We need to create two pages for visualizing awards and badges.

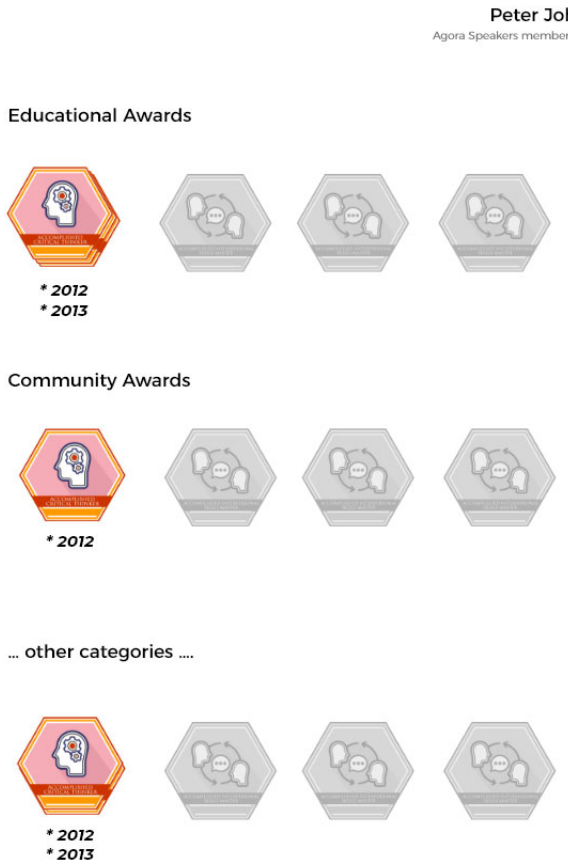
The first page will be hosted at `/pages/[user]/awards` and will display a general view of all the awards of a particular user, and the second page will be hosted at `/pages/[user]/award/[id]` and will display the detail for a particular award from the first page.

The values for the data to display in both pages will be retrieved via a REST GET call to a service endpoint that we will provide, which will return a convenient JSON object with all the information that needs to be displayed.

General information:

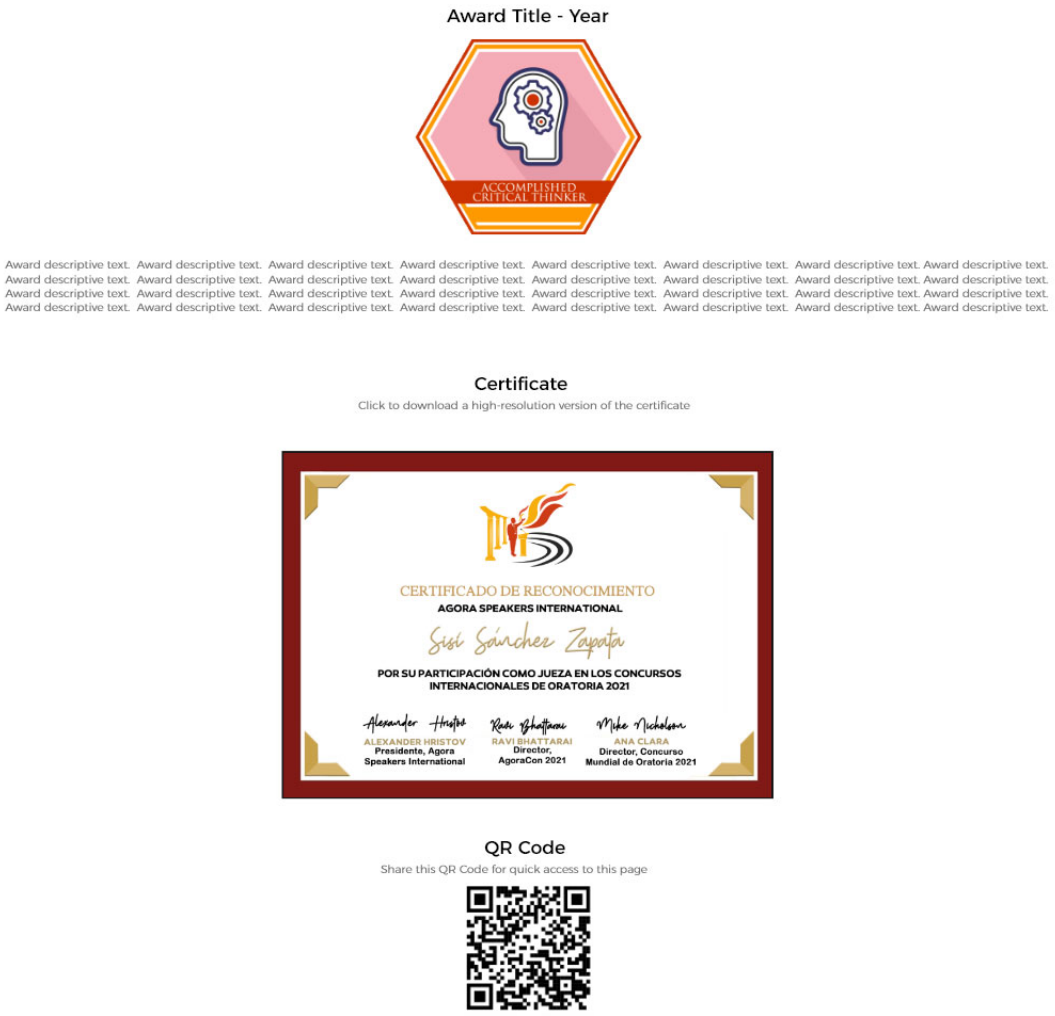
- The general design is provided, and no further design work is needed. We will provide a CSS file that lists the colors, styles and effects to use.
- All the graphics (badges, certificates, etc.) already exist and the REST services will provide URLs to their location.
- Must be mobile-friendly, but it's without obsessing over it.
- Fluid layout using react grid layout.
- Any icons must be from the Font Awesome iconset.
- The form must be I18N, using Next-i18Next. We will provide the message files for the supported languages in format the library requires.

