Web Technologies - An Online Blog

Development report

1 Introduction

This blog is my deliverable for the Web Technologies assignment. The main reason for developing a blog is that I want to practice both client side and server side skills based on what I learn form the course. Besides, I'd like to use the Google for sourcing the new technologies (of course open-source) which I can used for my development. Now before detailed description each file of this project, some features should be pointed out first:

Development environment: Mac Os 10.11.3
Server side environment: Node.js v4.3.2, npm v2.14.12, Express v4.13.1, mongodb v3.2.6

The reason I want to use Express because this kind of framework is more easier to start work with. At the beginning I use **Express application generator** to quickly create an application skeleton. The express framework is the basis for this project.

2 How to run the server

- Initialization and run mongodb
  - Download mongodb at https://www.mongodb.com/download-center#community.
  - Gunzip the file, rename the folder as mongodb.
  - Open a terminal, input the following code:

```
$ cd mongodb
$ mkdir blog
$ cd bin
$ ./mongod --dbpath ../blog/
```

  - Now the mongodb is running.
- Initialization and run the server
  - Open another terminal (do not close the original terminal)
  - Enter the server folder, input the following code:

```
$ cd blog
$ npm install
$ DEBUG=blog:* npm start
```

  - Now the server is running.
- http://localhost:3000/ to visit the port

3 Server side

app.js The start up file for the server. This file is responsible for:

- import node modules
- import router module
- set the template engine as ejs
- save the cookie in db for 30 days
- catch 404 and forward to error handler

settings.js This file is responsible for saving the configuration information such as the port of mongodb.

routes/index.js index.js is responsible for routing control and routing function implement. The routing planning summary is:

- /: homepage,
- /reg: users register,
- /login: users login,
- /post: write blog,
- /logout: users logout,
- /archive: the summary page of blogs,
- /tags: the summary page of tage,
- /p: the blog page,
- /edit: edit exist blog,
- /remove: delete exist blog,
- /u: the user page,
- /uploadImg: upload img though the KindEditor

4 Client side

Client files contains "views" folder for HTML5, public/stylesheets folder for CSS, public/-javascripts folder for javascript and public/images folder for Inkscape and GIMP.

HTML5 In this project I used the ejs (the Express default) as the template engine for HTML. In this case, there is a little bit **difference** between my HTML file and the Polyglot HTML5. The main difference is that the ejs is supporting dynamic rendering which means that I can pass some parameters from the server side to create the client HTML file. More specifically, the ejs includes:

- %= code % replace code as the parameter in .html file

CSS In this project I used both my own CSS file (style.css) and a open-source CSS library named Bootstrap (currently v3.3.6) which provided by Twitter. The Bootstrap library contains a lot of beautiful style such as buttons and nav bar. And Bootstrap could help me make some responsive div as well.

JavaScript In this project I used my own javascript file (script.css) to import a random background image as header and initialize the KindEditor which is a plug-in component for writing blog.

Inkscape and GIMP All the images in the project are created by myself using Inkscape and GIMP (expect the website logos in footer). At first I was used Inkscape to draw a logo for this blog, then I used GIMP to add layer to make it looks good as the background image. The favicon is also based on the blog logo.

5 Summary

In the process of this development, I spend a lot of time to try and wrong. For example, at first I used "markdown" for writing the content of post and comment and "multer" for upload images. But soon later I found the drawback of these modules so I try to using new one to replace them. The "markdown" editor is not friendly to common users so I choose "KindEditor" which is much easier to use. In this case, soon later I also found that the "formidable" works better with "KindEditor" for uploading images than "multer", I used new module again. As well as mongodb, as first I used "name" + "title" + "date" as the "key" to search the post, but there is a bug that the same user cannot post a blog with same title with the same day. Although this kind of situation is not so frequency, I still found a new way to solve the problem. The solution is that using the "_id" which automaticly created by mongodb as the key, it's more accurate and reasonable. I always change my mind in the process of this development like this, and the lasted idea is that I want to put my blog in Heroku.com.

Hopefully I will set it up in Heroku.com in the following week, and the blog website should be http://lazydingding.herokuapp.com/.

Special Thanks Ian. Holyer for the guidance in this course.