

CS5003 Practical 2 - Group G

[210030785]

March 18, 2022



University of
St Andrews

Word Count: 960

1. Overview

Because the other student in this group has already applied for Leave of Absence with ASC team. So, this practical is finished all by me. The basic requirements have been achieved in this submission. It includes:

1. Allow a user to create a new Kanban board which has a unique code, a title, three columns named 'waiting', 'in progress', 'completed'. At the first stage of this board, it contains no cards.
2. If the unique code is entered in, then the board associated with this code could be viewed with all the previous stored information (cards, columns, title, descriptions) and waiting for further modification.
3. Allow the user to create a card with title and description into the column and
4. this card can be moved in the order from left to right.
5. Furthermore, the card can be deleted as well by clicking on the delete button.

2. Technologies and Resources

I developed my code from the example code given on the CS5003 student resources as a beginning protocol to start with.

The browser I run to test the solution is Chrome, Firefox, and the Safari.

VUE.js is used as a framework in the client.js document. So as mentioned in the lecture

```
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
```

is added in the header section of the index.html.

The port being listening is **23078** (generated by the command: 'id -u' in the lab terminal)

The version of NodeJS used for the development is Node.js v16.14.0.

3. Design

a brief discussion of any interpretation of requirements and any design decisions taken. Focus on the reasons for your decisions rather than just providing a description of what you did.

Index.html:

The user interface web page includes three major parts:

First section is a legend-field set which allow the user to enter the code and title for the new board, the reason of choosing field set is allowing user to logically group sets of fields in order that the forms can be more descriptive: a board with a unique code and then a title.

The second section is the new card generating section:

In this section, the user can input the new card details and make change to it before adding it into the board.

The final section is the three columns section:

What is different from the other section is there are container section which can contain cards. Also, a 'btns' section attached with the VUE methods to manipulate the cards.

Client.js:

On the client side I use the VUE framework. Because it provides responsive visual components, and it makes it much easier to handle HTML blocks.

The structure of VUE make the whole client.js more understandable and neater.

We are facing **4** kinds of data in this project: 1. Board form, 2. Board, 3. card form, 4. Message

We are constructing **7** methods to do all the solutions of data representing and manipulating. Such as: dealing with error, status checking, adding, loading, moving, deleting.

Server.js:

The reason of creating a **const variable named boards** is to store al the boards submitted from the client side.

A createBoard function is built which is for collaborate with the app.get and app.post to get all the information from the client and reflect these details onto the clients once the code of this specific board is inserted by the user.

A formatBoard function is built to recognize the different status of board:

Waiting/in progress/completed which is corresponding to the previous function const variable board.

Delete:

When design the delete function, firstly, a validation should be made to ensure if this code exist or not. If it exists, then find this code and the index of that specific card, if the index of that card does not exist, return 400, else use the splice method to remove this specific card. Then the formatted board should be sent back to the client side to display the modification of removal.

4. Teamwork

I used a little bit of Git following the instructions given in the lecture, however because the other student is not available. I can't have interactions in the Git as I supposed to have.

However, I tried to use the Git installed on the local machine and create the repository, clone it as practise purpose. I hope I can use it in the next group practical.

5. Evaluation

The steps run to evaluate the project:

Use the terminal to go to the right directory, run the command '**npm install**', run the command '**node server.js**', then '**Listening on the port 23078**' occur in the terminal.

Then open the browser enter '**localhost:23078**' Then the interface displayed on the web browser.

←

→

↺

localhost:23078

☆

🔍

☰

KanBan Board

Boards

Code

Title

Add

load

Then to test the requirements:

Insert 1 into each blank to check if the information display correctly.

It shows that it can create an empty board with three columns, and can be added card on the board, also cards can be moved from the left to right.

← → ↻

localhost:23078

☆

🔒

☰

KanBan Board

Boards

Code 1TitleAddload

New Card

Card Title

Card Description

Add

Waiting

1MoveDelete

1MoveDelete

11MoveDelete

111MoveDelete

In Progress

Completed

The created cards can be moved from the waiting column to the in progress or completed columns.

