**OPENCLASSROOMS : PARCOURS DEV. WEB. JUNIOR**

**SOUTENANCE PROJET 3 : JAVASCRIPT**

Mentor validateur : Gaëtan Herfray

\*PARTIE 1 : PRÉSENTATION DU PROJET (15 MINUTES)

INTRODUCTION :

Bonjour. Je vais vous présenter le travail que j’ai effectué pour la création d’un site de location de vélo pour la ville de Lyon.

Dans une première partie je vais vous présenter de façon générale l’utilisation et les éléments constituants le site en lui-même (organisation de la page d’accueil, l’espace signature et la réservation, la confirmation de la réservation).

Dans une seconde partie, je vous présenterai la structure et l’organisation du code du site : les différents fichiers, la création des objets, l’appel à la fonction Ajax et le canvas.

1. PRÉSENTATION GÉRÉNALE DU SITE :

1° ORGANISATION DE LA PAGE D’ACCUEIL :

-site en one page

-header avec titre et sous-titre du site

-slider pleine page avec 4 images : 1 page de bienvenue et description du fonctionnement du site en 3 étapes

-slider actif automatiquement, actif au click avec possibilité de mettre sur pause, d’avancer et de reculer, actif au clavier

-deuxième section avec une carte active et un espace renseignant les informations de la station sélectionnée en temps réel : markers cliquables pour obtenir les infos

-bouton ‘réserver’ pour procéder à la réservation d’un vélo (on voit cela plus en détail plus loin

2° ESPACE SIGNATURE ET RÉSERVATION :

-en cliquant sur le bouton ‘RÉSERVER’ : ouverture d’une fenêtre permettant de procéder à la réservation d’un vélo

-on trouve le résumé des informations de la station sélectionnée : nom et adresse

-deux champs doivent être remplis : nom et prénom

=> on note que si on l’utilisateur ne rentre pas des lettres ou n’écrit rien : la validation de la réservation n’est pas possible

-signature obligatoire dans le cadre ‘canvas’ sinon réservation impossible

-s’il manque des champs non renseignés, les champs sont mis en évidence pour indiquer à l’utilisateur qu’ils sont obligatoires

-on note dans la console que les informations sont enregistrées dans le local storage de manière à ce que l’utilisateur conserve la réservation s’il ferme son navigateur

-l’utilisateur peut alors soit effacer les données inscrites soit valider sa réservation : dans les deux cas la fenêtre se ferme

3° RÉSERVATION EN COURS :

-le footer rassemble les informations de la station dans laquelle l’utilisateur a effectué sa réservation et le temps restant pour la disponibilité du vélo soit 20 minutes

-le timer est également enregistré dans le session storage pour que l’utilisateur ne perde pas sa réservation s’il recharge sa page

-le timer est un décompte et dure 20 min

-à l’issue des 20 minutes un message indique que la réservation a expiré

-l’utilisateur peut également annuler la réservation et peut en faire une nouvelle, le timer repartira à 20 minutes

4° RESPONSIVE ET CONFORMITÉ DU SITE :

-le site est passé entre les mains de W3C-Validator pour la structure : quelques warning sur des éléments comme ‘type’ ou des balises ‘h1’ mais pas d’erreur majeure

-de même pour l’accessibilité sur avec achecker.ca : pas d’erreur majeure mais des balises pour les icones fontawesome qui ne semblent pas être les bonnes alors même qu’elles sont simplement des copier-coller du site. Je préfère ne pas les modifier

1. STRUCTURE ET ORGANISATION DU CODE :

1° ORGANISATION DES FICHIERS :

-fichier HTML organisant les différentes sections qui constituent le site

-fichier CSS pour la mise en page

-fichier controler.js pour l’appel des différentes fonctions qui animent les différents objets

-fichier utils.js rassemble les fonctions génériques qui interviennent dans les différents objets : pour initialiser l’appel Ajax générique qui permettra de récupérer l’ensemble des données de l’API JC Decaux grâce à l’objet mapbox.js, la fonction qui permet de vérifier que l’utilisateur a bien rempli les champs nécessaires à la validation de la réservation, la fonction ‘timer’ qui initialise le décompte de temps de la réservation

-fichiers slider.js, mapbox.js, resa.js et canvas.js constituent les objets qui animent le site

2° LES OBJETS :

-j’ai défini les objets principaux en fonction de leurs actions :

-le slider

-la carte et tout ce qui s’y rapporte

-la réservation

-le canvas qui permet la signature

-l’ensemble des appels de fonction de tous ses objets se fait via le fichier ‘controler’

-certains de ces objets ont un constructeur (fonction init) d’autres pas, en fonction de leur action

