SW Engineering CSC648/848
Milestone 4

Fall 2018

Team: 12
Project: GatorList
Date: 11/28/18
Revised Date: 12/9/18
Version: 1.0

1. Product Summary

Name of product: GatorList

Functions:

All unregistered users may:

1. Register a new account
2. Log into an existing account (and logout afterwards)
3. Use the search engine to see the store's items
4. Change the search category to filter search results
5. Fill out the sell item post form
    a. Add from 1 to 4 images
    b. Add up to 500 words for description
    c. Add a price
    d. Are unable to submit the item post form for review until they log into or register for a registered account
6. Visit the different links (i.e. Home, Sell, About Team, Register, Login)

All registered users may:

1. Log out from their account
2. Use the search engine to see the store's items
3. Change the category to filter search results
4. Fill out the sell item post form
    a. Add from 1 to 4 images
    b. Add up to 500 words for description
    c. Add a price
    d. Submit the item post form for review
5. Visit the different links (i.e. Home, Sell, About Team, Register, Login)
6. Visit their user dashboard page
    a. See their item posts and statuses
    b. Delete their item posts
7. Message another user regarding an item

Admins may:

1. Log out from their account
2. Use the search engine to see the store's items
3. Change the category to filter search results
4. Fill out the sell item post form
    a. Add from 1 to 4 images
    b. Add up to 500 words for description
    c. Add a price
    d. Submit the item post form for review
5. Access the admin dashboard
    a. View posts that need review
    b. Approve or deny an item post
    c. View a list of all the registered users

6. Delete an item post
7. Delete a registered user's account

URL: https://csc648team12.herokuapp.com/

2. Usability Test Plan

The search function shall be tested for usability.

1. Test Objectives
    a. Have the user search various entries in the search bar; i.e. book, chair, a color, ect
    b. Have the user change the category for another search
    c. Have the user check whether the search entry is retained after clicking search
    d. Have the user use search while being on one of the different links/routes (Home, About Team, Sell, Register, Login)
    e. Have the user type specific search entries; i.e. Use non-alphabetic characters
2. Test Plan

    This test plan is intended to see whether any person may use the search function with ease and check whether the search function is working correctly. To highlight any abnormalities with the search function, we shall instruct the user to search various entries, change the categories, and confirm whether the entries are retained after using search or clicking the other routes of the website. Lastly, to prevent injection or security risks, we will need to confirm that non-alphabetic characters in search queries are removed. These are the core elements of the search function. The search function is expected to be simple and straightforward. By having the user test all of these scenarios, we will have a better grasp of its usability.

    Task instructions for test user:

    1. Search for "book"
    2. Change category to "home" and search for "book"
    3. Change category to "All Categories" and search for "$%^book"
    4. On the Sell page, search for "{&}tv"
    5. On the About Team page, search for "desk*#@!"
    6. On the Register page, change category to "book" and search for "^^^book"
    7. On the Login page, change category to "all categories" and search for "(*)batter&&"

3. Questionnaire

Please check the statements that correspond to your experience with the search function.

You had trouble using the search function

❏ Strongly Disagree
❏ Disagree
❏ Neutral
❏ Agree
❏ Strongly Agree

You had trouble searching for all the specified terms

❏ Strongly Disagree
❏ Disagree
❏ Neutral
❏ Agree
❏ Strongly Agree

The non-alphabetic characters affected your search results negatively

❏ Strongly Disagree
❏ Disagree
❏ Neutral
❏ Agree
❏ Strongly Agree

You had trouble changing the search categories

❏ Strongly Disagree
❏ Disagree
❏ Neutral
❏ Agree
❏ Strongly Agree

You had trouble retaining the search entries and categories you selected after executing a search

❏ Strongly Disagree
❏ Disagree
❏ Neutral
❏ Agree
❏ Strongly Agree

The search results were relevant to the search's category

❏ Strongly Disagree
❏ Disagree
❏ Neutral
❏ Agree
❏ Strongly Agree

3. QA Test Plan

QA Test Objectives

Make sure that search functionality works as expected, there is no security issues related to unexpected strings as user inputs and that items displayed are correct.

HW and SW setup

We are testing on MacBook Pro with macOS 10.14.1. Used browsers are Google Chrome 70.0.3538.110 and Mozilla Firefox Quantum 63.0.3.

URL: https://csc648team12.herokuapp.com/

Feature to be tested

Just as in the usability test plan, we are going to design the QA test plan for Search.

