Instructions générales

Coder pour du web est parfois difficile, mais coder pour une télévision ou des appareils mobiles exige des considérations de performance très pointues. Gardez donc toujours à l’esprit l’optimisation dans la réalisation des instructions et l’aspect multiplateforme. La majorité des points seront sur ces deux éléments.

## Points considérés

* Respect des instructions
* Performances
* Portabilité
* Qualité du code
* Gestion d’erreur
* Maintenabilité
* L’accessibilité

## Préparation

* Faites un clone local du repository à partir de GitHub:

git clone <https://github.com/LVLStudio/LVL.WebExam.git>

* Créez-vous une branche locale de travail nommée exam:

cd LVL.WebExam

git branch exam

git checkout exam

* Committez vos changements dans cette branche.
* Générez une patch nommée exam.patch contenant vos changements de la branche:

git format-patch master --stdout > exam.patch

* Finalement, envoyez-nous votre patch à job@lvlstudio.com pour que nous puissions constater votre magnifique travail! ;)

## Tâches

1. En dessous de 1024px de largeur, il est souhaité de cacher le menu par défaut pour que le reste du contenu puisse prendre toute la largeur.
   1. Le bouton de menu (actuellement caché) doit apparaitre
   2. Au clic sur le bouton, le panneau de menu doit alterner entre l’état ouvert et l’état fermé, dans une animation de glissement qui pousse le restant du contenu vers la droite.
2. Quelque chose ne va pas avec l’intégration des assets graphiques. Corrigez les boutons de menu et play/stop et l’icône de « extra » pour qu’ils s’affichent correctement (les hauteurs et largeurs en CSS sont correctement définies).
3. Le web service <http://examen.lvlnetworks.com/> retourne des blocs d’information JSON. Construire le module qui permettra de récupérer et d’utiliser ces données.
   1. Chaque appel à l’API retourne des items pour les 10 prochaines secondes et les valeurs startTimestamp et endTimestamp.
   2. La liste doit être tenue à jour, assurez-vous de répéter l’appel aux API en conséquence.
4. Utiliser les données récupérées à l’étape précédente pour remplir la liste d’éléments #items.
   1. Basez-vous sur l’élément .item pour remplir la liste
   2. Inclure les éléments au fur et à mesure que leurs valeurs timestamp est passée.
   3. Les nouveaux éléments doivent être ajoutés en haut de la liste.
   4. La liste ne doit pas contenir plus de 100 items à la fois. À ce point, les éléments à la fin doivent disparaître pour laisser la place aux nouveaux.
5. Au clic sur un item :
   1. Le background devient alors gris (#444444) et le texte blanc (#ffffff).
6. L’intégration comporte diverses erreurs d’accessibilité, corrigez-les!
7. L’application se veut être responsive, apportez d’autres améliorations si vous en voyez.