui te permettra de progresser et d'acquérir de nouvelles compétences techniques.

Ce petit test technique te parvient pour évaluer tes capacités naissantes et surtout pour stimuler ta volonté d'apprendre. En fonction de tes résultats et de ceux des autres candidats, tu recevras une réponse personnalisée ainsi que des conseils pour continuer à avancer dans ton apprentissage.

Objectif

Ta mission, si tu l'acceptes, est d'intégrer une landing page selon une maquette définie. Tu as carte blanche sur la manière dont tu abordes cette intégration, que ce soit avec de la recherche, de la documentation voire consultation d'une IA !

Technologies

Pour la structure, on te demandera de faire tout simplement du **HMTL**.

Côté styles, CSS sera de toutes évidences le langage approprié, mais attention ! On te demandera de ne pas utiliser de bibliothèque tierce comme Bootstrap.

Pour terminer, le Javascript ! Comme il te sera nécessaire de dynamiser la page côté navigateur, ce langage sera ton meilleur allié. Cependant, tout comme pour le CSS, on te demandera de faire du vanilla. Pas besoin de sortir l'artillerie lourde avec React, ou d'utiliser une bibliothèque comme jQuery !

Critères d'évaluation

Tu t'en doutes très probablement, mais l'objectif de ce test technique est de nous permettra de découvrir qu'elles sont tes aisances et tes difficultés afin que nous puissions t'accompagner le mieux possible durant ton séjour chez nous !

Pour être en totale transparence avec toi, voici les critères sur lesquels nous seront attentifs :

Ta motivation et ta curiosité

Même si tu n'as pas su aller jusqu'au bout ou que tu bloques à une étape, on va vouloir découvrir ce que tu as essayé de mettre en place et les recherches que tu as pu faire pour tenter de te débloquer.

La propreté de ton code

On attendra de ton code d'être facile à lire et à comprendre. Cela va du nommage de tes IDs, classes, variables et fonctions, de l'indentation et des commentaires explicatifs si leur présence est nécessaire. Trop de commentaire tue le commentaire !

Le fonctionnement de ta réalisation

Bien que ce critère est moins important à nos yeux que les deux précédents, il s'agit tout de même d'un critère qui nous intéresse.

Là où nous seront beaucoup plus regardants, c'est sur le respect des consignes du test technique.

(Bonus) Le versionning de ton code

Si tu te sens suffisament à l'aise avec Github, on te recommande vivement de te créer un repository **public** afin de mettre en ligne ton test technique. Dans le cas où ça te semble trop alambiqué, ce n'est pas un souci pour nous.

Idéalement, on regardera comment tes commits sont nommés, leur fréquence tout comme la manière dont tu utilises les branches.

Livraison

Les fichiers attendus (à minima)

- index.html
- style.css
- app.js

Moyens de partage

- Github
- Archive ZIP sur boite de dépot :
  - https://kdrive.inseeders.io/app/collaborate/861905/1feb7cc0-77da-49f9-8323-8f62e42b3c06

L'envoi des livrables

Peu importe le moyen de partage (Github ou archive ZIP sur boite de dépôt), tu devras nous indiquer la remise de tes livrables par email à l'adresse info@inseeders.io.

Nous attendons ton retour pour le **dimanche 25 février à 23h59**.

En cas de difficulté (ou d'empêchement), n'hésite pas à nous le faire savoir.

Étapes de réalisation

0 – Préambule

Avant de lire le contenu des étapes ci-dessous, on t'invite fortement à prendre en considération l'étape 0 (qui n'en n'est pas vraiment une), puisqu'il s'agit de la plus importante de toutes.

Il est normal de se retrouver bloqué sur une étape : ça a été pris en considération.

Ce test technique n'est pas évident si on tient à le réaliser de A à Z en voulant faire en sorte que ce soit parfait.

Heureusement, ce n'est pas notre but. Si l'application est "parfaite", c'est top. Mais ce qui nous intéresse, c'est de comprendre comment **TU** arrives à avancer sur des objectifs, entre difficulté et trivialité.

Alors ne prends pas peur, tu as le droit de sauter une étape pour passer aux suivantes ! N'hésite pas à nous le préciser dans un commentaire, dans un commit ou par email, on essayera de t'apporter tous nos conseils pour que les prochaines fois tu sois en mesure de plus facilement t'y retrouver !

1 - Intégration HTML et styles basiques

Commence tranquillement avec le découpage des différents éléments qui composent la page, en prenant soin de faire attention aux balises utilisées.

Ne t'inquiète pas, tu pourras évidemment revenir sur la structure HTML par la suite !