QA Test Plan

Test #	Title	Description	Input	Expected Correct Output	Results
1	Search with category	Search for an item and filter the results to only show “Electronics”	Type into search field: “Laptop” and change category to “Electronics”	Shows only 1 result (Laptop)	PASS PASS
2	Search for similar items	Search for an item with an incomplete string	Type into search field: “Phy” and change category to “All Categories”	Shows only 1 result (Physics Book)	PASS PASS
3	Search for numerical string	Search for a literal number	Search: “1”	Shows 9 results (current number of items in database is 9)	PASS PASS

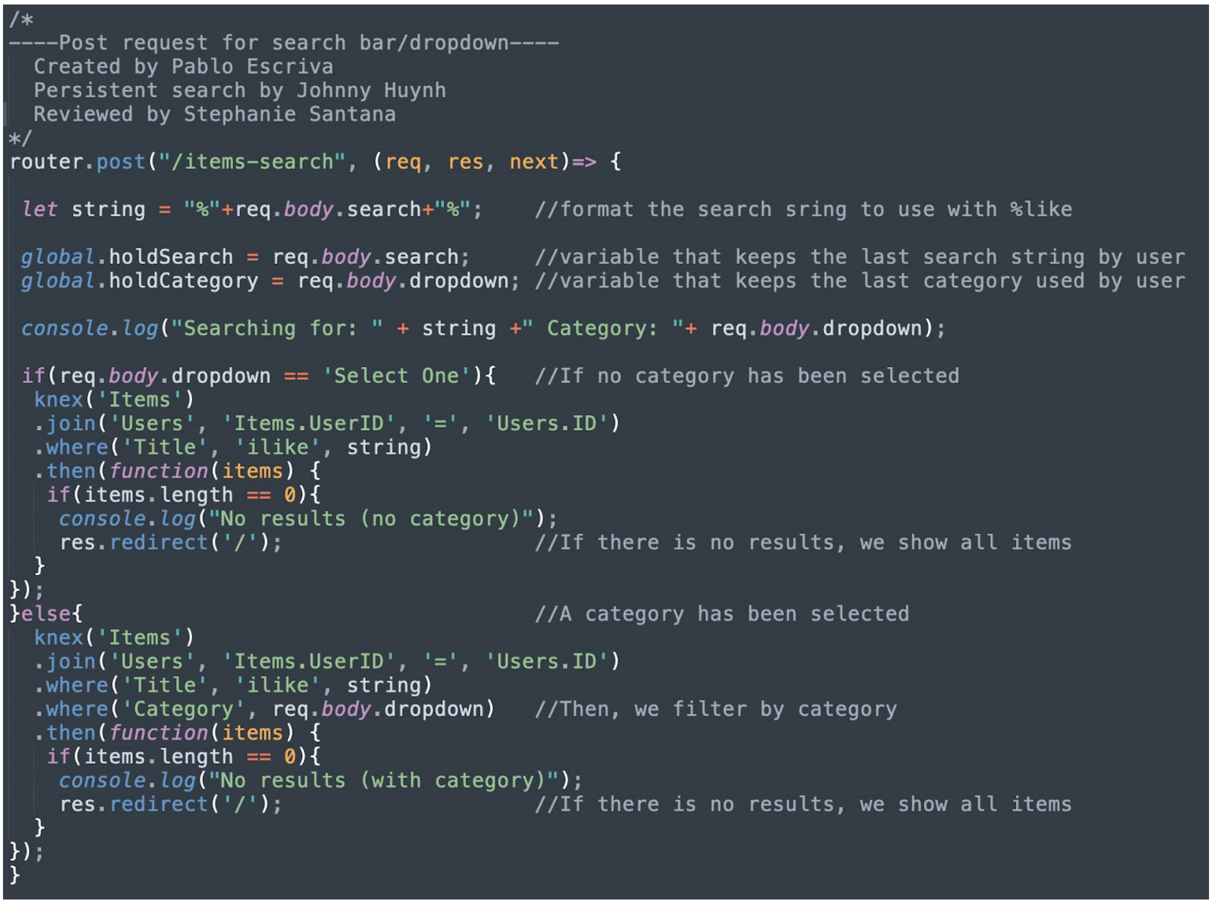
4. Code Review

Coding Style

We try to write self-explanatory, we name the variables in a way that makes it easy to understand what we use them for. We use ES6 standard for javascript (let -> var, arrow functions...). We did not use header comments but we are working on this.