3° L’APPEL AJAX ET L’API JC DECAUX :

-le fichier nommé ‘utils.js’ rassemble les outils = les fonctions génériques à certaines fonctionnalités elles-mêmes actives dans des fonctions spécifiques de chaque objet

-on y trouve la fonction ‘ajaxGet’ qui permet ainsi via l’objet ‘myMap’ de récupérer les données en temps réel de l’API JC Decaux et d’intégrer les données à notre site

4° LE CANVAS ET LA SIGNATURE :

-lorsque l’utilisateur veut valider sa réservation, il doit signer dans le cadre prévu à cet effet : on a créé un objet ‘canvas’ qui permet de signer

-cet objet existe grâce au calcul de la position de la souris sur l’écran, de son mouvement et surtout du click qui permet l’écriture donc la signature

-le canvas est également actif au touché sur tablette et mobile pour répondre aux normes responsive

CONCLUSION :

Voilà l’ensemble de notre projet aboutit. Toutes les fonctionnalités requises sont là et ce site répond aux attentes de tout utilisateur qui souhaite effectuer sa réservation facilement en ligne.

J’espère ainsi avoir répondu à toutes vos attentes.

\*PARTIE 2 : RÉFLEXION SUR LE PROJET (8 – 10 MINUTES) SLIDE 1

-LE RÔLE DU DÉVELOPPEUR FRONT-END SLIDE 2

-ce projet permet de mettre en action toutes les qualités d’un bon développeur Front-end à savoir :

-comprendre et analyser un cahier des charges client et web = technique

-traduire les exigences d’un client et mettre en valeur ses différents objectifs pour valoriser son activité ou sa visibilité

-imaginer et réaliser la structure visuelle du site telle que l’utilisateur va le voir, donc se mettre dans la peau de l’utilisateur pour que le site soit le plus accessible, le plus clair et le plus ergonomique possible

-traduire cette structure visuelle dans un premier temps en langage HTML et CSS

-puis organiser son code pour créer les différents éléments actifs qui vont ‘animer’ le site avec le Javascript et la création d’objets

-être capable de travailler de façon claire en commentant son code pour permettre toutes les modifications et un travail en équipe si besoin

-faire appel à plusieurs outils techniques comme l’utilisation des API pour répondre aux attentes de l’utilisateur en temps réel = répondre à des problématiques d’application et pas seulement d’un simple site web

-penser en termes de développement d’appli mobile et pas seulement s’arrêter au site même

-LES DIFFICULTÉS RENCONTRÉES ET LEURS SOLUTIONS SLIDE 3

-ma principale difficulté a été d’intégrer la logique de construction en Javascript

* J’ai d’abord cherché toutes les documentations possibles (cours, tuto, exercices…) sur internet
* Je me suis tournée vers plusieurs communautés actives sur le net (Workplace, Facebook) pour trouver de l’aide et des explications
* J’ai beaucoup travaillé ! et n’ai pas hésité à reprendre systématiquement au début les bases et à approfondir par des recherches et des exercices en ligne
* J’ai trouvé des applications ludiques pour travailler et apprendre de façon ludique : merci à GrassHopper ! la petite bête qui m’a beaucoup aidé à comprendre et à CodeCombat ! le petit bonhomme qui court et qui saute

-organiser les éléments en objets, comprendre leur fonctionnement et l’intérêt de l’objet

* recherche de documentation
* exercices, beaucoup d’exercices,
* faire des dessins pour comprendre l’interaction
* écrire en pseudo-code pour ensuite traduire la construction de l’objet

-LES PERSPECTIVES D’AMÉLIORATION ET ÉVOLUTION DE L’APPLICATION SLIDE 4

-À TITRE PERSO : faire moins d’appels au DOM dans mes objets pour rendre mon site plus rapide, rationnaliser le code et le rendre plus optimal pour plus de clarté en vue de travailler avec une équipe pour faire évoluer le projet

-permettre à l’utilisateur de se créer un compte personnel qui enregistre toutes ses données perso (utilisation d’une API) et qui peut faire plus rapidement une résa sans avoir à signer (pas toujours pratique surtout sur un smartphone)

-envoyer un texto à l’utilisateur pour le prévenir quand il ne reste que 5 minutes par exemple de validité de sa résa avant qu’elle n’expire

-possibilité de sélectionner des stations préférées et des plages horaires favorites pour visualiser en un coup d’œil les dispos de vélos

-créer un compte premium / payant pour étendre la durée de la réservation à 40 minutes par exemple

\*PARTIE 3 : QUESTIONS – RÉPONSES AVEC L’ÉVALUATEUR (5 – 10 MINUTES)

\*FICHIER JSON :

Document qui rassemble des informations sous forme de tableau avec clé / valeur et que l’on peut alors récupérer pour utiliser des informations choisies.

