**Review for usability and accessibility**

This website design has so far met the standard requirements mentioned in the lectures and assignment description. The feature details will be shown in the feature plan.

The website is able to perform the basic usability in the milestone stage according to the code testing. Some advanced features have not yet been implemented. They will be added into the code in the further coding process as we learn more about implementing web applications throughout this course.

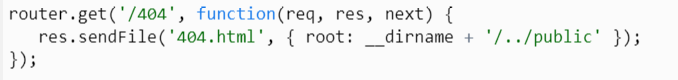
## **The basic design of this website has been peer reviewed and the feedbacks are shown below:**

* Additional information should be added to the manage profile page. For example, phone number of user or manager.
* Brief and clear page layouts. Pictures with detailed content can be attractive to the web browsers!
* On the home page is not good for user to enter the user name and email again. May need to redesign the home page.

## **Record any updates to your design made as a result of the above testing and review:**

* Add phone number of users will need to be reconsider.
* Need a clearer design of profile page, add a ‘back’ button will be considered.
* Home page will be left with two buttons, sign up and log in only.
* The appearance of login page has been improved to be concise and clear.
* Add confirm password section in the sign-up page.

**Reflection**

1. Change main colour from blue to pink.
2. Design a webpage logo, add landing animation for a logo gif to all pages.
3. Instead of provide sign up and sign in page only, redesign the index page with webpage logo, title and main function screenshots then add a sign-in section.
4. Within the Todo page. First, it needs to redirect(refresh) the Todo page to show/get the new tasks when it is been create. Then, I choose to use v-select and Bootstrap Modal Plugin to create a add task function, it will not need to redirect other pages when user create a task, it will hide when data-dismiss="modal" is add to a button.
5. After add a task, a post request is use to get the new task list to update the task. Frist, it needs to refresh the page to show the changes, then I use $("#tasklist").empty(); to empty the div and add the task list to div again. Btw, in this project, I start to learn jQuery, because it is more simple and shorter than js.
6. Instead of writing the same html code in nav-bar and footer, I choose to render each section in route.js then send it to each page when its needed.
7. I try to use fullcalendar plugin, finally it can get my sql database by calendar.addEvent()
8. In task and calendar page , each task has category and priority, and it has different colour in css file when we render the task list.
9. For users more easily use the webpage on their phone, I use @media(max-width:359px) to change some style files in responsive.css.
10. Update the render function in top-nav-bar, for admin user, they can see the page in user section, then they can access edit user, user list and so on. For normal user, the user section in menu bar will not be seen.
11. For 404 request, I use to write code in route.js by 

Then I got the idea from project10, to use next(createError(404));

Finally it becomes to 