Project work

Last update 7.1.2025

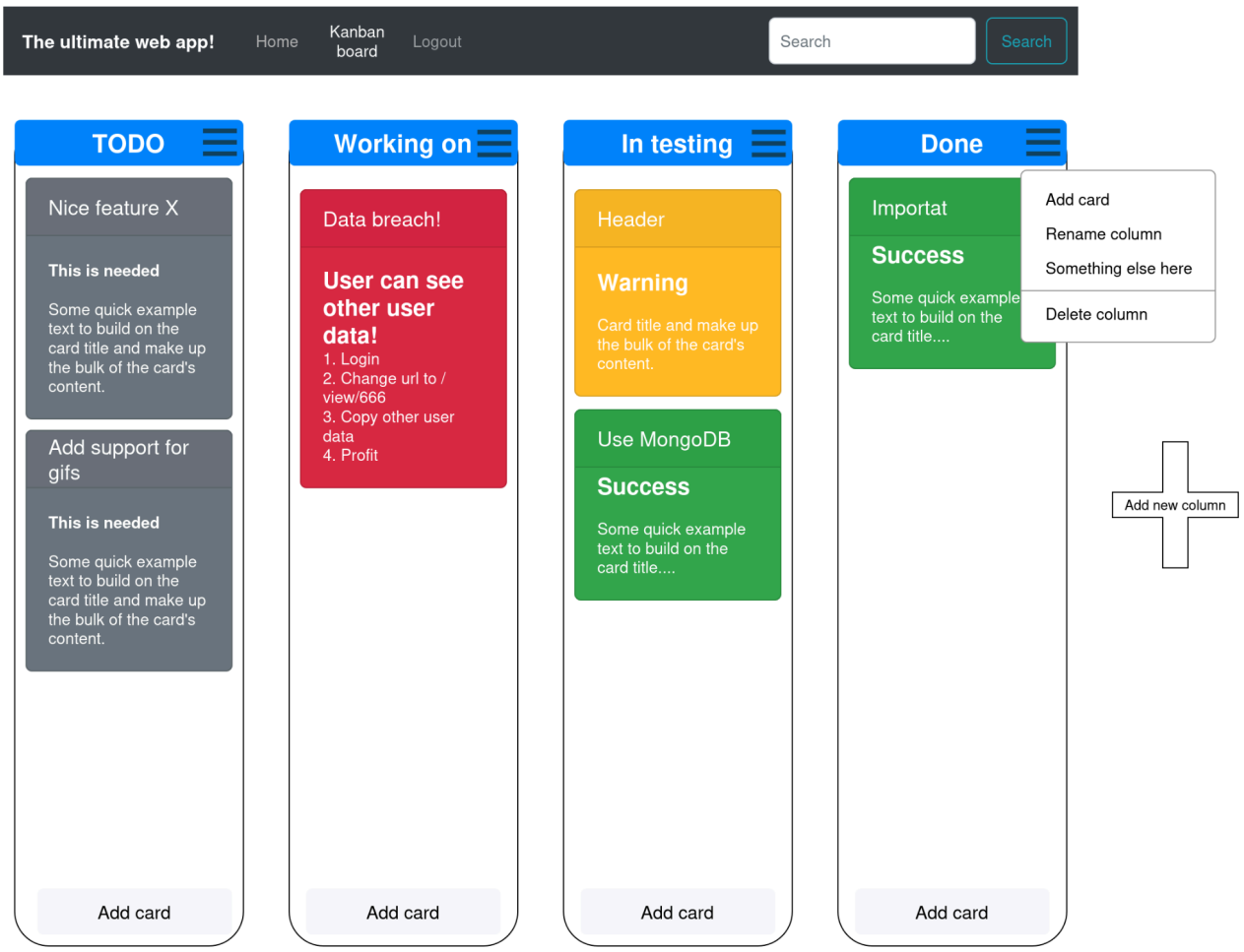
The DL for the project is 2.3.2025.
If you submit before 23rd of February you will get 1 bonus point.

