SW Engineering CSC648/848 Spring 2022 Section 2

19 April 2022

Milestone 4 – Beta Launch, QA, and Usability Testing and Final Commitment for Product Features

Team 2

Team Lead: Zubin Kanga (zkanga@sfsu.edu)

Front End: Cat Tuong Vu, Gurinder Singh, Sebastian Wcislo

Back End: Anudeep Katukojwala, Brandon Butler, Zubin Kanga

GitHub Master: Sebastian Wcislo

History Table

Submitted	19 April 2022
Revised	

Product Summary

Tech Connect

http://ec2-54-183-169-123.us-west-1.compute.amazonaws.com/home

Final Functions

1. Search for jobs while also being able to filter based on tech area, job positions, and skills
2. Companies can post in 9 different tech areas and specify if the job is active or inactive.
3. Posts up to date tech trends.
4. Allows SFSU students and recruiters to register and login.
5. Notification for registered users when a job matching their preferences gets posted.
6. Administrator capabilities to trigger notifications to corresponding users.
7. Email confirmations
8. Resume review and uploads

Usability Test Plan

Test Objectives:

We are testing the search functionality, specifically we are testing the filters that a user can specify, and those filters are used to query the database. We want users to be able to select a filter and type some text relating to that filter. The reason why we are testing this feature is that since students would want to look up job opportunities based on their interest, location, because of which search is an essential part of our website and need to be tested for usability. So, let's say a user wants to search based on tech areas, so they can choose the tech area as a filter and then type the tech area that they want to know more about in the search field. Another example would be of a user who wants to view jobs that are available in a particular city so they can use the city filter and enter the city name in the search field. We also have the option for users to search based on the job type, so if a user wants to look up jobs that match their job type, they can select the job type filter and enter the job type in the search field. After a user selects the filter and enters their search query, we display the results matching their queries on another page. We also want to ensure that we are only showing the relevant search results.

Test Background and Setup:

We want to test the search functionality for students, so SFSU students interested in tech are the intended users that will be using our search functionality. We want to give the opportunity to search for jobs based on their interests, location. That's why we want to test the search functionality.

The search functionality we want to test can be found at the following url: http://ec2-54-183-169-123.us-west-1.compute.amazonaws.com/TestHomePage. This url contains our search with a text

input field and a drop down which acts as a filter. Our focus is user satisfaction evaluation, so we want to measure how comfortable and easy to use our search feature is

Usability Task Description:

1. Select the Tech area Filter
2. Enter a tech area like “Artificial Intelligence”
3. Submit the search
4. Scroll through the search results

In order to measure effectiveness, we would have to ensure that users are able to select a filter that they want to apply for example, tech area. Then we would want to make sure that users can type the tech area they are interested in for example, Artificial Intelligence. Then we would want users to be able to submit their search and go to the result page that only shows the relevant search results.

In order to measure efficiency, we would want to measure the average time it takes to select a filter. Then we would want to measure how many clicks it takes to select a filter, which will help us determine how much effort is spent while selecting a filter.

Lickert subjective test:

		Strongly Disagree			**Strongly Agree**	
	Question	**1**	**2**	**3**	**4**	**5**
1	**It was easy to select a tech area filter**					
2	**It was easy to enter search query in the search input field**					
3	**It was easy to submit the search**					

QA Test Plan

Test Objectives:

The objectives are to make sure that our user's searches are accurate and relevant to what they are looking for.

HW and SW Setup:

To setup out hardware we simply needed to get a working computer or mobile device, preferably both. This is to give us a wide array of hardware that our users may use to access our site. Once we have gotten the proper devices, we need to ensure that they have installed many different web browsers such as Internet Explorer, Safari, Firefox etc. This is to also ensure that we are test many different browsers since our users may be using many different browsers.

Feature to be Tested:

Search functionality

QA Test plan table:

Test#	
Test Title	
Description	
Input	
Expected Output	
Result	

Code Review – Anudeep Katukojwala

Coding Style:

A. Naming Convention

   a. Variable names will be written in Camel case. Underscores are allowed in certain conditions when readability is in question.

