Session 5

Before you do anything at all, please read all of these instructions. Failure to do so will make it  
hard to answer two of the questions in a precise, honest manner.

For these tasks, you may freely choose any of the five DBMS from the previous session or another relational DBMS of your choice.

In addition to setting up a DBSM, also set up a web server of your choice (a simple NodeJS like  
in the example code is fine).

Then, replicate and extend the functionality of the example code of the session to create a HTML form (of your own design) that interacts with a database (also of your own design) and displays (some of) its contents over HTTP. You are free to use any additional NodeJS modules you like in building your version. You are also free to substitute NodeJS with some other technology you already master or would like to learn more about. Please discuss all such substitutions in your responses.

You can either keep it simple to learn at a steady pace or add complexity to challenge yourself.

This is a good moment to apply what you have learned in terms of front-end skills to add CSS  
and JavaScript instead of just rolling with static HTML like in the example code.

Please include code snippets and screenshots in your responses. Remember to properly cite all  
sources you have consulted; it would be great to also mention those that you checked but did  
not end up relying on.

1. Before starting to build the web service, estimate how many hours of work in total you  
   think this assignment presents to you. Which parts do you expect to be time consuming  
   and why?
2. Which operating system (and which version) did you choose for these tasks and why?  
   Are you using it virtualized, on a remote machine, or locally installed?
3. Which DBMS did you choose and why?
4. Which kind of a web server (NodeJS or something else) did you set up and on which  
   port? Why did you make those choices? Did you try any other ports first that turned out  
   to be unavailable? What errors did you encounter, if any?
5. What fields does your form include? How do they map with the table(s) of your database?
6. Do you offer any filtering options to view just some of the database entries? When does  
   it make more sense to filter at the back end upon generating the response instead of  
   sending all of it to the front end and using CSS/HTML/JavaScript to choose what to  
   view and how?
7. Are you using any validation of the user input at the front end or at the back end?  
   Should you maybe use more validation in a real-world setting than you do now?
8. Read about SQL injection attacks, unless you are already familiar with the concept. Is  
   what you built vulnerable to that? What could you do to protect it better? Test and  
   discuss.
9. How well did the time estimate you made for the first question hold? Were any tasks  
   harder or easier than you expected?

Sources:

<https://medium.com/swlh/read-html-form-data-using-get-and-post-method-in-node-js-8d2c7880adbf>

<https://www.youtube.com/watch?v=YYEC7ydDj4k>

<https://www.youtube.com/watch?v=4fWWn2Pe2Mk>