Assignment

Hacked Hogwarts Student List

This is the final assignment in the Coding Visual Design theme. To make it easier to plan and execute, there are three "sub-assignments" (11C.04.01, 11C.04.02, and 11C.04.03), that gradually builds to this one. This description, however, includes every requirement, so you could skip the three sub-assignments, and make your own progressplan for this one.

The backstory (for those who prefer a reason behind the requirements):
You have been hired as the frontender for Hogwarts School of Witchcraft and Wizardry, to build a system to help the administrators handle student lists. As test-data you are given the students from the infamous class of 1991.

First, you are just asked to provide an interface to show the list, sort by firstname, lastname, or house, as well as filter by house. Also the interface must provide a "popup" window with detailed information about each student, including photo and house-crest and colors.

As you finish your assignment, you are asked to expand on the solution - now the administrator must be able to expel individual students, and see a list of expelled students. On a less dramatic note, two students from each house can be selected as prefects, and this should show in their "popup".

Then the customer experiences a massive shift in political view, by change of ministry and head-mistress, and you are bombarded with additional requirements. Now you must add "blood-status" to each student, basically implement racial profiling - something the original data doesn't have, so you must make the system "artificial intelligently" decide, based on additional data from the ministry about family-names. You are also asked to let the administrator appoint individual students to the inquisitorial squad. Only Slytherin-students, and pure-bloods from other houses, can join the inquisitorial squad.

You grow a bit tired of these new modifications, so you decide to hack the system. First you want to infiltrate the school, so you "inject" yourself in the list of students (in the house of your choice). You also make sure that you can't be expelled, and that the administrator will get a nasty message, if they try.

You mess up the pure-blood algorithm, by inserting code AFTER the original algorithm, that changes all non pure-blood to pure-blood, and all original pure-blood to randomly pure-, half- or muggle-blood.

You break the inquisitorial squad by adding a timer that removes any student added to the inquisitorial squad, after a few seconds. Making sure that the admin will get frustrated by thinking it works, then suddenly change.

Requirements

The list

The webpage must display a list of students. The list is read from http://petlatkea.dk/2019/hogwartsdata/students.json (or https if you prefer). You can download images from: http://petlatkea.dk/2019/hogwartsdata/images.zip, or use your own, but keep the same filenames!

This list of familynames: http://petlatkea.dk/2019/hogwartsdata/families.json can be used for determining blood-status.

Note that the JSON only delivers "fullname" for each student, but that includes first-name, optional middle-name, optional "nick-name", and last-name. The list is as it is - you need to handle picking and using the various name-parts.

Sorting and filtering

The user must be able to sort the list on first-names, last-names or houses. And filter the list to only show students from a selected house. In later iterations you'll also need to distinguish between expelled and non-expelled students.

Details about each student

The user must be able to select a student, and get a "popup" with details. Usually that might be a modal-window, but the design is entirely up to you. It could be a slide-in, an accordion, or a window opening.

The details must show:

- First name
- Middle name (if any)
- Nick name (if any)
- Last name
- Photo of student (if exists)
- House Crest and colors

In later iterations, also:

- if the student is a prefect or not
- Expelled - if expelled
- blood-status
- member of inquisitorial squad or not

Details about the list

The interface must show some details about the lists:

- Total number of students (not expelled)
- Number of students in each house
- Number of students expelled

Details in the list

The list **must** include first name, last name and house of each student.

You decide if you want to add additional info, like prefect-status, blood-status, inquisitorial squad member, image, house-colors, etc. Design the list to be easy for the administration to use, and to look good. Try to reduce the number of clicks required for each action, but also don't clutter the interface with a bunch of buttons and icons!

Expelling students

The user must be able to expel a student. You decide if the expelling should be done from the list, or from the "popup". Expelling removes a student from the list of students, and adds it to another list of expelled students. Once expelled, a student cannot return to the original list.

Note: Since the JSON is read from the server each time the page is reloaded, this will reset everything, including expelled status. That is a task for our back-end people, once they get hired.

Prefects

Only two students from each house can be selected prefects. Usually a boy and a girl.

The user must be able to make any student a prefect, and also revoke the prefect-status at any time. You must include some sort of system to prevent more than two prefects from each house. It is up to you how to design this. If the user must manually revoke one prefect before creating a new, or if there is simply a confirmation box for this. You can also decide if you want to continue the gender-specific prefects, or allow two boys or two girls.

Blood-status

The system must calculate blood-status for each student. This is an indication of whether the student is from a pure wizarding family, or from a half-wizard, half-muggle family, or just plain muggle.

