**Unexpected Prop Types**

D’abord on voit should be trending. Donc je recherche dans le code à quel endroit se trouve le composant qui définit le trending. C’est le composant **Popularity**. Je remarque que pour que ça soit trending, il faut que la props popularity ait une valeur de **1**. Or pour le bug en question, on lui passe une chaine de caractère **“trending”.** J’ai donc remplacé cela par la valeur **1** pour réglé cette partie du bug.

Je remarque aussi que le rating n’est pas de la bonne valeur. Donc je remplace 0 par 3.5 et je règle ainsi la deuxième partie du bug.

On voit aussi que le stock est mis à **null** alors qu’on a besoin d’une valeur supérieure à 5 pour avoir la valeur In stock. Donc j’ai remplacé **null par 30**.

Le bug est attrapé !!

**Changing Props**

Premièrement, les boutons **like** et **unlike** ne marchent pas. On verra que le composant en question est **LikeButton**. A l’intérieur de se composant, c’est la valeur de la variable **likeValue** qui détermine si le bouton est liké ou non. Mais en regardant de plus près on remarque que sa valeur est changée manuellement. Cela peut poser problème parce que à chaque fois que la valeur change, le composant n’est pas **re-rendu** et le visuel n’est pas changé. J’ai donc fait en sorte que la gestion de la variable **likeValue** soit faite à travers un **useState**.

Le problème maintenant est que lorsqu’on appuie sur le bouton level down, les valeurs augmentent. C’est parce qu’on utilisait une props **attributes** pour afficher les valeurs. Et à l’intérieur on modifiait attributes avant de l’utiliser. Mais le fait est que à chaque rendu, bien que le level change, on a aussi les attributes qui se modifient. Or la logique voulais qu’on ajoute sur attributes des nombres pour permettre que le level puisse faire diminuer ou augmenter les valeurs. Pour résoudre le problème, j’ai créé un nouvel objet **newAttributes** que j’utilisais et modifiait à la place. Ce qui fait maintenant qu’à chaque re rendu, il reprend la valeur basique initiale **attributes** avant d’appliquer les changements en fonction du level.

**Passing Props**

Là ce sont les props qui n’était pas passée correctement. Au niveau du bouton like, les nouvelles props étaient passées avant celles par défaut, ce qui faisait qu’elles n’étaient pas prises en compte.

Au niveau du header, on faisait varier le level du heading en fonction du level choisi alors que celui-ci devait être constant et égal à 3.

Au niveau du summary, lorsque le composant PurchaseSummary a été utilisé, les bonnes props ne lui ont pas été passées.

**Unexpected State Changes**

Le composant ShySpider manquait la prop level que j’ai ajouté.

**Hook Side Effects**

Un des problèmes était le countdown. Il fallait nettoyer le setInterval dans le useEffect.

Il fallait aussi supprimer le useEffect avec onQuantityChange.

On devait aussi supprimer le deuxième argument du useEffect à l’intérieur de useTrackAnalytics

**Strict Mode**

Il fallait juste cliquer sur le bouton.

**Delayed Render**

Il fallait enlever le useEffect avec onQuantityChange à l’intérieur.

**Conditional Render**

Il fallait enlever le rendu conditionnel au niveau de VolumeDiscount parce que la condition demandait des calculs complexes qui ralentissaient le rendu.

**Excessive Render**

Il fallait enlever le useEffect qui changeait la quantité, il créait une boucle infinie.

**Deffered Render**

I suffisait d’enlever le delay sur le label.