# Kanban Project Documentation

## Technologies used

This project was built using modern full-stack web development technologies with the following key components:

Frontend:

- HTML5 & CSS3: Provides the structural foundation and styling for the application.
- JavaScript: Handles all interactive and dynamic features of the user interface.
- Materialize CSS: Ensures a responsive, clean and consistent user interface design across all devices.
- Sortable.js: Enables smooth and intuitive drag-and-drop functionality for rearranging columns and cards on both desktop and mobile devices.

Backend:

- Node.js & Express: Forms the core of the server-side logic
- MongoDB & Mongoose: Manages data storage and handles database interactions.
- Multer: Manages file uploads (avatar images) and stores files in a designated folder.
- JWT with jsonwebtoken: Provides secure authentication and session management by generating and validating tokens.
- Bcrypt: Secures user credentials by hashing passwords before storage and comparing hashes during login.
- dotenv: Manages environment variables, ensuring sensitive data (like JWT secrets and database URIs) remain secure and configurable.

## Installation guidelines

1. Clone the project repository to your local machine using Git.
   `git clone https://github.com/TommiUp/KanbanProject.git`
2. Change to the cloned repository
   `cd KanbanProject`
3. Run npm install to install all the dependencies listed in the package.json.
   `npm install`
4. Start the application with npm start.
   `npm start`
5. Open your browser and navigate to address http://localhost:3000.

## User manual

Getting Started:

- Register: Navigate to the homepage and click the "Register" button. Complete the registration form with your email, username, and password.
- Login: After registering, log in using your credentials to access your personal Kanban board.

Using the Kanban Board:

- Adding and renaming columns:
    - Click the "Add New Column" button to create a new column.
    - To rename a column, simply open the column dropdown and select “Rename Column” or double-click (double-tap on touch devices) on the column title.
- Managing cards:
    - Within a column, open the dropdown menu to add new cards.
    - Enter a card title, description and optionally select a color to create a card.
    - You can edit a card’s title or description by double-clicking (or double-tapping) on them. If no changes are made, simply pressing Enter will close the edit field.
- Reordering columns and cards:
    - Drag and drop columns and cards using the provided drag handles to rearrange them as needed.
- Managing comments:
    - Click the "Comments" button on a card to view, add, or edit comments.
    - Edit or delete comments by double-clicking/double-tapping on them and using the provided options in the dropdown.

## Points Justification

Based on the basic project requirements and the project specific features, I have listed all project specific features that are implemented in my code with reasoning. I calculated total of 50-52 points and my points are justified as follows:

| **Project features** | **Points** | **Reasoning** |
|---|---|---|
| Basic features (as stated in the previous chapter) with well written documentation | 25 / 25 | Application contains basic features and the report is clean and structured. |

| | | |
|---|---|---|
| Cards can be reordered with drag and drop | 2 / 2 | The user can drag and drop cards within a column or move them |
| User can set the color of a card | 1 / 1 | The user can set the color of a card when creating one. Default color is white. |
| There is an admin account that can see all the users, all the boards and can remove or update them | 3 / 3 | The user is able to create an admin account that can see all columns/cards and remove/update them. |
| Provide a search that can filter out only those cards that have the searched keyword | 3 / 3 | Users can filter cards by entering a keyword, displaying only the columns and cards that contain that keyword. |
| User has the option just to double click any edible content (like header or card description) and edit it | 4 / 4 | Users can double-click column titles, card titles, descriptions, and comments to edit them. |
| User profiles can have images which are shown on the main page and in the chat | 3 / 3 | When creating an account, users can choose an avatar, which will be displayed on the main page after logging in and in the cards and comments chat. |
| Cards can have comments in them, one or many | 3 / 3 | Users can add unlimited comments to any card. |
| Cards and comments have visible timestamps when they have been created and updated | 4 / 4 | Each card and comment has a visible timestamp showing when it was created and last updated. |
| Translation of the whole UI in two or more languages | 2 / 2 | Website has 3 languages available: English, Finnish, and Swedish. |
| Extra feature: Users can edit and delete comments | 1 / 1 ? | Users can edit and delete their own comments, admins can delete and edit every comment. |
| Extra feature:  Webpage has colored notifications for all actions | 1 / 1 ? | Color-coded notifications for login, logout, registration, and errors. |

## Declaration of AI Usage

In the development of this document and the associated project, the following AI systems were used for help to improve clarity, correctness, and overall quality:

ChatGPT-4o:

- Usage: Provided suggestions for grammar, phrasing, and overall drafting of this documentation.
- Where: Used primarily for proofreading and refining the written content.

ChatGPT O1 and ChatGPT O3-mini-high:

- Usage: Assisted in generating code improvements, debugging, creating solutions to problems, drag & drop and adding descriptive comments to functions in the Kanban project.
- Where: Utilized for code suggestions in creating different functions and features, improving user interface interactions, and optimizing the overall codebase (drag and drop, touch support, double click/tap, sortable.js help, materialize help, styling the frontend side with CSS, visible notifications, timestamps, modifying typescript files to not continuously use “any” type). These tools have been utilized mostly in index.js, users.ts, columns.ts, cards.ts and styles.css.

DeepSeek:

- Usage: Helped with troubleshooting and debugging both backend and frontend issues.
- Where: Utilized during the development process to resolve technical challenges related to server and client-side functionality such as user registration fails, UI responsiveness for mobile devices and ensuring files get deleted from the database correctly. This tool has been utilized mostly in index.js, register.js, users.ts as well as in index and register html files.

All AI systems were used strictly as support tools. The final submission has been thoroughly reviewed by the student to ensure that it meets all project requirements.