Michał Guspiel & Matthew Simpson
25.2.22

Dynamic Web Applications with JavaScript – Project One

Introduction

For the first project we were allowed to work in pairs and given a choice between the following web sites.

- To-do List
- Smart form
- Free choice (to be discussed with the teacher)

Michał Guspiel and I decided to work in a pair, and we agreed that we would build a to-do list.

Requirements

The requirements for the project are as follows: -

- The items entered in the field are assembled into a list on the page by pressing the button.
- The content of the input fields must be checked for blank as well as incorrect (e.g., too short) content
- Incorrect content in the input field causes an error message as well as an incorrect field highlighting (e.g., red border).
- It must be possible to remove elements from the list and mark the task as done
- The application CSS and JavaScript must be specified in an external CSS file
- The information in the list is stored in a browser, e.g., local storage

If we had time, we were also allowed to add functionality.

Michał Guspiel & Matthew Simpson
25.2.22

Did we meet all requirements?

In short, yes. All requirements were met.

- All items were assembled into a list of sorts. We decided to go with a different approach to the typical to-do list app and make each item its own note and they were assembled like post-it notes on a whiteboard.
- The title and description of the notes were checked for content and we added a little animation to display that whichever field was left empty, had to be filled.
- Incorrect content was too vague and difficult to discern what it actually meant. The end user knows what they want to add to each note so who is to say what is incorrect or not?
- All tasks can be marked as done without removing from the website. Tasks can then be removed either individually or all at once.
- All CSS and JavaScript comes from external CSS and JS files respectively.
- Local storage is used to store notes.

Any additional functionality?

Here is a list of the additional functions we added to the site.

- Dropdown menu added to select task priority level
- Checkboxes used to toggle between completed and incomplete tasks.

In addition to HTML, CSS, and JavaScript we used Bootstrap 5 to help style the site and improve the UI. In accordance with the high-level requirements, we used only vanilla JavaScript. We also used DOM scripting and forms.

The source code is hosted on GitHub here and is published to the web using GitHub Pages here. The teacher originally requested we publish on Netlify but also agreed that GitHub Pages is sufficient to fulfil the online publishing requirement.

How do we think we performed?

As a team we are both confident enough in our work and satisfied with the final product to warrant a high enough grade we can both be happy with.