CT30A3204 Advanced Web Applications - Luento-opetus 31.10.2022-5.3.2023

web project

files are stored in [kongbai23330/webproject (github.com)]

Mandatory requirements

The backend uses Node.js, uses Express, and the language is JavaScript,

The database uses MongoDB,

Users have the option to register and log in, using tokens for authentication, only authenticated users can post,comment

Authenticated users can:

post new code snippet

Comment on an existing post

Unauthenticated users can see posts, comments (and vote counts)

There are some pages that list all the posts, each post is displayed in the form of a card, click the button to display comments

Responsive design is used, the react framework is used, and the Materialize library is used for rendering.

Installation Guide:

Project's pakage.json page has various instructions.

tips: In my github version, **I don’ t know why npm run dev:client will report an error, so just “cd client” and then “npm install --force”, “npm start”**

In the process of npm install, there may be a lot of warnings. This is because I use some dependent libraries of lower versions, which have no effect on the running of the program and can be ignored.

And so we run **"npm install --force"**

In order for the server to work,you need to run **npm run preinstall**

Then ,run”npm run dev:server”

```
{
  "name": "week144",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ 调试
  "scripts": {
    "install": "cd client && npm install",
    "dev:client": "cd client && npm start",
    "test": "echo \"Error: no test specified\" && exit 1",
    "preinstall": "cd server && npm install",
    "dev:server": "cd server && npm run dev"
  },
  "author": "",
  "license": "ISC"
}
```

```
PS E:\web\test1231\webproject-main> cd client
PS E:\web\test1231\webproject-main\client> npm install --force
npm WARN using --force Recommended protections disabled.
npm WARN ERESOLVE overriding peer dependency
npm WARN While resolving: @mui/styles@5.11.11
npm WARN Found: react@18.2.0
npm WARN node_modules/react
npm WARN   peer react@">=16.8.0" from @emotion/react@11.10.6
```

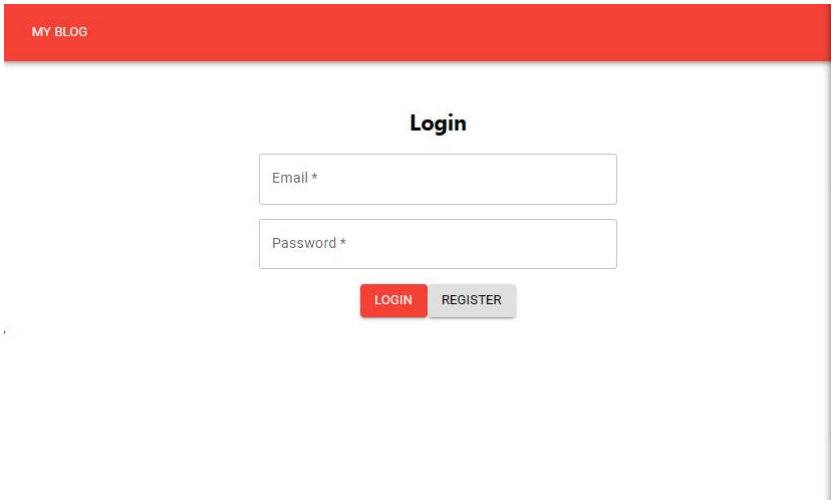
```
PS E:\web\test1231\webproject-main\client> npm start

> client@0.1.0 start
> react-scripts start
```

```
PS E:\web\test1231\webproject-main> npm run dev:server

> week144@1.0.0 dev:server
> cd server && npm run dev


> server@0.0.0 dev
> node ./bin/www
```

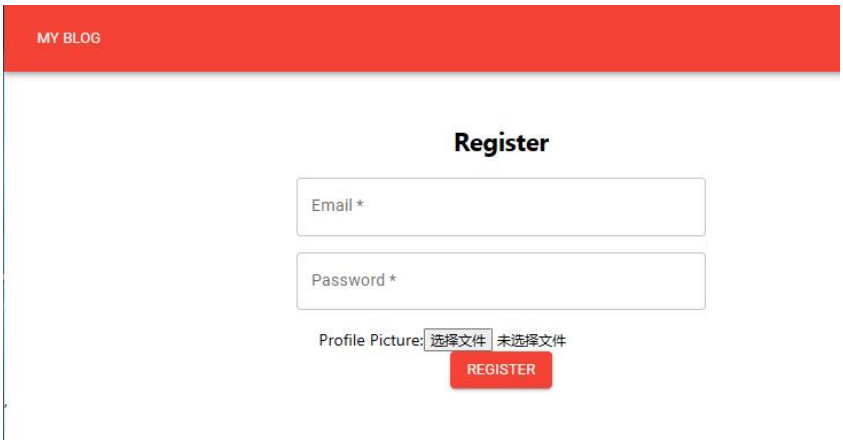
Both the front end and the server end need to run the program to proceed Normally.

