SW Engineering CSC648/848 Spring 2024
CSC 648/848 Milestone 4: QA and Usability Testing
Application Title: GatorMarket
Date: 5/22/2024

Meet The Team

- ➔ Jackson Hill, jhill@sfsu.edu
- ➔ Maxwell Lewis, mlewis13@mail.sfsu.edu
- ➔ Jose Rios, jrios7@sfsu.edu
- ➔ Javi Buenrostro, jbuenrostro@mail.sfsu.edu
- ➔ Rene Antoun, rantoun@sfsu.edu

History

Date Submitted	Date Revised
5/22/2024	

Milestone 4

Team 2 - GatorMarket

1. Product Summary

GatorMarket is a specialized online marketplace designed to serve the San Francisco State University (SFSU) community, including faculty, students, and staff. The platform is meticulously tailored to the unique needs of the academic environment, facilitating the easy purchase and sale of a wide array of products relevant to educational and everyday needs. Users can browse, search, and filter products ranging from the latest electronics to essential textbooks and diverse teaching materials, ensuring the available options fit their budget and requirements. Unique features such as the ability to search by class number and a "meetup" feature, which integrates a campus map and real-time weather conditions, significantly enhance the user experience by simplifying the logistics of transactions and ensuring both safety and convenience. Furthermore, GatorMarket is committed to fostering a community-centric marketplace by providing a secure platform where users can exchange goods with trust and ease. This platform also encourages sustainable practices by facilitating the sale of used textbooks and other reusable materials, reducing waste and promoting environmental consciousness within the campus community. With a focus on user-friendliness, GatorMarket includes features like detailed product descriptions, user ratings, and direct messaging capabilities that empower users to make informed purchasing decisions and maintain clear communication channels. By offering these comprehensive services, GatorMarket aims to become an indispensable resource within the SFSU community, enhancing campus life by making shopping for school supplies and personal items more accessible and efficient

At GatorMarket, we offer a seamless and user-centric shopping experience tailored for all users, from visitors to admins, each with distinct privileges and capabilities to enhance their interaction with the platform. Non-registered users can freely browse and view product details, add items to their cart, proceed to checkout, and even register as a user. They also have the unique ability to search products by class number, which is particularly useful for academic materials, and can reach out to support for any assistance required.Once registered, users gain additional capabilities such as logging in and out of the platform, posting product listings, and messaging other users for queries or transaction details, fostering a dynamic marketplace environment.Admin users inherit all the privileges of registered users but with the added responsibility of verifying product reviews to ensure their authenticity and reliability. This role is crucial in maintaining the trustworthiness of the platform.

What sets GatorMarket apart is its specialized search functionality, allowing users to find textbooks and academic materials by class number. This feature directly caters to the academic community, aligning the shopping process with students' schedules and needs, making it a uniquely valuable resource for their educational journey.

List of Major Committed Functions

- Non-registered Users
    1. Shall be able to browse products
    2. Shall be able to view product details
    3. Shall be able to add items to the cart
    4. Shall be able to proceed to checkout
    5. Shall be able to register
    6. Shall be able to search by class number
    7. Shall be able to contact support
- Registered Users
    8. All of the requirements of non-registered users
    9. Shall be able to login/logout
    15. Shall be able to post listings of products to sell
    16. Shall be able to message sellers and buyers
- Admin
    17. All of the requirements of registered users
    19. Shall be required to verify product reviews from registered users