We only add code to Developer branch via pull request. Once there is a new feature that a team member has finished a new pull request needs to be created. Then, someone else pulls that changes and tests it locally. If everything works, the pull request is accepted and the changes are added to Developer. We only push to master before every milestone and we freeze the branch, master always has a working version of our page.

Code example



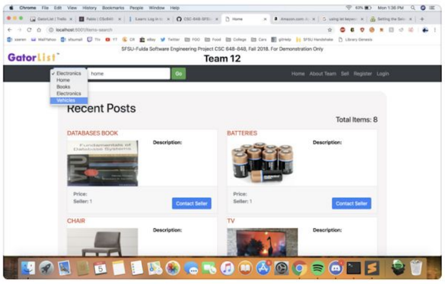
```
/*
----Post request for search bar/dropdown----
  Created by Pablo Escriva
  Persistent search by Johnny Huynh
  Reviewed by Stephanie Santana
*/
router.post("/items-search", (req, res, next)=> {

 let string = "%"+req.body.search+"%";    //format the search sring to use with %like

 global.holdSearch = req.body.search;     //variable that keeps the last search string by user
 global.holdCategory = req.body.dropdown; //variable that keeps the last category used by user

 console.log("Searching for: " + string +" Category: "+ req.body.dropdown);

 if(req.body.dropdown == 'Select One'){   //If no category has been selected
  knex('Items')
  .join('Users', 'Items.UserID', '=', 'Users.ID')
  .where('Title', 'ilike', string)
  .then(function(items) {
   if(items.length == 0){
    console.log("No results (no category)");
    res.redirect('/');                    //If there is no results, we show all items
  }
});
}else{                                    //A category has been selected
  knex('Items')
  .join('Users', 'Items.UserID', '=', 'Users.ID')
  .where('Title', 'ilike', string)
  .where('Category', req.body.dropdown)   //Then, we filter by category
  .then(function(items) {
   if(items.length == 0){
    console.log("No results (with category)");
    res.redirect('/');                    //If there is no results, we show all items
  }
});
}
```

Peer review

Pablo 12:19 PM
Hi Johnny, persistent search is working for search bar but not for categories, could you check Developer branch to see if you find the bug?

Johnny Huynh 12:33 PM
yeah ill look into it now

Johnny Huynh 1:36 PM
Pasted image at 2018-11-05, 1:36 PM ▾



i have it half correct so far

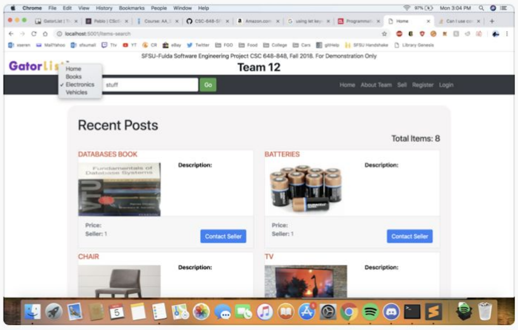
im unsure how to make the checkmark stay on the selected category

the given code block doesnt give me any hints on how to change it

from the dropdown menu, the first line holds the current category by name only, then the for loops lists all categories

[current category name]
[books]
[electronics]
[home]

3:05 PM Pasted image at 2018-11-05, 3:05 PM ▾



nevermind, got All Categories included now too. ill make a new branch with the changes now for review

Pablo 3:42 PM
good! thank you

Johnny Huynh 3:42 PM
it's in VP-ConsistentCategory

Pablo 3:43 PM
Good! I'll check it out

Ok works perfect now, I have merged it

thanks Johnny!

5. Self-check on best practices for security

Major protected assets:
Registered user information: username, first name, last name, and password.
Item information: images, description, and price.

PW encryption:
on track

Input data validation:
on track

6. Self-check: Adherence to original Non-functional specs

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO). **DONE**
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. **DONE**
3. Selected application functions must render well on mobile devices. **DONE**
4. Data shall be stored in the team's chosen database technology on the team's deployment server. **DONE**
5. No more than 50 concurrent users shall be accessing the application at any time. **DONE**
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **DONE**
7. The language used shall be English.**DONE**
8. Application shall be very easy to use and intuitive.**DONE**
9. Google analytics shall be added **DONE**
10. No e-mail clients shall be allowed **DONE**
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated. **DONE**
12. Site security: basic best practices shall be applied (as covered in the class) **ON TRACK**
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development **DONE**
14. The website shall prominently display the following exact text on all pages *"SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application). **DONE**