Web, Mobile and Security

Lab: Open Movie Database continued (JS Code Quality)

1a. Continuing from your own source

If you've completed the Open Movie Database exercise from week 2, use your own codebase as a starting point. This will be a fair challenge to revisit past sources.

1b. Starting from example source

If you didn't manage to get that far, or you simply want a clean slate, we have an example source to start from.

2. Implement the 7 elements to improve oyour JavaScript maintainability

Implement each of the following principles of code maintainability, as seen in the lecture session.

1. Adhering to coding conventions
2. Limited and clever usage of global variables
3. Error handling
4. HTML templates
5. Leveraging HTML data attributes and some DOM best practices
6. Cognitive clarity
7. Legibility through structure

Try not to follow the example from the morning lecture too closely, but apply it as you see fit with **your** components, namings, ...

3. Have it reviewed

Choose another student with whom you will switch code bases. The aim is to review each other's code, and add any remarks via gitlab.

In the example below **reviewee* is the student whose code is being reviewed, the **reviewer** is the one doing the review.

CAUTION: this example expects you to have completed the mandatory `git` leho course to a certain degree as well as the session on 'Gitlab' by Mr. Van Eekchout.

1. **Reviewee**: add permissions to see your repo on gitlab
   Project information > Members
   They will require **Developer** access

2. **Reviewer**: Create an issue titled "Project code review"

3. **Reviewer**: Create a merge request that goes with it

4. **Reviewer**: Clone the repository and make sure you switch to the branch that has the new merge request active

```
git checkout --track origin/NAME_OF_BRANCH
```

1. **Reviewer**: Make any modifications to the code you think should still happen. Remember: atomic commits!!
2. **Reviewer**: Push your changes
3. **Reviewer**: Go back to gitlab and mark the merge request as "ready". Tell your friend
4. **Reviewee**: Review changes, learn from them and either modify (clone the branch) or accept and merge into main
5. **Reviewee**: If you want to continue to work on the repo locally, you'll need to **pull changes** before moving on

4. Expand functionality

Now that your code has been reviewed, the project is ready for expansion.

Add a **'Watchlist'** functionality to your project. This consists of:

- Displaying a button **add to watchlist** with every movie. Think about where you'll add this, the list or the detail page. Both have pros and cons
- When clicking the button, the **movie is added to an in-memory array**, containing the user's watchlist
- Displaying a view watchlist, which shows the watchlist when clicked.

Further expansions

- Add the watchlist to local storage. Think carefully about what this does for your code. Should you add another component? Do any others need modifications? Remember: if you need make large changes in too many artifacts, they are too large.

- SE only: Think about the watchlist CSS. does it make sense to add it to the large CSS file containing the main layout? Why (not)?