Web Applications I – Exam # 2 (deadline 2020-07-12 at 23:59)

"Kanban Board"

FINAL VERSION – Additions and corrections are shown in red

Design and implement a web application to create and handle Kanban boards, i.e., card-based boards to organize and prioritize some projects (for instance, steps towards the completion of this project, organizing your next trip – see https://en.wikipedia.org/wiki/Kanban_board).

A Kanban **board** is a single page with multiple columns. Each board has a name, an owner, and may be shared with other users. Each **column** in the board has a *title* and zero or more **cards** in it. Columns and cards can be dynamically created, edited, and deleted. A board must have at least 2 columns. A column may have any number of cards (including 0). Each card has some properties:

- a title (mandatory)
- a description (mandatory)
- any interesting links (a list of 0 or more URLs)
- a deadline (optional).

Such properties may be inserted when the user creates the card, and may be modified later, when the user edits the card.

Cards can be moved within and between columns through a set of 4 buttons present in each card (up/down within the column, and left/right to adjacent columns). Cards can be *archived*, i.e., still owned by the column, but moved at the bottom of the column. Archived cards show just the title and the description and offer just one available action: unarchive (that will put the card back in the upper part of the column).

The web application allows authenticated users to manage their own boards, which are shown in the main page (after the login), and to create new boards. The main page lists all the boards (owned by or shared with the user), with their name, the number of total cards and the number of expired cards (cards that have a deadline, and it is passed).

The user can perform various operations on the board, on its columns, and on its cards (see Table 1). After each modification (Table 1), the application should immediately save the change on the server's database.

The creator of a board can share the board to any other ~~authenticated~~ existing user. When a board is shared with a user, he/she will be able to do exactly the same operations on the board as the original owner (including deleting it). A specific page of the application must list two sections: "Boards I shared" and "Boards shared with me".

Handling concurrent modifications is not required. Assume that a given board is open by only one user at a time.

The web application, when visited by unauthenticated users, should display a read-only (static) "example board", to incentivize such visitors to login and to start creating their own boards.

Table 1: Actions available in the Kanban board

Operation	From…
Create a new board	… the main page
Delete a board	… the main page or the shared boards page
Share a board with another user	… within the board
Create, rename, or delete a column	… within the board
Create a card	… within a column
Edit a card	… within a card (except archived ones)
Move (up/down/left/right), delete, archive card	… within a card (except archived ones)
Unarchive a card	… within an archived card

Project requirements

- The application architecture and the source code should be developed by adopting the best practices in software design, in particular for single page applications using React and REST.
- The project must be realized as a React Application, interacting with a REST API implemented in Node+Express. The database should be stored in an SQLite file.
- The communication between client and server must follow the "React Development Proxy" pattern and React must run in development mode.
- The top-level directory must have a README.md file and contain two sub-directories (client and server). The project must be run with the following commands: "cd server; nodemon server.js" and "cd client; npm start". A skeleton of the project directories will be provided.
- The whole project must be submitted on GitHub, in the repository created by GitHub Classroom.
- The project must not include the node_modules directories. They shall be re-populated by running "npm install", immediately after "git clone".
- The project may use popular and commonly adopted libraries (such as moment.js, react-bootstrap, etc.), if applicable and useful.
- User login and access should be protected with a JWT token, stored in an http-only cookie. No additional protection is required.
- The project data-base must be defined by the student and must be preloaded with at least 5 example users (the registration procedure is not required), and at least 4 boards with some columns and cards each.

Contents of README.md

The README.md file must contain the following information (a template is available in the project skeleton – in general each explanation should be no longer than 1-2 lines):

1. A list of the React Application Routes, with a short description of each route's purpose
2. A list of REST APIs offered by the server, with a short description of the parameters and the exchanged entities
3. A list of the database tables, with their purpose
4. A list of the main React components adopted in the application.
5. A screenshot of the Kanban board editing page (embedding a picture stored in the repository).
6. The usernames and passwords of the 5 test users. Identify one user, who shared at least one board and has at least one board shared with him.