General List (`/pages/[user]/awards`)



- Awards are classified in several different categories. Scroll must be vertical to be able to view them all
- Within each category, there will be awarded badges and badges that have not yet been achieved.
- Within each category, scrolling is horizontal.
- A badge may be awarded multiple times at different dates, in which case the image of the badge will be stacked as many times as it has been awarded, with an upper limit of 5.
- Clicking on an achieved badge will lead to the second page with the deail of that badge.
- All badges (achieved or unachieved) will have a tooltip popup with a short explanation for the award.

Badge Detail `/pages/[user]/award/[award-id]`



Notes:

- The QR-Code must be generated client-side and encode the current URL
- Clicking on the certificate will go (in a new browser window) to a different URL (that wll contain the high resolution PDF file of the certificate)
- This page can be accessed directly – do not assume that the caller is always the previous page.
- There should be no interaction between the two pages, other than the parameters passed as part of the dynamic route.

Requirements

- Framework: Next.js. We will provide an enclosing project.
- Language: Typescript
- A proper component grouping hierarchy must be created. All components must be located in `/components/members`.
- The developed components must be of functional type.
- Hooks should be used whenever necessary, instead of old-style approaches.
- No dragging in of whole UI libraries into the project.
- You can use any third-party components that are *single* UI components or provide some additional functionality as long as:
    - They're typescript-typed
    - They have a MIT or similar liberal license.
    - They're actively being developed or at least supported.
    - They're officially listed modules in the npm registry
    - They do not require ANY file-based configuration (such as a custom properties file or similar) or ANY kind of special processing (packing, compilation, etc.)

- You may add to the `package.json` that we provide, but you may not remove or change the versions of existing packages. Your development must be compatible with what we have.
- Please include the word “Persephone” in your quote to verify that you’ve read up to this far.
- Modifications of the workflow are not allowed – this includes insertion of tools into the toolchain such as transpilers, packagers, minifiers, CSS compilers, etc.
- There must be no typing errors nor warnings.
- The contract requires transmission in exclusivity of all the IP and associated rights of the work that you create under this assignment.
- Please note that we do not pay for partially-completed work. This ia an all-or-nothing task, and it’s either fully done (in which case the whole budgeted amount will be paid), or not done at all (in which case nothing will be paid and a refund of the escrowed amount will be requested).
- We will require *daily* progress updates, even if it’s simply saying “I didn’t have time to do anything”. But we need to hear from you *every business day*.
- When bidding, you must indicate the exact amount you're bidding for, as well as the time you will need for delivering the work. Make sure to set a realistic schedule : if the delivery is late by over 10 calendar days after the date you indicated, we will cancel the contract without pay.