\*FONCTION CALL-BACK

La fonction call-back est une fonction de rappel = une fonction qui est passée en paramètre à une autre fonction pour qu’elle en fasse usage => exécute du code dynamiquement

\*MÉTHODE AJAX :



Méthode ajaxGet générique

-Asynchronous Javascript and XML = requête HTTP à un serveur web qui renvoie du HTML

-Permet :

- de ne pas transmettre une page web complète à chaque action de l’utilisateur

- envoyer des requêtes http sans interrompre la navigation

- mettre à jour une partie seulement de la page

- diminution du temps de chargement

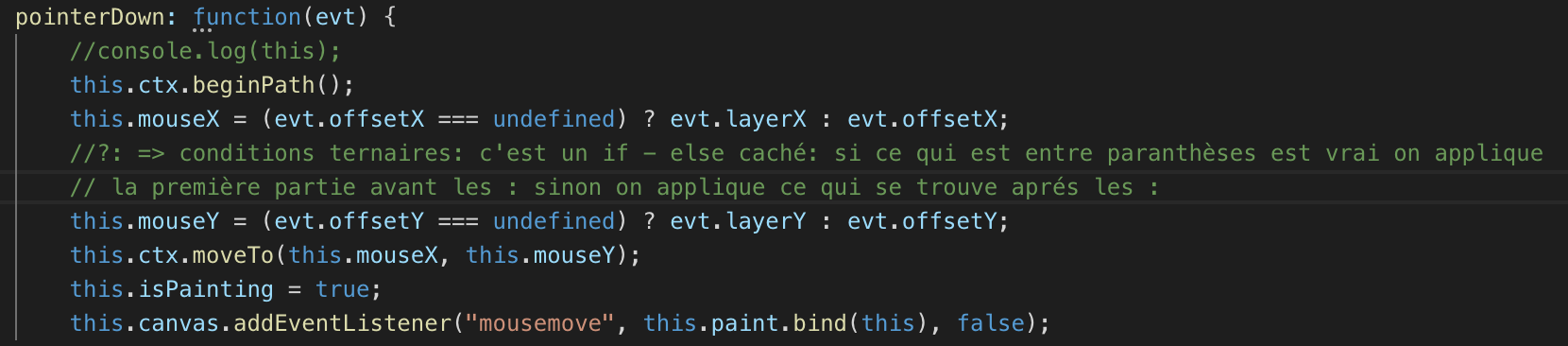
- nouvelles possibilités d’interaction avec l’utilisateur



Objet map qui utilise l’appel ajaxGet

\*CANVAS :

Surface de dessin fixe. On commence par définir le contexte d’écriture / dessin (méthode getContext()) pour ensuite définir les coordonnées du tracé et l’afficher (méthodes beginPath() et paint())



\*OBJETS JS :

La programmation orientée objet : on créé une sorte de moule pour généraliser la construction d’un élément clé de l’animation du site, sorte de moule à gaufre pour un objet type que l’on peut réutiliser pour la création d’autres éléments mais avec d’autres paramètres.

Ex : l’objet map est créé pour la ville de Lyon mais j’aurai pu me servir de cet objet pour créer un objet Paris par exemple sans avoir à réécrire le code.

\*BIND ET THIS / CONTEXTE :

This fait référence au contexte d’une fonction : lorsque l’on utilise une fonction dans une autre, il faut rappeler à quel contexte elle appartient.

La méthode bind() qui prend this comme paramètre, s’utilise en particulier quand le callback fait référence au this.



\*WEBSTORAGE :

2 interfaces, sessionStorage et localStorage qui permettent de mémoriser

sessionStorage : mémorise les données sur la durée d’une session de navigation ; portée limitée à la fenêtre de navigation ou de l’onglet actif = lors de la fermeture les données seront effacées

localStorage : mémorise les données sans limite de durée de vie, ne sont pas effacées lors de la fermeture de l’onglet ou du navigateur = portée plus large en utilisation sur plusieurs onglets actifs en même temps.

\*COOKIES ET RGPD :

Cookie : fichier généré par le serveur du site et installé sur l’ordinateur de l’utilisateur d’un site, permet d’enregistrer des données : navigation, mots de passe, analyse des pages vues, paramètres, connexion aux réseaux sociaux…

Rgpd : règlement général sur la protection des données. Encadre le traitement des données collectées de manière égalitaire en EU : renforcer les droits des personnes, résponsabiliser les acteurs traitant des données, crédibiliser la régulation grâce à la coopération entre autorités de protection des données.