01UDFOV/01TXYOV – WEB APPLICATIONS I

JAVASCRIPT IN THE BROWSER

During this third lab, you will add some "dynamic" behavior, through JavaScript and the DOM manipulation, to the user interface realized in the previous lab.

EXERCISE 1 - POPULATE THE TASK LIST

Starting from the outcome of the previous lab, replace the hard-coded tasks present in the "main content" (right area of the page) with the dynamic generation of a list of tasks starting from a data structure containing task information. Currently, the initial content of the data structure should be predefined in the JavaScript code. By changing the content of the JavaScript data structure and reloading the page, the displayed list of tasks will update accordingly.

The JavaScript array will contain the tasks with all their properties (i.e., *id, description*, *important*, *deadline*, *private*), as defined in the previous Labs. Consider the *important* and *private* properties as boolean *flags*.

Similarly, the list in the webpage will maintain the same appearance and functionality as in Lab 2 (important tasks have a red text, etc.).

The data structure must be contained in a *separate* .js file, properly included in the HTML file.

Hints

1. *Continue to use Bootstrap 4.6 classes and components to style the webpage, if needed: https://getbootstrap.com/docs/4.6/getting-started/introduction/*
2. *To dynamically generate the list of tasks from the JavaScript array, use the methods for DOM manipulation seen during the lecture (e.g., getElementById, createElement, …)*
3. *You can use the following <script> tag to import and use the day.js library:*
   <script defer src="https://unpkg.com/dayjs@1.8.21/dayjs.min.js"></script>
4. *You can use the solution available for Lab 2 as a starting point, if you prefer: https://github.com/polito-WA1-AW1-2021/lab2-html-css*
5. *You can use the constructor functions developed during Lab 1 to model Task and TaskList objects: https://github.com/polito-WA1-AW1-2021/lab1-node/blob/master/l01-e01.js*

EXERCISE 2 - FILTERS!

Make the "filters" work, by extending the previous exercise. In particular, you should enable the actions of the following *filters*, present in the left sidebar of the HTML page:

- **All**, to display *all* the tasks (as in Exercise 1);
- **Important**, to display tasks marked as *important*, only;
- **Today**, to show tasks whose deadline is today (e.g., *March 29, 2021*);
- **Next 7 Days**, to show tasks whose deadline is between tomorrow and the next 7 days (e.g., *Saturday 4th of April to Friday 10th of April*, inclusive);
- **Private**, to display tasks marked as *private*, only.

One and only one filter may be active at any time. The default filter is ‘All’.

All these actions will update the task list in the same “main content” area populated in the first exercise, i.e., without creating a new page or a new area. In other words, this will create different *views* of the same task list without modifying the content of the JavaScript data structure. Add some suitable tasks to the data structure created in Exercise 1 to test all the filters. **Beware:** filters should be mutually exclusive, i.e., when a filter is selected, all the others have to be deselected.