Côté CSS, l'idée n'est pas de faire au pixel près, mais l'esthétique globale devra être respectée. Tu trouveras dans l'archive du test technique toutes les ressources nécessaires :

- Maquettes
- Codes héxadécimaux des couleurs
- Images
- Ressources utiles

Ne te tracasse pas à vouloir faire du neumorphisme sur cette étape, il s'agit d'un bonus pour l'avant-dernière étape (bonus) !

2 – Intéractivité

Tu l'auras sans doute remarqué, en haut à droite des maquettes se trouvent un élément bien précis : un bouton actif/inactif (switch).

Ce dernier concerne l'inversement des couleurs contrastantes sur la page, pour passer d'un mode sombre à un mode clair et inversement.

Comme cette étape ne concerne que Javascript, ton objectif sera de rendre fonctionnel le changement d'état sur ce switch.

Pas la peine de te focaliser de suite sur le changement du schéma de couleurs sur la page, seul le switch nous intéresse pour le moment !

On verra ce changement du schéma de couleurs à l'étape d'après !

3 – Interversion du mode sombre et clair

Allez, c'est le moment de mettre ça en place !

À l'aide des deux maquettes à ta disposition, tu dois faire en sorte que les deux thèmes (sombre et clair) soient interchangés en fonction de l'état du switch en haut à droite de la page.

4 - Responsive

Maintenant que nous avons une page fonctionnelle sur ordinateur, il serait très appréciable de la rendre disponible sur tous les autres périphériques, notamment les téléphones portables !

L'objectif est simple (sur le papier) : rendre la page consultable sur les périphériques de 360px de largeur.

Au passage, ne cherche pas les maquettes mobiles pour cette étape : il n'y en a pas. Tu as carte blanche sur le rendu final, même si ce dernier n'est pas "joli". Après tout nous sommes des développeurs, pas des UI designers.

Pas besoin de gérer une largeur inférieure à 360px. Désolé les utilisateurs des montres connectées !

5 – Serveur (bonus)

Tout d'abord, si tu es arrivé jusqu'ici tu peux prendre une pause et prendre le temps de t'applaudir !

Maintenant que la pause est passée, revenons aux choses sérieuses...

Et si on utilisait quelque chose pour gérer la logique back-end de notre application et servir les pages ? Ce serait top, non ?

Puisqu'il s'agit d'un bonus (et du premier !), on va augmenter les contraintes !

NodeJS

L'objectif de cette étape est de créer un petit serveur HTTP avec NodeJS. Son but est uniquement de nous retourner la page que tu as déjà intégrée plus tôt.

Tu as plusieurs solutions qui s'offrent à toi :

- Utiliser un framework (Express, Koa, Fastify, ...)
- Tout faire à la main (module natif "http" disponible dans NodeJS)

Tu devras également rajouter une page 404 si l'URL d'accès n'existe pas, pour éviter d'avoir une page non gérée sur le serveur !

PS : durant ta formation avec Simplon, tu verras PHP en lieu et place de NodeJS. Mais l'objectif est le même. Tu peux donc prendre cela comme une pré-initiation potentielle ;).

6 – Neumorphisme

Si tu ne l'as pas déjà fait, on va vouloir maintenant faire ressembler notre page à la maquette. Il y a une pratique UI que l'on appelle le "neumorphisme".

Cette pratique consiste à donner du relief aux différents éléments, au travers d'ombrages.

La propriété CSS principale : box-shadow

Mais si tu le souhaites, il y a des outils en ligne qui génèrent le code selon l'embossement souhaité et la couleur de fond !

7 – Détection de la préférence du mode sombre et mode clair

Promis, cette fois-ci c'est la dernière fois avec cette histoire de modes !

Étant donné qu'il est possible de définir notre préférence de rendu entre mode clair et mode sombre sur les navigateurs, il faudrait que notre page s'adapte automatiquement à son chargement.

Ainsi, si un utilisateur préfère le mode clair : on lui présente d'office le mode clair. Et inversement, si l'utilisateur préfère le mode sombre, on affiche la page dans le mode sombre.

8 – Le bonus des bonus

Tu es encore ici et tu en redemandes ?

Alors vas-y : tu as une carte blanche totale !

On sera ravi de voir ce que tu as pu imaginer, mais pense à bien documenter cette partie pour que l'on puisse vraiment comprendre ta réflexion.

Conclusion

Bon courage à toi, car il est vrai que ce test technique peut sembler intimidant !

N'oublie pas que l'objectif n'est pas de faire quelque chose de parfait, mais de nous démontrer comment tu appréhendes ces tâches et que nous, derrière, puissions t'apporter un support adapté à ton aisance en tant que développeur/développeuse !