General Instructions

Coding for the web is sometimes difficult, but coding for television sets or mobile devices requires strict performance considerations. To that effect, please always keep two things in mind when carrying out the following instructions: optimisation, and the multi-platform nature of the product. The majority of the following points will require consideration of these two elements.

## To be considered

* Respect of instructions
* Performances
* Portability
* Code quality
* Error handling
* Maintainability
* Accessibility

## Preparation

* From GitHub, make a local clone of the repository:

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

* Create a local work branch named exam:

cd LVL.WebExam

git branch exam

git checkout exam

* Commit your changes in this branch.
* Generate a patch named exam.patch containing your branch changes:

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

* Finally, send us your patch at job@lvlstudio.com so that we can marvel at your work! ;)

## Tasks

1. When smaller than 1024px width, the menu should be hidden by default so that the content can take up the rest of the space.
   1. The menu button (currently hidden) has to appear.
   2. When the button is clicked, the menu panel must alternate between its open and closed states, in a sliding animation that pushes the rest of the content to the right.
2. Something is wrong with the integration of the graphical assets. Fix the Menu and Play/Stop buttons, as well as the “extra” icon, so that they display correctly (the height and width are correctly defined in CSS.)
3. The web service <http://examen.lvlnetworks.com/> returns JSON data. Build a module that will retrieve and use this data.
   1. Each call to the API returns items for the next 10 seconds, as well as the startTimestamp and endTimestamp values.
   2. The list must be kept up to date, ensure that you repeat the call to the API to keep the data flowing.
4. Use the data retrieved in the previous step to fill the #items list.
   1. Use the .item element to fill the list.
   2. Include the elements in order of their timestamp value.
   3. The new elements must be added at the top of the list.
   4. The list must not contain more than 100 items at a time. At that point, elements at the end of the list must disappear to leave room for new ones.
5. On clicking an item:
   1. The background turns grey (#444444) and the text turns white (#ffffff).
6. The integration includes some accessibility error and bad practices. Correct them.
7. The application must be fully responsive. Make additional fixes if required.