Homework #4
Points: 20
Deadline: 11 December 2021, 23:59

The main aim of this homework is to assess your capabilities concerning the use of **Node.js, EJS, and a PostgreSQL database** to create a realistic web application.

You can use your previous homework (Figure 1) as a baseline for this homework. However, you are required to develop a **website** that has the structure shown in Figure 2.

![Figure 1 index.html from the previous homework](image)
Tasks

1- Your server should be able to serve at least four different “pages/views”:
   a- **posts.ejs** that should query the **database** and list all available posts within the **posts.ejs** page, if there are no posts within the database, the page should show a message “you did not post anything”. Each post should have a title, a body, and a like button as well as the number of likes. Clicking on the like button should increase the number of likes by one only once, i.e., after clicking on the like button, it should become inactive. The structure of **posts.ejs** should look like Figure 3, pressing on any of the posts should redirect you to **singlepost.ejs**, which will present the post you have clicked on. (5 points).

![Figure 3 posts.ejs structure](image-url)
b- *singlepost.ejs* should present a single post, the post content should be fetched from the database based on its **id**. Each post should be accompanied with a **delete button** that if you click on, the post should be **deleted from the database**. The structure of *singlepost.ejs* should look like Figure 4 (3 points).

![Figure 4 singlepost.ejs structure]


c- *addnewpost.ejs* should include a **form** that takes at least post title, post body, and any other information you think is required for your post. Pressing on the add new post button should insert the new post into your database. The structure of *addnewpost.ejs* should look like Figure 4 (3 points).

![Figure 4 addnewpost.ejs structure]

d- *404.ejs* should appear when we request any page that does not exist within our website, i.e., any page that is not **posts, addnewpost, or singlepost**. (1 point).
2- All previously mentioned 4 pages share content that can be transformed into **partials** (partial templates). Check the shared content among them, and try to create all possible partials (2 points).

3- Style each of the pages relying on CSS, the pages should look close enough to the style of your previous project (Figure 1) (2 points).

4- None of your pages will output the required content without a database. Based on the description of the content of the pages, create a database that contains the required table(s), and connect it to your server to guarantee that all your pages/views will work as expected (4 points).

**Rules for homework submission and discussion**

1. **Through Moodle**, submit a **text file (*.txt)** that contains your Name(s), student ID(s), and a **valid and accessible link to the repository** that contains your “project”. Due to the nature of this homework, do not worry about sharing your database, i.e., you do not need to share it. You’ll present your whole project during the discussion session.

2. **If the link to your repository is not accessible or valid for any reason, you might not be allowed to discuss your homework or at least you will lose 5 points.**

3. **All team members should attend the discussion** of their homework; you **will not be allowed to discuss** if your team is **not complete**. If you already know that your team will not be complete because one or more of the members cannot attend due to another commitment, contact me as soon as possible and we can find a solution.

4. You have to submit your homework by the defined deadline, and **you will lose 0.5 point for each hour of delay.**

*The previous rules will be strictly enforced and there will be no exceptions*