User Manual:

First of all, the running program should open the local address localhost:3000, and the initial interface is login

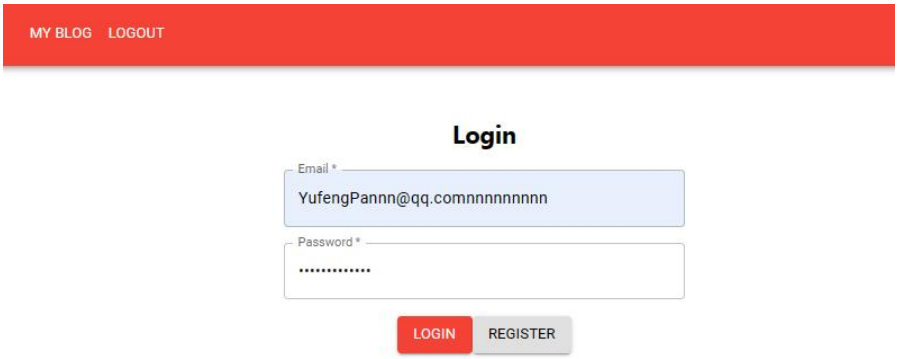


For the first use, you should register first, click the register button to enter the registration interface.After entering the registration interface, you can fill in your email and password, and you can also upload user pictures

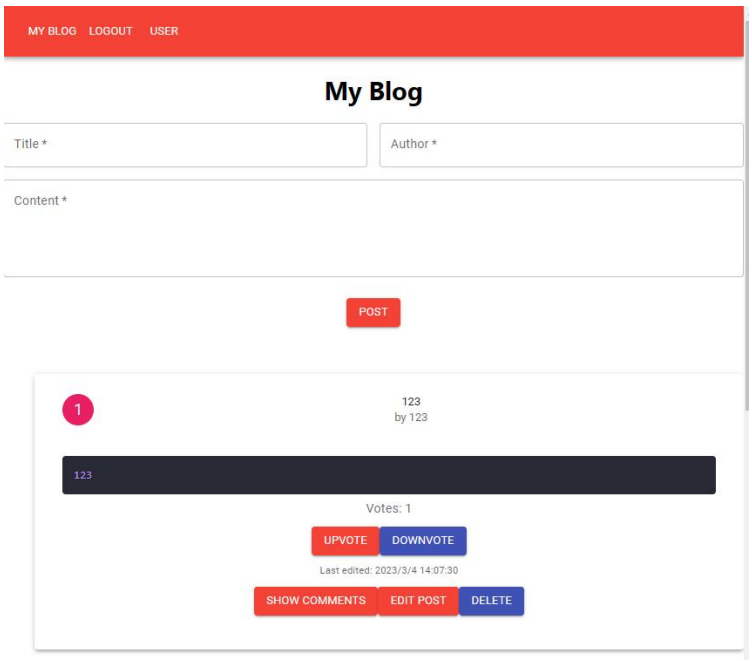


After registration, enter the login page, click the login button to log in successfully, and you will jump to the blog interface, or you can click the MY BLOG button

If you want to log out, please click the LOGOUT button



Click MY BLOG, then enter the blog interface



After entering the blog interface, you can write a blog, press the post button to upload, and at the same time, you can also realize the above functions

If you want to display comments, click the SHOW COMMENTS button, edit by pressing EDIT POST, and delete by pressing

DELETE, only the creator of the post can edit and delete, only logged in can post and comment, delete.

