01UDFOV/01TXYOV – WEB APPLICATIONS I

BIGLAB 2: FULL STACK TODO LIST

During the four weeks of the second BigLab, you will develop the back-end for your web-based task manager using Node + Express. In addition, you will improve the front-end of the task manager accordingly and connect the front-end to the back-end. As for the previous BigLab, to create your repository, you must login to GitHub Classroom and accept the assignment (if needed, join your group). For more details, please have a look at the GitHub Classroom instructions. Here you can find the links for the BigLab2 repository on GitHub Classroom:

- Web Applications I: https://classroom.github.com/g/NFnVVlQH
- Applicazioni Web I [A-L]: https://classroom.github.com/g/hZPQRYna
- Applicazioni Web I [M-Z]: https://classroom.github.com/g/1mhnogjm

To better keep track of your progress, we suggest you work incrementally "week-by-week", e.g., by creating, inside your repository, a branch for each week of the BigLab.

WHAT ARE WE BUILDING IN THESE WEEKS?

a) In the first week, we will create a **basic back-end for the task manager**. To do so, we will use the Express framework. The back-end has to implement a series of **APIs** to support the main features of the web-based task manager we developed in BigLab1: **create**, **read**, **update** and **delete** the tasks. The data will be persistently stored in an **SQLite database**.
b) In the second week, we will **update the front-end** of the web-based task manager (i.e., the final result of the BigLab1) to use some of the **APIs** designed in the previous week. In particular, we will **get the tasks** to be displayed from the server, and we will **save new tasks** on the server-side database.
c) In the third week, we will continue to **update the front-end** of the web-based task manager to **use all the APIs** designed in the first part of this BigLab. Specifically, we will use the APIs for filtering tasks, and for deleting and updating them.
d) In the fourth and last week, we will add the possibility of **having multiple users** to both our back-end and front-end applications, enabling them to **authenticate** (login and logout functionalities) and manage their own tasks. Task list page **access will then be refused** to non-authenticated users.

EVALUATION CRITERIA & DEADLINES

The points received for your work are added to the final exam score to each member of the team.
We will follow these evaluation criteria for evaluating your submission:

- The team members will receive 1 point if the submitted application is *complete*, i.e., it successfully implements *all functionalities of the 4 weeks*, i.e., points a), b), c), and d).
- The team members will receive 0.5 points if the submitted React + Express application is *partially complete*, i.e., it successfully implements the functionalities of *at least 2 weeks* (e.g., the team implemented only points a and b).

- The team members will receive 0 points otherwise.

The assignment must be submitted in the **master/main branch** before **Sunday, June 6 at 23:59 CEST** (see the the GitHub Classroom instructions for the details on the submission procedure). Remember that the last commit of your assignment must be **tagged** with "*final*".

If your **master/main branch is empty**, or you did not push a commit tagged with **final**, we assume that you decided **not to submit** the BigLab for evaluation.

The final repository structure **must** keep the client and server sub-directories of the provided template; thus, the project will need to execute correctly with the following commands:

i) for the back-end: "cd server; nodemon server.js;"
ii) for the front-end: "cd client; npm start;".