

Documentation about my project

Name: Hongwei Tian

Student ID: N13761741

Introductions

I use node JS + Express + MongoDB to implement my web application.

In this application, all users will be regarded as guest user by default, who has permission to upload pictures and view all pictures which has been sorted by uploading time.

Note: When there is no previous page or next page, “previous” button or “next” button will not appear on the web page.

If someone sign up an account and sign in that account, when he uploads pictures, the application will link this picture to the user in the database. In addition, he can view all his own pictures and delete them.

Note: I did not implement any GUI notification when deletion of images success. But it indeed has been deleted because the record in database has been deleted.











Guest user will see 2 navigation item on navigation bar, and sign-in user will see three --- including “My Own Pictures” item.

In the database, there as 2 collections --- “userinfos” and “imageinfos”. In “userinfos” collection, it will store information about users, including username and password. In “imageinfos” collection, it will store information about images, including caption, image path, the user submitting it and uploading time.

How to Run This Web





1. Install node JS and MongoDB.
2. Start MongoDB server with database path:
path_to_this_app/mongodb/M_DB
and with log file path:
path_to_this_app/mongodb/M_LOG/mongodb.log
3. Within the root directory of this application, run “node app” or “node ./bin/www” to run the web.
4. Finally, use browser to view “localhost:8080”. **The sever listens 8080 port.**

Architecture of This Application

Name
 app.js
▶  bin
 Instructions
▶  modules
▶  mongodb
▶  node_modules
 package.json
▶  public
▶  routes
▶  views

The entrance of this web is “app.js”.

In “modules” directory, they are database communication modules, like this:

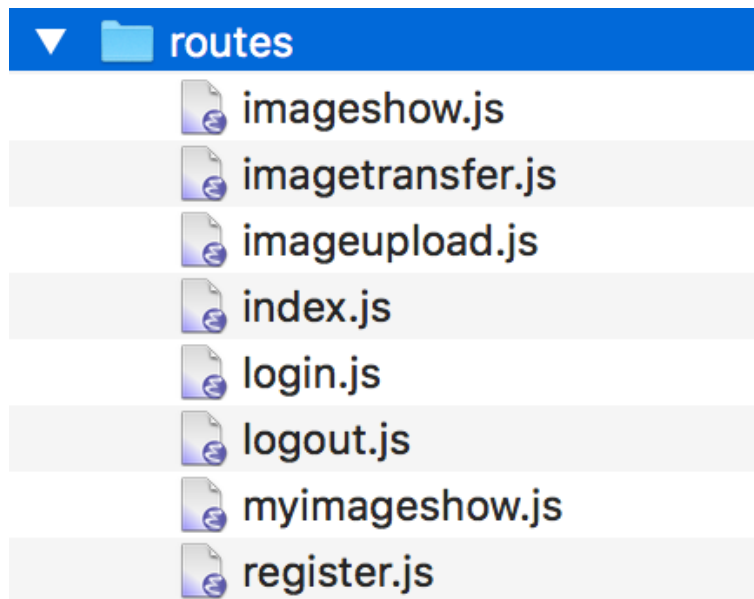
▼  modules
 db.js
 images_db.js
 user_db.js

“db.js” is Mongoose initialization module.

“user_db.js” has some functions to interact with “userinfos” collection in the database.

“images_db.js” has functions to interact with “imageinfos” collection.

In “routes” directory, there are all router modules which is used for different request url. In addition, most of those have the corresponding view module in “views” directory.



“index.js” deals with request to the index page of this web, it will redirect the request to “imageshow” page.

“imageshow.js” deals with request of showing all pictures.

“myimageshow.js” will show only those pictures belonging to the sign-in user.

“logout.js” will log the user out, I mean clear the session.

“register.js” deals with request to sign up an account.

“login.js” deals with sign in request.

“imageupload.js” deals with request to upload a file.

“imagertransfer.js” deals with “GET” request to images.

Test Result

This web serve normally as expected and I leave the test images and database in the folder to let you know the result.

I have 2 test account in the database:

- | | |
|----------------------|------------------|
| 1) Username: Jecky | Password: 123456 |
| 2) Username: JeckyOH | Password: 123456 |