Click on the User interface to enter the information page of the registrant

If you upload a picture, you can see the picture on this page

Next is **the implementation of detailed functions**

First list the functions I implemented

Basic features (as stated in the previous chapter) with well written documentation 25

Users can edit their own comments/posts 4

Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other 5

Use some highlight library for the code snippets, for example https://highlightjs.org/ 2

Admin account with rights to edit all the post and comments and delete content (if a post is removed, all its comments should be removed too) 3

Test software for accessibility; 3

Vote (up or down) posts and comments (only one vote per user) 3

User can click username and see user profile page where name, register date, (user picture) and user bio is listed 2

Last edited timestamp is stored and shown with posts/comments 2

Create (unit) tests and automate some testing for example with Cypresshttps://www.cypress.io/ (at least 10 cases have to be implemented) 5

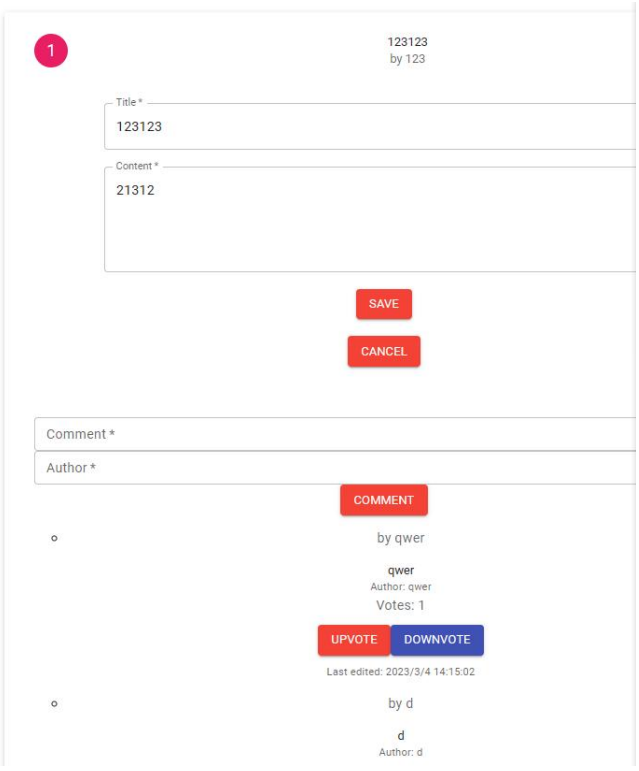
The target score is 57 points. Some parts may be done according to my inappropriate understanding. I hope the teacher will understand

What follows is an introduction to each function:

Users can edit their own comments/posts

You can change the post and comment after clicking EDIT POST

Click save to save the changed content.

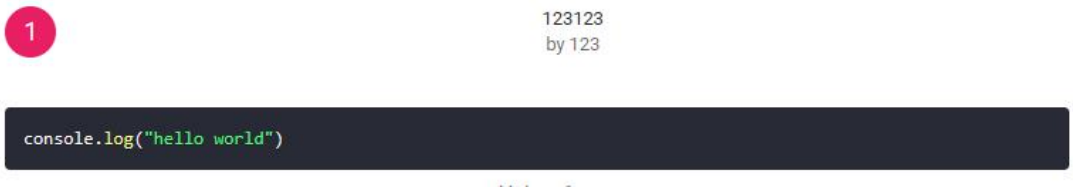


Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other

Using the react framework

Use some highlight library for the code snippets, for example https://highlightjs.org/

Using the highlight library



Admin account with rights to edit all the post and comments and delete content (if a post is removed, all its comments should be removed too)

It should first register an administrator account

The administrator account is Administrator@gmail.com

password is !Abc123321123

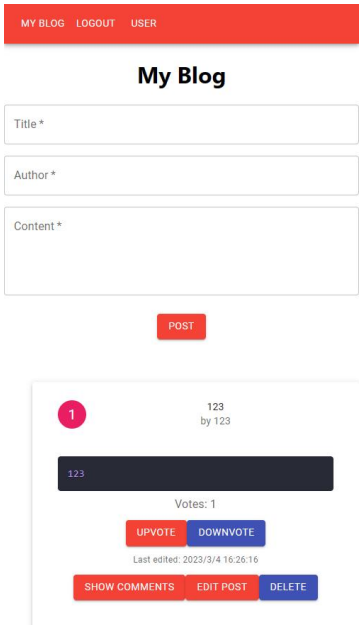
can edit and delete anything.