URL:http://ec2-18-224-202-27.us-east-2.compute.amazonaws.com/

2. **Usability Test Plan**

Search usability test plan

- **Test objectives**
  - Ease of use, to gauge how easy searching is for users
  - Reliability, the search consistently working
  - Responsive, to gauge how fast searching is
  - Efficiency, to figure out if it takes too many clicks to do simple tasks on the site
  - Accurate, to figure out if the search needs any algorithms added or changed to make it more accurate if need be
  - Satisfaction, to gauge how satisfactory the search is for users to use
- **Test background and setup**
  - System setup
    - Use one of the approved browsers for your computer and use the link below under URL to proceed. Follow the test plan provided in the QA test plans in the table below.
  - Who's the audience
    - The audience to be tested are SFSU students and faculty, faculty involving older SFSU faculty but younger ones are included
  - URL
    - http://ec2-18-224-202-27.us-east-2.compute.amazonaws.com/
  - Test environment
    - ***(you guys choose with a comment)*** **we may need to say in a classroom with someone to monitor them just to count completed searches or we can keep the one I said below**
    - can be done at home, with no cameras present or monitoring, with untrained individuals (not shown how to use the search by one of us), the focus being SFSU students and faculty
- Plan for Effectiveness
  - We could have the testers try out the search and have them note if they found the item they searched for, leaving room for any comments about missed searches on some form that then calculates a %number of completed searches and the other two columns for errors and comments
- Plan for Efficiency
  - Either the testers note down the time it took them to complete a search (successful one preferable) or a monitor does it to then be compiled into an average time taken to complete the tasks.
  - Have some algorithm to count the clicks or have a monitor there to note them down along with the pages taken to complete a search which will then be averaged out into an average for both
- Plan for Evaluation of User Satisfaction

    - Usability task descriptions
        - Search for an item
            - Find item searched
            - Report time taken
        - Sort a search
            - Sort your search
            - Find item searched
            - Report time taken
        - Go to the dashboard
            - Find your messages
            - Report time taken
- Likert scale evaluation entries
    - will calculate the standard deviation once we get all the data

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Searching was easy to do					
Space for comments					
Sorting helped me find my item faster					
Space for comments					
It was easy to access my dashboard					
Space for comments					

3. QA Test Plan and QA Testing

- **Test Objectives:**
  - Test for application to operate with at least 2 different browsers.
  - Test for successful #LIKE functionality using the search bar at the top of the homepage.
- **HW and SW Setup:**
  - URL: http://ec2-18-224-202-27.us-east-2.compute.amazonaws.com/
  - Use one of the approved browsers for your computer and use the link above to proceed. Follow the test plan provided in the QA test plans in the table below.
- **Feature to be tested:** Search Functionality

Browser: Google Chrome

#	Title	Desc	Input	Output	Results
1	Search by Category	Test % like In search with category	Select the “Books” category, and leave the search bar empty.	Check that you get 3 results, **with images resembling a book**.	**PASS**
2	Search by Term	Test % like In search with term	Select the “All” category, and enter “game”.	Check that you get 2 results, **Wooden Chess Set** and **Gaming Console**.	**PASS**
3	Searby by Category and Term	Test % like In search with both category and term	Select the “Electronics” category, and enter “headphones” in the search bar.	Check that you get one result, **Noise-Canceling Headphones**.	**PASS**

Browser: Opera GX

#	Title	Desc	Input	Output	Results
1	Search by Category	Test % like In search with category	Select the “Books” category, and leave the search bar empty.	Check that you get 3 results, **with images resembling a book**.	**PASS**
2	Search by Term	Test % like In search with term	Select the “All” category, and enter “game”.	Check that you get 2 results, **Wooden Chess Set** and **Gaming Console**.	**PASS**
3	Searby by Category and Term	Test % like In search with both category and term	Select the “Electronics” category, and enter “headphones” in the search bar.	Check that you get one result, **Noise-Canceling Headphones**.	**PASS**

4. Peer Code Review

RA

Rene Antoun
To: Maxwell Lewis

Sat 5/18/2024 1:01 AM

Hi Max,

I hope all is well with you. I'm reaching out because I've been working on refining the database query functionality of our web application, and I could really use your insight. This particular piece of code is vital as it directly affects both the performance and user experience of our app. Given its importance, I'd love to get your perspective to ensure everything runs smoothly for our upcoming Web Application and usability tests.

Here's the code snippet I'm focusing on:

```
database.query(query, (err, result) => {

  if (err) {
    req.searchResult = "";
    req.searchTerm = "";
    req.category = "";
    next();
  }

  req.searchResult = result;
  req.searchTerm = searchTerm;
  req.category = category;

  next();
});
```

This section is crucial because it handles how our application responds to and recovers from database errors,something that directly impacts how users experience our service. I'm particularly interested in your insights on several specific aspects:

1. **Error Handling Efficiency**: How effectively does the code manage errors? Are there any potential edge cases that I might have missed which could cause unexpected behavior or failures?
2. **Code Clarity and Maintainability**: Is the code easy to understand at first glance? Could the readability be improved through better naming conventions or more detailed comments?
3. **Performance Optimization**: Do you see any opportunities to optimize the code for better performance? For instance, are there any redundant operations or possible improvements in the way we handle the database results?

