

Sprint2

Instructions for 'How to run your product'

Team: Nanxin Jin, Yang Shi, Weijie Meng, Jintao Zeng, Yaoxi Liang

1. The architectural dependency

The product relies on the node.js and mongodb, we introduce how to set up required package and install database in the second part.

Live website (view only):

<https://cs408.herokuapp.com/>

2. User manual

Install instruction (run in localhost):

0. Install node.js & mongodb

<https://nodejs.org/en/>

<https://docs.mongodb.com/v3.2/tutorial/install-mongodb-on-os-x/>

1. Run this command line in the location of your CS408 folder, this will install the required package.

```
npm install
```

2. Run mongodb in terminal

```
mongod
```

3. Run server in terminal

```
node server.js
```

4. Open localhost url. However, the web makes API call will server on heroku, not in local host. You can use [Postman](#) to test API calls.

<http://localhost:3001/>

File structure:

/server.js:

The basic server file that runs all the function of the website.

/model/:

The schema of the database.

/controller/:

The controller to handle the API request function.

/app/route.js:

Router to route all API calls to functions in controller.

/service/:

Some helping files like CORS.

/middleware.js:

A file that will check the API before goes to route.js. For example, if a user include a auth header, it will check whether the user is valid, then it will go on next execution or respond an error.

/public/:

The front-end website file including html, css and javascript.

API:

Check /route.js and /controller/*.js for API call details.

For example, here is a post function from /route.js and /controller/market.js:

```
app.post('/api/market', middleware.requireAuthentication, market.postMarket);
```

```
postMarket: function(req, res) {
  var body = _.pick(req.body, 'title', 'content', 'price');
  body.createdAt = moment().valueOf();
  var newMarket = new Market(body);
  newMarket.save(function(err, item) {
    if (err) return res.status(500).send();
    req.user.postedMarket.push(item);
    req.user.save(function(err, doc) {
      if (err) return res.status(500).send();
      res.status(204).send();
    });
  });
}
```

Detail of API Call is:

Method: POST

Url: baseUrl + '/api/market'

Header: 'auth' : tokenString

Body:

```
{
  title: titletext,
  content: contentText,
  price: priceNumber
}
```

Front-end:

Some of the html has its own js file in the /public/js file. We use AJAX to send API call in front-end. When you are using login in the home page, it calls the ajax in /public/js/homePage.js

Here is the tutorial to use AJAX:

http://www.w3schools.com/jquery/jquery_ref_ajax.asp