Test software for accessibility;

Tested on the andriod virtual machine, you can input something, and it can change with the change of the screen size,

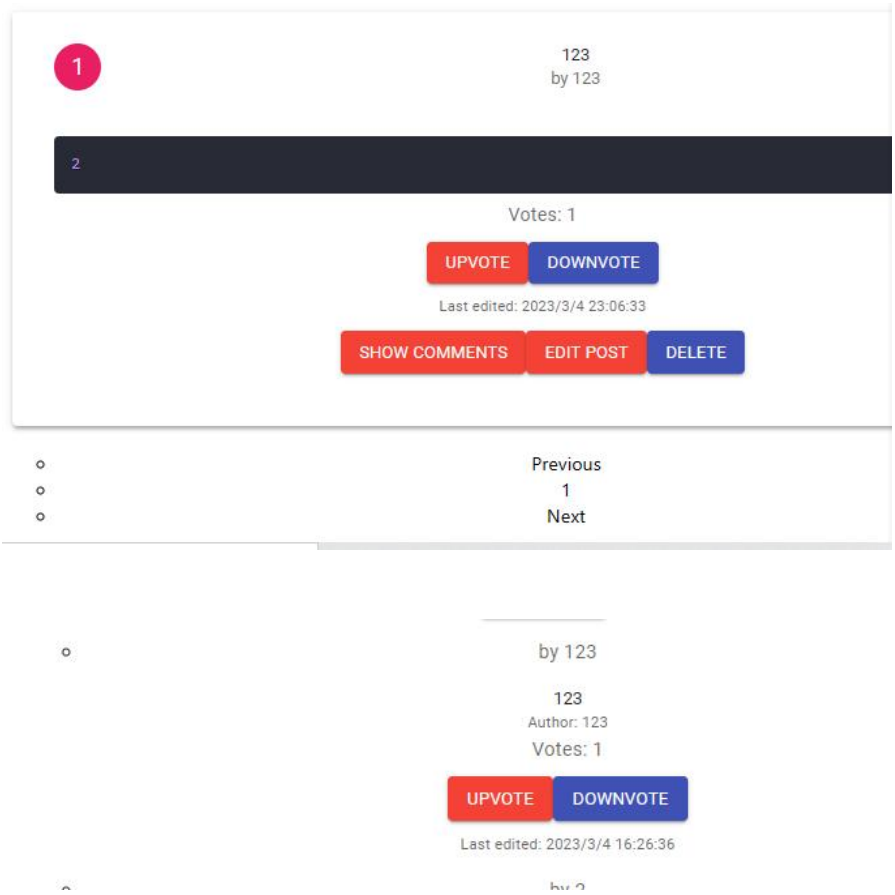
It can be operated completely by keyboard, using the tab key, enter key and letters to operate

Screen readers work with my application



Vote (up or down) posts and comments (only one vote per user)

Can vote, each person can only cast one vote, vote for or against, comments can also vote.



User can click username and see user profile page where name, register date, (user picture) and user bio is listed

You can click the user name to enter the user profile page

Last edited timestamp is stored and shown with posts/comments

The time of the last edit exists, in posts and comments.

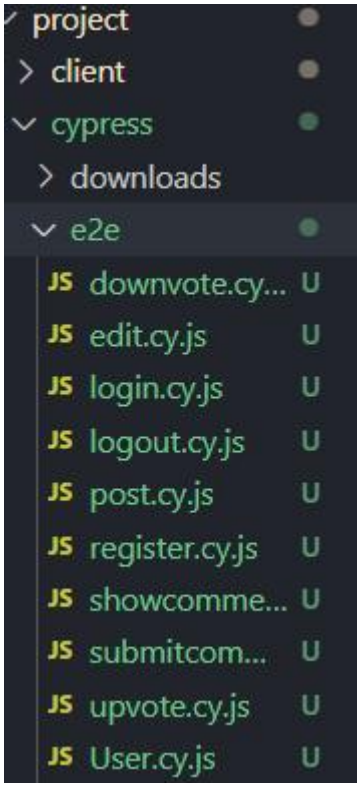
Create (unit) tests and automate some testing for example with Cypresshttps://www.cypress.io/ (at least 10 cases have to be implemented)

