Shengkun Xie

Colton Vermilyea

Zhen Wang

Github link: <https://github.com/NWEN304Group/GroupProject>

**Set up project**

1: Use command > node app.js to start server.

2: Open browser and visit <http://localhost:8080/>

(We binding Facebook login URL to our Heroku URL, so Facebook login will not work on the local server.)

**Using Heroku web service**

<https://nwen304-shopping-project.herokuapp.com>

**REST interface**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| URL + HTTP Verb | Parameters | Descripti  on | Response status  and body (when  successful) | Response status and body (when failed) |
| ‘/products/:category\_id  ’+get | category\_id | a page that show all products of one category | 200 OK | 500  **message**:**"Internal Server Error"**  Request:**/products/'**+req.**params**.category\_id |
| ‘/product/:product\_id  ’+get | product\_id | a page that show details of a product | 200 OK | 500  error:err,**message**:**"Internal Server Error"**, **request**:**'get /products/'**+req.**params**.product\_id |
| ‘/page/:page  ’+get | page | a page that show products that index is :page | 200 OK | 500  error:err,**message**:**"Internal Server Error"**, **request**:**'GET /page/'**+ req.**params**.**page** |
| ‘/pagenotlogin/:page  ’+get | page | Get a page that show show products that index is :page with login buttons | 200 OK | 500  **error**:err,**message**:**"Internal Server Error"**, **request**:**'GET /pagenotlogin/'**+ req.**params**.**page** |
| ‘/product/:product\_id  ’+post | product\_id  User\_id | add product to user's cart and update cart | 200 OK | 500  error:err,**message**:**"Internal Server Error"**, **request**:**'POST /product/'**+ req.**req**.**body**.product\_id |
| ‘/cart  ’+get | user.\_id | Show cart page | 200 OK | 500  **error**:err,**message**:**"Internal Server Error"**, **request**:**'GET /cart'** |
| ‘/removeall  ’+post | user.\_id  req.body.item | remove all quanlity of one product in cart | 200 OK | 500  error:err,**message**:**"Internal Server Error"**, **request**:**'POST /removeall'** |
| ‘/removeone  ’+post | user.\_id  req.body.item | Remove 1 quanlity of one product in cart | 200 OK | 500  **error**:err,**message**:**"Internal Server Error"**, **request**:**'POST /removeone'** |
| ‘/payment’ + get | User id | confirm payment | 200 OK | error:err,**message**:**"Internal Server Error"**, **request**:**'GET /payment'** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GET / | NaN | get index page of website | index web page  200 OK | 404 Not Found |
| GET /login | NaN | get login page | login web page  200 OK | 404 Not Found |
| POST /login | email and password | login user | profile page  200 OK | User not found/Password wrong  401 Bad Request |
| GET /profile | NaN | get profile page | profile page  200 OK | user not found  401 redirect to login page |
| POST /profile | name, address | update profile | profile page  200 OK | user not found  401 redirect to login page |
| GET /signup | NaN | get signup page | signup page  200 OK | 404 Not Found |
| POST /signup | name, email, password | signup a new user | profile page  200 OK | email already exist  401 Bad Request  redirect to profile page |

**Error handling has been implemented in our system**

We added error handling for each interface in our server .

For example:

***product*** .**find**({**category**: req.**params**.category\_id})  
 .populate(**'category'**)  
 .exec(**function** (err, products) {  
 **if** (err) {  
 res.**status**(500).render(**'error\_handle/errorpage'**,  
 {  
 **error**:err,**message**:**"Internal Server Error"**,  
 **request**:**'get /products/'**+req.**params**.category\_id  
 });  
 **return**;  
 }  
 res.**status**(200).render(**'product/productsOfCategory'**, {  
 **productsFound**: products  
 });  
 });

If error occurs, server will send an error page to client that shows some basic error information. Server will not crash when error occurs.