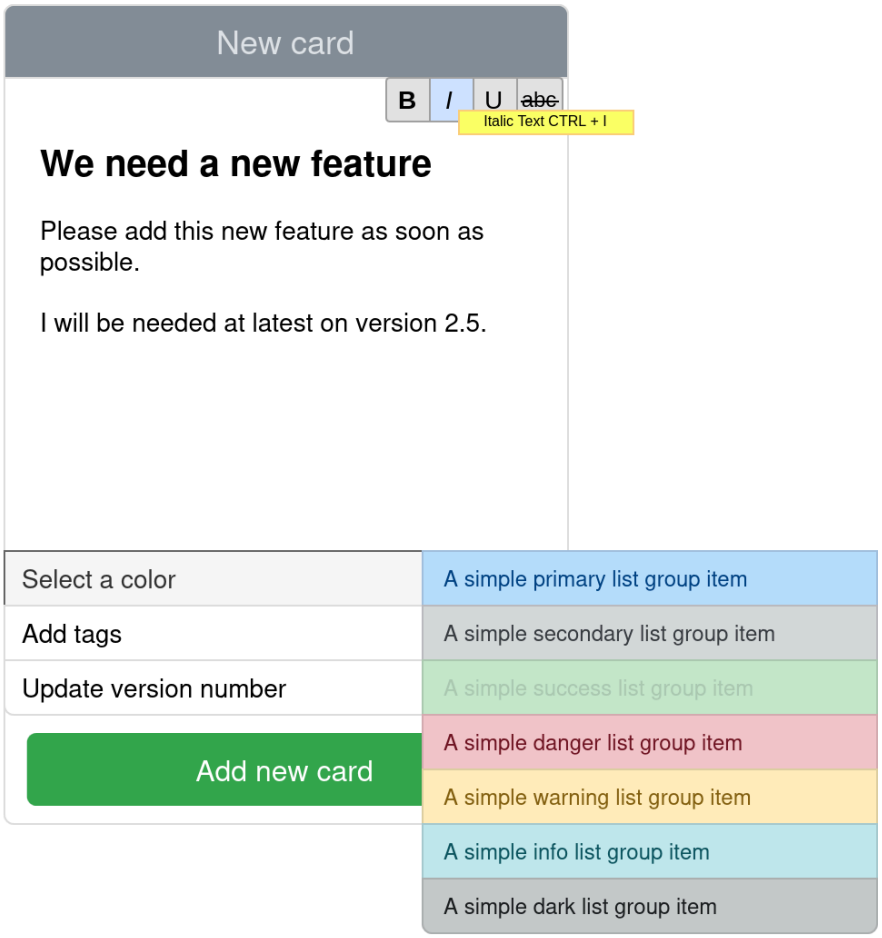
Introduction

Your task is to implement a system that lets users register, login and post cards to a kanban board. User can have as many columns she wishes in her board and she can move her notes across the board columns. In essence this is a similar system to, for example, Trello. Non-authenticated users cannot see anything. Data has to be saved to a database and usage of content management systems (CMS, e.g. WordPress or Drupal) is not allowed. Documentation has to be written. These basic features give 25 points from the project.

Besides the basic features you can implement as many additional features as you wish to gather more points and thus higher grades from the course. The maximum number of points one can get is 50.

The project is individual work and it could have views something like these (note that these images show features that are not required to pass the project work):





Mandatory requirements

- Implementation of backend with Node.js
  - You can of course use Express, Meteor or any other additional framework
  - Java, PHP, Perl, Python, Ruby or other language solutions are **not allowed** in this course
- Utilization of database
  - MongoDB, MariaDB or any other you see fit
  - All the data needs to be stored in a database
- Authentication
  - Users have to have an option to register and login
  - You can use JWT or session based authorization
  - Only authenticated users can see, add or remove columns or cards
- Features
  - Authenticated users can:
    - Add/remove/rename columns to/of their own board
    - Add/move/remove cards on/of their own board
    - Cards should be movable to both up and down and between columns (you can simply build buttons or have drag and drop feature)
    - Logout
  - Non-authenticated users can register and login

- Responsive design
    - The app needs to be usable with mobile devices and desktop browsers
        - Use of Materialize, Tailwind or Bootstrap is recommended
- Documentation
    - There needs to be documentation describing the technology choices, installation guidelines and user manual
    - Also list the features you have implemented and the number of points you are aiming at
    - **The documentation MUST also include declaration of AI usage:**
        - 1) Name all AI systems that were used in the development of the contents of this document, and for each
        - 2) How and where they were used (illustrations, proofreading, getting ideas for text, to generate diagrams etc.) or
        - 3) Clearly state that no AI assistance or tools were used in this assignment.
        - This Declaration is a mandatory part of the submission and leaving it out means that the work is incomplete and should not be graded. Declaring that no AI was used in the development of the document, but getting a high number in TurnitIn AI check is also a valid grounds for failing the work just by itself.

Possible features

Feature	**Max points**
Basic features (as stated in the previous chapter) with well written documentation	25
The application does not work	-100
No documentation	-100
If some parts are missing (such as no database or authentication does not work) points are reduced	0 – -25
If TypeScript is not used or it is used badly (like continuous use of any type)	0 – -10
Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other	3
Cards can be reordered with drag and drop	2
Columns can be reordered	1
User can set the color of a card	1
Login with Facebook, Google, X or other accounts (use Passport.js)	3

There is an admin account that can see all the users, all the boards and can remove or update them	3
Test software for accessibility; can it be used only with keyboard / voice command? Can screen readers work with your application?	3
Provide a search that can filter out only those cards that have the searched keyword	3
User has the option just to double click any edible content (like header or card description) and edit it	4
User profiles can have images which are shown on the main page and in the chat	3
Cards can have comments in them, one or many	3
Cards and comments have visible timestamps when they have been created and updated	4
Cards have estimated time, when the work is done	1
User is able to register the time she has spended with the task/card	1
Translation of the whole UI in two or more languages	2
Multiple users can work with the same board and the block (card/column/comment) is blocked from another user so that they are not messing with each other's work	5
Create (unit) tests and automate some testing for example with https://www.cypress.io/ (at least 10 cases have to be implemented)	5
Inappropriate content, including hate speech related memes and other trash	-100
Code is not written and commented in English	-10
Code is not commented at all	-10
Code is not commented properly	-5
<Your own feature: describe why should it be accepted> You can implement whatever you want your project to have – although it has to be somehow relevant to the theme.	n

Notes

- Front and backend do not need to be under the same project/port/whatever, but they can be different implementations
    - If more than 60k ports are used, your project will not be graded

Submitting

Submit your code to CodeGrade via link in Moodle.

Plagiarism

All the projects will be checked for plagiarism and all the incidents will be reported to the LUT misconduct. By submitting someone else's code without any mention of it you are doing a misconduct.