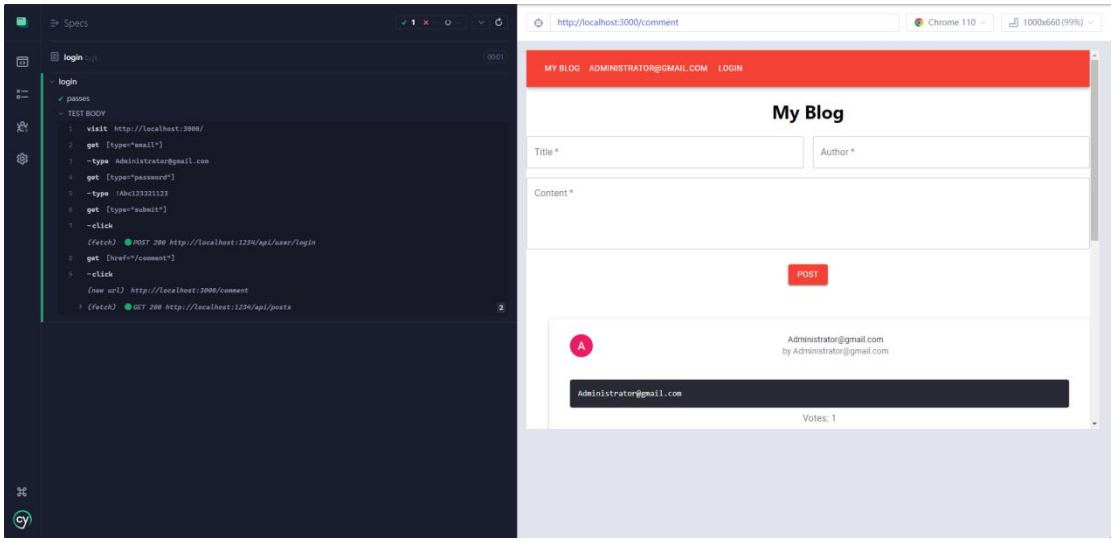
First npm init, npm install, npm install cypress, and

Use "npx cypress open"; to test

```
PS E:\web\project> npx cypress open

GET /__/ 200 2.280 ms - -
GET /__/assets/polyfills-72632428.js 200 8.774 ms - -
GET /__/assets/index-db587a01.css 200 1.309 ms - -
GET /__/assets/index-857c938c.js 200 5.987 ms - -
GET /__cypress/runner/cypress_runner.css 200 1.196 ms - -
GET /__/assets/Specs-da0a70ec.js 200 1.385 ms - 485
GET /__/assets/paper-airplane_x16-b7a1d50c.js 200 2.441 ms - -
GET /__/assets/Index-3c96f22c.js 200 2.052 ms - -
GET /__/assets/CreateSpecModal.vue_vue_type_script_setup_true_lang-04a0d69e.js 200 0.
GET /__/assets/box-open_x48-ba3feff5.js 200 1.515 ms - -
GET /__/assets/CreateSpecModal-ede9e79a.css 200 1.094 ms - 309
GET /__/assets/SpecPatterns.vue_vue_type_script_setup_true_lang-92900aad.js 200 1.454
GET /__/assets/SpecNameDisplay.vue_vue_type_script_setup_true_lang-6d479e8c.js 200 2.
GET /__/assets/ResultCounts.vue_vue_type_script_setup_true_lang-f2778e24.js 200 1.820
```

Conduct 10 e2e test





The above is all my introduction

Re-list the features:

First list the functions I implemented

Basic features (as stated in the previous chapter) with well written documentation	25
Users can edit their own comments/posts	4
Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other	5
Use some highlight library for the code snippets, for example https://highlightjs.org/	2
Admin account with rights to edit all the post and comments and delete content (if a post is removed, all its comments should be removed too)	3
Test software for accessibility;	3
Vote (up or down) posts and comments (only one vote per user)	3
User can click username and see user profile page where name, register date, (user picture) and user bio is listed	2
Last edited timestamp is stored and shown with posts/comments	2
Create (unit) tests and automate some testing for example with Cypresshttps://www.cypress.io/ (at least 10 cases have to be implemented)	5

The target score is 54 points.