2. **Code Clarity and Maintainability**: Is the code easy to understand at first glance? Could the readability be improved through better naming conventions or more detailed comments?
3. **Performance Optimization**: Do you see any opportunities to optimize the code for better performance? For instance, are there any redundant operations or possible improvements in the way we handle the database results?
4. **Best Practices Compliance**: Does this implementation adhere to known best practices in Node.js and MySQL interactions, especially in regards to security and data handling?

Could you please take a look and let me know what you think? If there's more information you need, or if other parts of the code could help you form a better review, just let me know. Your feedback will be invaluable as we prepare for further development and testing.

Thanks so much for your time and help. I'm looking forward to your thoughts.

Best regards,

Rene

ML

Maxwell Lewis
To: Rene Antoun

Reply | Reply all | Forward

Sat 5/18/2024 4:08 AM

Why hello Mr. Antoun,

I have received the code that you have provided in the email. So far, the code you have provided is an amazing piece of work. I can 100% say that this code does fit number 4 as it does fallow the best practices compliance. For number 2 I see that it's very easy to understand from the if statement and the req functions. It seems to handle the errors very well but as a suggestion make error console for terminal reading so we can see if we are dealing with a 404 error, for example to show the error it should display something like this for example:

"res.status(500).json(error: "404 server failed to connect")" after making the proper if statement that associated and if we are using express.js extension.

This could also be used for related errors. For optimization you seem to be doing pretty well in terms of keeping it simple, fast, and easy to read. I think your on the right track but in the future please do add comments for better documentation efforts. All and all this were a very good snippet of code you have sent and wish you luck on your project.

Best regards,
Maxwell Lewis

Hi Maxwell,

Thank you for the thorough review and the positive feedback on the code snippet I sent over. I appreciate your insights and I'm glad to hear that the code aligns well with best practices and is easy to understand.

I've noted your suggestion about enhancing error handling by using more specific error messages in the console, such as implementing res.status(500).json({error: "404 server failed to connect"}). That's a great point, and I'll incorporate it into the project to improve our error reporting and make debugging easier for the team.I also acknowledge your comment on the need for better documentation through comments. Moving forward, I will ensure that each significant block of code is well-documented to make the codebase more maintainable and easier for new team members to understand.

Thanks again for your constructive feedback and encouragement. I'll make these adjustments and keep you updated on the progress. If there's anything else you'd like to add or a new area you think we should focus on, please let me know.

Best regards,

Rene

5. Self-Check on Security

Asset to be protected	Types of possible attacks	Consequence of security breach	Your strategy to protect the asset
Information System	System Confidentiality and Integrity	Financial loss, Restoring system, Communication	Grant users minimal permissions necessary and use data encryption
User Database	Data Confidentiality	Financial loss, Harm to Users	Require users to authenticate themselves
Individual User Record	MySQL attack (SQL injection)	Direct loss is low, Loss of Reputation	Validate searches

- Confirm PW Encryption in DB **[DONE]**
- Confirm Input Data Validation **[DONE]**

6. Self-Check on Adherence to Non-Functional Specs

1) Application shall be developed, tested, and deployed using tools and servers approved by Class CTO and as agreed in M0 **[DONE]**
2) Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers **[DONE]**
3) All or selected application functions shall render well on mobile devices **[DONE]**
4) Data shall be stored in the database 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 **[DONE]**
7) The language used shall be English (no localization needed) **[DONE]**
8) Application shall be very easy to use and intuitive **[DONE]**
9) Application shall follow established architectural patterns **[DONE]**
10) Application code and its repository shall be easy to inspect and maintain **[DONE]**
11) Google Analytics shall be used **[DONE]**
12) No e-mail clients shall be allowed. Interested users can only message sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application **[DONE]**
13) Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **[DONE]**
14) Site security: basic best practices shall be applied (as covered in the class) for main data items **[DONE]**
15) Media formats shall be standard as used in the market today **[DONE]**
16) Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools **[DONE]**
17) The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only" at the top of the WWW page Nav bar. (Important so as to not confuse this with a real application). **[DONE]**

7. Use of genAI tools

There has been no use of ChatGPT or genAI for this milestone.