Correction bug

Slider image blanc :

J’ai dans un premier temp consulter le « react dev tools » dans la partie « container » le slider afin de regarder le state et voir comment le slide se comporte au changement des images. Je constate que le slide comporte normalement 3 images mais que celui-ci va chercher une image avec un id 3 or dans un « tableau » avec 3 images, il y a un id 0, 1 et 2 donc il y a un souci sur l’appel d’un id 3.

Correction :

Pour corriger ce problème, je me suis rendu directement dans le fichier qui correspond au slide présent sur le site web et j’ai lu le code. Je sais que le problème vient de la longueur du tableau donc je me suis attardé sur ce point-là dans le code.

Je vois donc la ligne de code correspondant et vois qu’effectivement la longueur du tableau est proportionnel au nombre d’image dans le tableau, donc 3 mais nous avons que l’id 0,1 et2 donc j’applique un -1 sur « lenght » a la ligne 15 du fichier src > containers > slider > index.js

--------------------------------------------------------------------------------------------------------------

Point position image slider :

Je vois maintenant que les petits points qui servent à savoir sur quelle image est l’utilisateur ne fonctionne pas. En inspectant directement les points, je me rends compte que le checked est bloquer sur le dernier input radio.

Correction :

Le soucis était que le .map qui servais à généré les points du slider, était dans un autre .map donc à chaque fois qu’ont généré le slider ont généré les points.

Ordre slider :

Un problème au niveau du slider m’as été communiquer, les image sont en ordre croissant alors que le client les veut en ordre décroissant.

Correction :

Je me rends donc dans le fichier index.js qui correspond au slider, je regarde le code ou sont trié les images du carrousel et je vois que l’ordre de trie n’est pas correct. L’erreur viens du chevron « < » il était dans le mauvais sens donc mauvais sens de trie. Pour le corriger rien de plus simple, on inverse le chevron « > » et hop problème résolus !

---------------------------------------------------------------------------------------------------------------

Image footer dernier évènement :

Le souci ici c’est que quand je regarde avec react dev tool le composant EventCard il y a image.src et title qui sont undefined.

Correction :

Je me suis donc penché directement sur le code de celui-ci je vois que le code a l’air correct au premier abord je décide donc de regarder comment est utiliser le composant EventCard dans l’index.js de Home. Là je vois que des données sont transmise à EventCard via la variable last. Je décide donc de mettre un console.log juste en dessous de « const last » et surprise « last » est undefined. Je vois que « last est = usecontext » donc je regarde d’où est importer UseContext et je me rend dans le code. J’observe le code et là je m’aperçois que le code est bon que les données demandées sont correct mais que les valeurs partager sont data et error. Je ne connais pas la fonction useContext donc je me documente dessus et je m’aperçois en vue de ce que j’ai appris que le code qui compose de usecontext est correct mais que les valeurs donner sont data et error alors que moi dans le home > index.js je demande last. Donc cela ne va pas du tout ! je décide d’applique n’y plus n’y moins ce que j’ai appris dans ma documentation et la miracle ! je reçois des donnée sur la variable data de useContext !!! c’est bien mais je dois filtrer maintenant les données reçus afin de récupéré uniquement le « .cover » de l’objet le plus récent. J’utilise donc la fonction reduce( ) je crée mon filtre je modifie le code de imageSrc et title et la parfait ! le bug est résolus !

---------------------------------------------------------------------------------------------------------------

Les filtres non fonctionnels :

Les filtre de la section « Nos réalisations » ne fonctionne pas.

Correction :

J’ouvre react dev tool, je regarde ce qui compose les cards et je m’aperçois que c’est le composant EventList qui génère l’ensemble de la section, y compris les filtre. Je me rends donc immédiatement dans ce fichier et j’interprète ce code.

Le souci qu’il y a c’est que la logique de filtrage des événements ne prend pas en compte le type sélectionné. Le filtrage est uniquement basé sur l’index des évènements.   
Pour corriger cela, il faut filtrer les évènements en fonction du type sélectionné et ensuite appliquer la pagination .