Lab assignment 3

TI1506 2016/17; ti1506-students.slack.com

Introduction

In the first part of this assignment, you will you will write node.js code and use Ajax/JSON to enable the client and server to communicate with each other. In the second part of this assignment you will complete the first prototype of your Web application, by adding CSS.

Deadline: when this assignment is due depends on the cluster your team is in. Every team is assessed bi-weekly. Make sure you know your cluster and what this means for your lab deadlines. All of this information is available in the course syllabus (on blackboard).

In order to pass the assessment, you have to make a valid attempt at working through the lab assignment.

1. Node.js

So far your application can handle basic interactions, through client-side JavaScript. Lets now implement a first node.js script; if you don't know where to start look at Chapters 5 & 6 of the Web course book and Example 6 of the node.js lecture's code examples[1].

Your script should be able to do the following:

- Keep a list of the todos **in memory** on the server.
- Allow the client (the browser) to **retrieve the todos** from the server; use the **JSON** format for this task.
- Allow the client to **add** a todo to the server.
- Allow the client to **update** a todo on the server.
- Allow the client to **delete** a todo from the server.

2. Ajax

Use Ajax to allow the dynamic updating of the todo page in the browser (i.e. without reloading of the complete page). The lecture's code *Example 6* and book chapters 5 & 6 of the Web course book will help you again if you are stuck.

Note: Example 6 also contains a hint of how to enable repeated checking of the server (i.e. every X seconds the todo page polls the server for the todos). This can easily be achieved by wrapping this line into a timer function:

[1] https://github.com/chauff/TI1506-node.js

```
setInterval(function () {
    console.log("Fetching the todo list from the server.");
    $.getJSON("todos", addTodosToList);
}, 2000);
```

This piece of code will retrieve a list of todos from the server every 2 seconds (it is your job to check whether those todos differ from what is already there).

---

3. CSS

Now that we (finally) covered CSS, you will continue to work on your two HTML documents (the splash screen and todo page) and style them with CSS. To ensure that your CSS is of high quality, you can make use of CSS Lint: http://csslint.net/.

Choose three browsers (e.g. the latest versions of Brave, Google Chrome and Microsoft Edge) on which to test your Web application - the application should look and run in the same manner across those three browsers.

You are not allowed to use external libraries or preprocessing tools. Your application should have a modern look and feel (i.e. use sufficient CSS styling).

3.1. Splash screen

First, work on your splash screen and style the page with CSS according to your design. To ensure that every student learns the basics of CSS, we provide a list of **must-have** CSS properties. Your code must include at least one instance of each of the following:

- Pseudo-classes :hover :active
- Pseudo-elements: ::after ::before
- Box model: margin, padding, border
- At least two different position attributes, e.g. position:relative and position:absolute.
- At least one CSS animation

You are of course welcome to use more CSS properties to style your splash screen.

Once you have completed your CSS implementation, make a **screenshot** and add your implemented design to **your Blackboard discussion forum** (where you already posted your initial design). Does your implementation deviate significantly from your initial design? Write a **paragraph** comparing the two.

3.2. Todo page

Finally, work on your todo list page and implement your design in CSS. The look of the todo list page should be coherent with the splash screen. You will be able to reuse some of the CSS written for the splash screen - make sure that you are efficient and do not duplicate existing CSS code if possible.

Your design must visually distinguish between:

- todo list items that are **overdue**,
- todo list items that have **successfully** been tackled, and,
- todo list items of different **importance** rating.

You are of course welcome to use more CSS properties to style your splash screen. Include **at least one CSS animation**.

Once you have completed your CSS implementation, make a **screenshot** and add your implemented design to **your Blackboard discussion forum** (where you already posted your initial design). Does your implementation deviate significantly from your initial design? Write a **paragraph** comparing the two.