KanBan Board

Boards

Code 1TitleAddload

New Card

Card Title

Card Description

Add

Waiting

111MoveDelete

111

In Progress

1MoveDelete

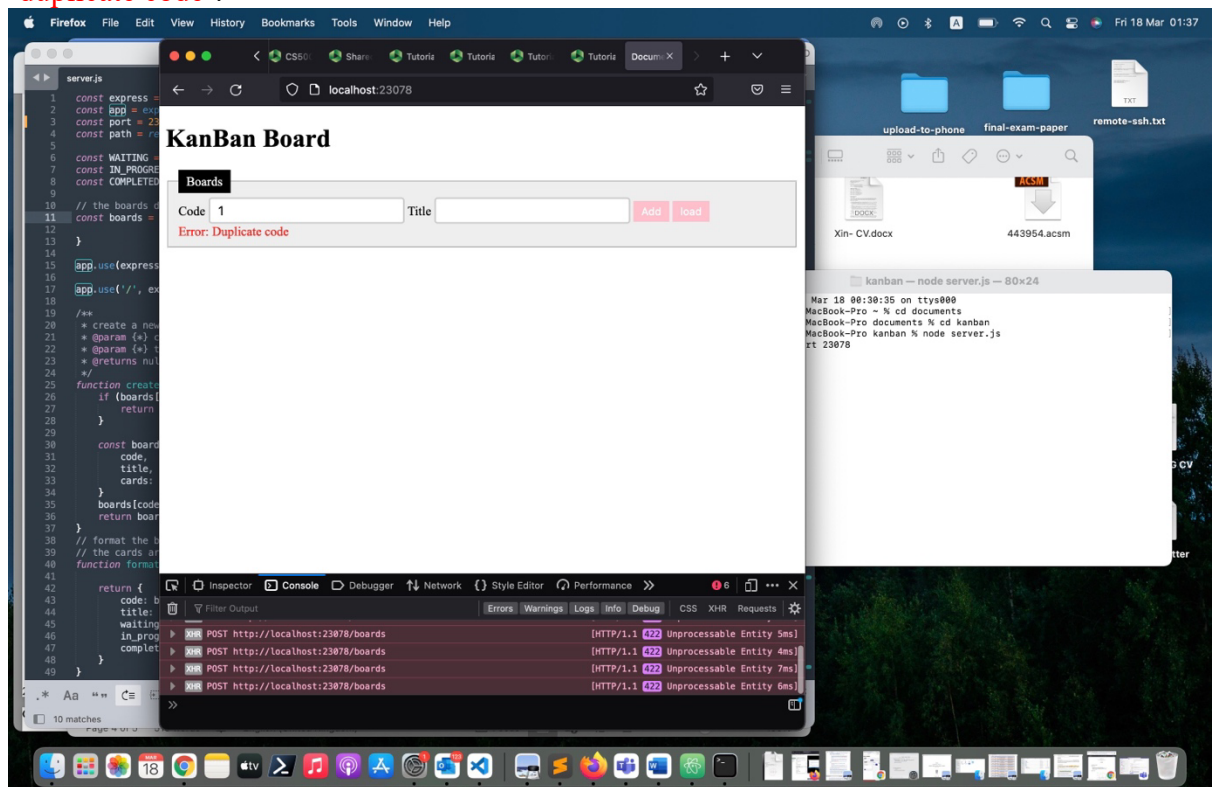
1MoveDelete

Completed

11Delete

11

When '1' is inserted again, the system would not create a new board by a message of 'duplicate code'.



Then when the '1' is typed in the code section, click the load button the previous details could be shown exactly the same:

KanBan Board

Boards

Code Title

New Card

Card Title

Card Description

Waiting

111

111

In Progress

1

1

Completed

11

11

When click the delete button, the card would be deleted.