The only way to determine this, is to use the list of family-names, and see if the student's lastname is from one of the pure-blooded families. Note that some previously pure-blooded families, has since been mixed with muggles, and are now half-blooded. Some names thus exists in both lists, and it is up to you to decide what is preferred: the "benefit of the doubt"-version of marking a half-blood as pure-blood, or the "everything not certain must go"-version of making a pure-blood as half-blood.

Inquisitorial Squad

The user must be able to appoint students to the inquisitorial squad, and remove them again. Any number of students can be appointed, but only pure-blood or students from Slytherin (should any non-pure-blood be in that house).

Hacking

You must inject yourself, with your own name, into the list of students. If the user tries to expell you, there should be a warning of some sorts.

Randomly modify the blood-status of students. Make sure that all students originally designated non pure-blood, will always show as pure-blood. Let all pure-blood students get completely random blood-status. Preferably do the modification on every redisplay of the list.

Adding a student to the inquisitorial squad should only work for a limited time, before the student is automatically removed again.

Visual design

You are free to design the web-page as you like, with the few requirements about images, crests and housecolors.

However, make it as visually interessting an experience as possibly. Don't just have a table of student-names, use your design-skills to make something that looks cool.

Add animations and visual feedback. When expelling a student, don't just reload the list without that student, but add an animation that removes the student visually from the list, and otherwise keep the list as is.

When preventing the user from expelling you, be as creative as possible. Think about bad hacker-movies from the 90s! Go crazy, use video and sounds if needed!

Give visual feedback when revoking inquisitorial squad membership from the timer. Make sure that the user sees that the status has changed! You want the hacking to be obvious.

Think about what to include in the list visually. Prefect-, blood, inquisitorial-squad status? Could you use icons? Make it visually exiting!

You can "steal" graphics and icons from the web for this assignment, it isn't a requirement that you draw everything yourself.

Code design

You must use pure vanilla JavaScript (with CSS and HTML) for your solution. No frameworks or plugins.

Your code must run in "strict" mode, and you must use ES6 JavaScript. That means let and const rather than var, and object literals, rather than .prototype definitions.

Write all the code yourself. If you need to use a part you have found on the web, then black-box it as much as possible, and remember to identify where you got it. That means put it in a function, or have comments before and after the lines. Put the URL in the comment, with a short description of what the code is supposed to do.

Use as many functions as possible. Many smaller functions rather than a few large. Think separation of concerns, as well as split between model, view and controller.

The code must use at least one object, a Student-object, with all the properties of a student. Create your own object, when reading the JSON-file, and calculate the properties from that file. Also let some properties have default values, if not found in the file.

Process

It is recommended that you split your workload into three major "sub-assignments":

11C.04.01 - filtering, sorting and display details

11C.04.02 - expelling

11C.04.03 - blood-status and inquisitorial squad

11C.04.04 - hacking, injecting and vandalism

For each one you should create:

- activity diagram
- pseudocode or sketched call-graphs

but only for the additional features you are adding in this version. E.g. you would create a complete activity diagram for sorting, filtering, and displaying details for the first version, but only for expelling in the second version.

It might be easier for you to have one large activity diagram for the entire app, but don't overwhelm yourself with designing everything up front!

Documentation

The assignment must be handed in as a report with this information:

- Wireframe for the visual design
- Activity diagram (or multiple diagrams for each phase)
- A list or drawing of the various parts of the application
- A can/can't list of each "component" or activity, with a note on whether you know how to do it or not (at the beginning of the design)
- Pseudocode for major parts, ie. sorting, filtering, expelling, blood-status, and inquisitorial squad with hacked timer. (Parts can be as call-graph sketches)
- Complete call graph for the finished code.
- A list of Student-object properties, e.g. firstName, lastName, imageName and so on.

Collect all documentation in one pdf, make sure it is readable on screen as well as (theoretically) print-out.

The frontpage must include

- Your name
- A link to the final solution (written out, as well as clickable)
- A screenshot of the final solution
- A link to your github repository (written out, as well as clickable)

Hand-in details

Important: This is a **mandatory submission**, which means that you **must** hand it in, to be eligble for the exam. If you miss the deadline, you will get a new wiseflow-invite before the start of the exam-project, where you will have to re-hand-in.

Where
Hand-in to **Wiseflow**, in the flow: Hacked Hogwarts Student List

How
A pdf as described under documentation.

The pdf must have a link to your working solution, as well as to your github repository

When
Before the **Wiseflow** deadline (Sunday the 22nd of September, at 23:59)

Naming
Name the pdf: *your-name*_hacked_hogwarts.pdf

Where *your-name* is replaced by your actual name.

Feedback
The assignment will be given an approved / not-approved in wiseflow.

If not-approved, you will have to re-submit before the exam. You will get a new invite for Wiseflow should that be the case.

The deadline has passed.