KanBan Board

Boards

Code 1TitleAddload

New Card

Card Title

Card Description

Add

Waiting

In Progress

1MoveDelete1Delete

Completed

11Delete11

Moreover, input a different code '2' to create a new board:

KanBan Board

Boards

Code 2Title 2Addload

New Card

Card Title

Card Description

Add

Waiting

22MoveDelete22MoveDelete222MoveDelete222Delete

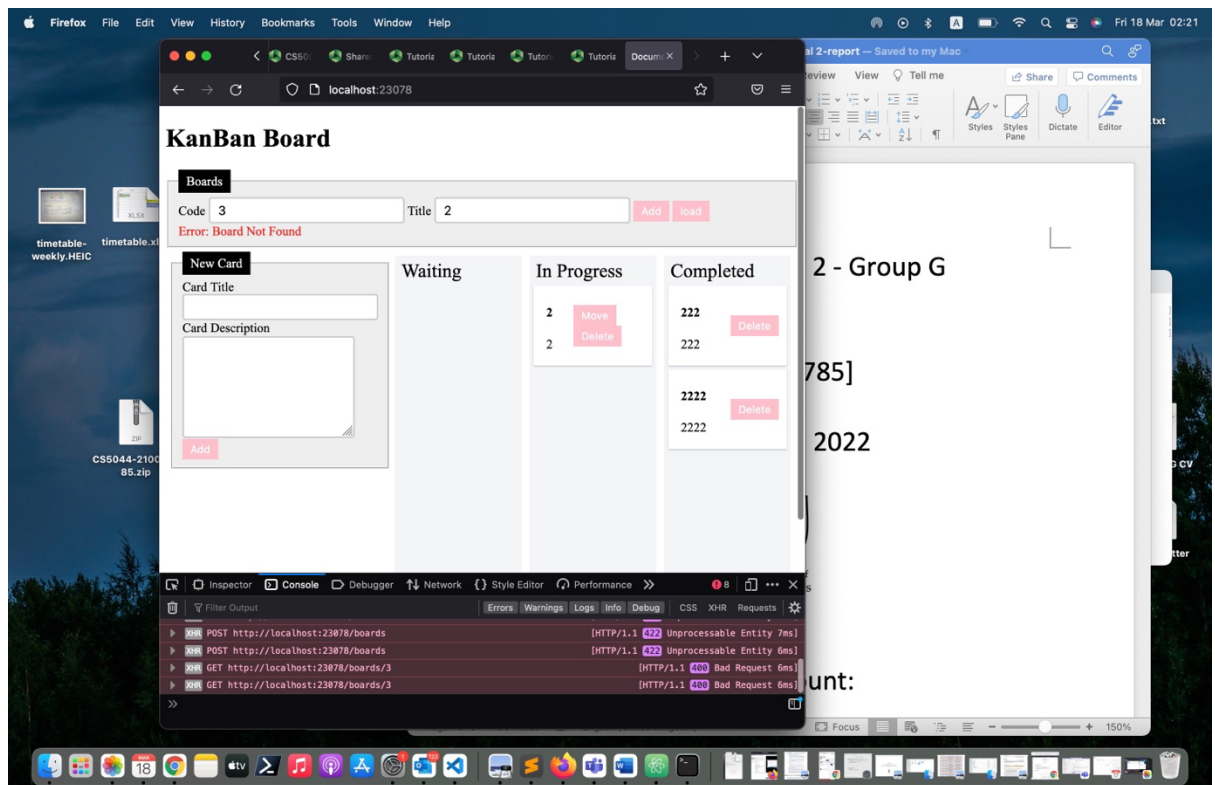
In Progress

2MoveDelete2Delete

Completed

2222Delete2222

When an **invalid** code is offered by the user , then the user click the **load** button ,then the system would pop out a message: **board not found**



Self-reflection:

If I have more time I wish I can go deeper to build more code to fulfil the intermediate and advanced requirements.