B. Line Length

   a. Lines should not be longer than 80 characters

C. Indentation

   a. Four spaces should be used as the unit of indentation. Tabs must be set exactly to 4 spaces.

D. Comments

   a. End of line comments are preferred over block comments. But if the written code demands longer comments, one can block comments.

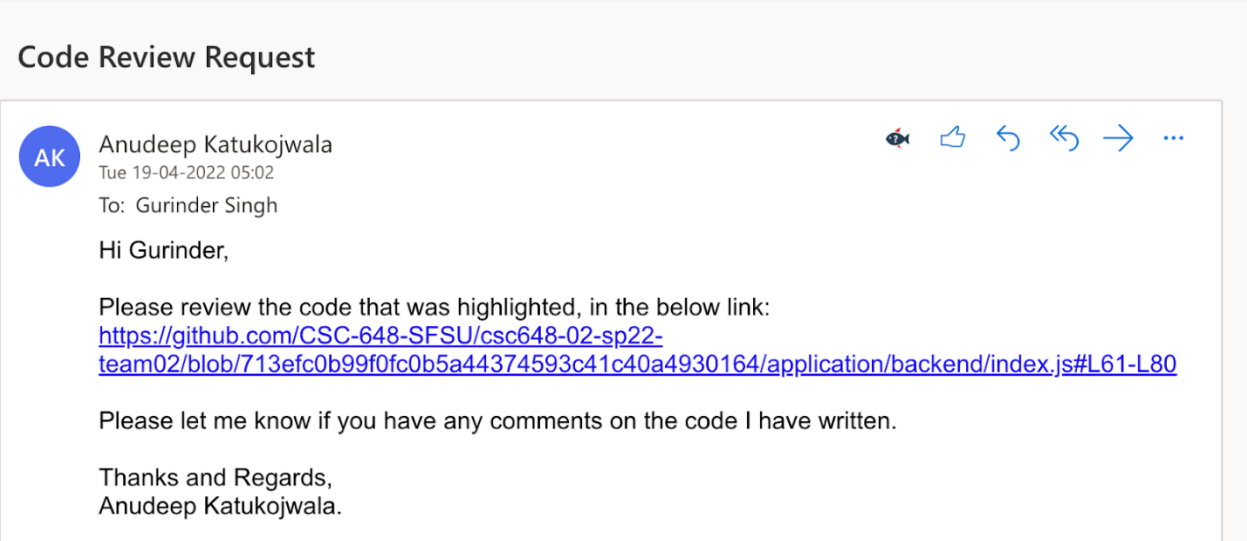
Code Review Example:

Code that is requested to review

https://github.com/CSC-648-SFSU/csc648-02-sp22-team02/blob/713efc0b99f0fc0b5a44374593c41c40a4930164/application/backend/index.js#L61-L80

```
61  //Here we finalize what query should be used based on the user input provided by frontend
62  if (!request.query.searchTerm) {
63    console.log("No Input");
64    queryName = queryFile.finalQuery.emptySearch;
65  } else if(request.query.searchTerm && request.query.category === "all") {
66    console.log("User entered search parameter and have selected the dropdown as ALL");
67    queryName = queryFile.finalQuery.searchString;
68  } else if (request.query.category === "jobType") {
69    console.log("User selected Job type");
70    queryName = queryFile.finalQuery.type_of_job;
71  } else if (request.query.category === "city") {
72    console.log("User selected City");
73    queryName = queryFile.finalQuery.job_city;
74  } else if (request.query.category === "tech_area") {
75    console.log("User selected techArea");
76    queryName = queryFile.finalQuery.tech;
77  }
78
79  console.log("Query used:" + queryName);
80
```

Code Review Request Email



Code Review Request

AK

Anudeep Katukojwala
Tue 19-04-2022 05:02
To: Gurinder Singh

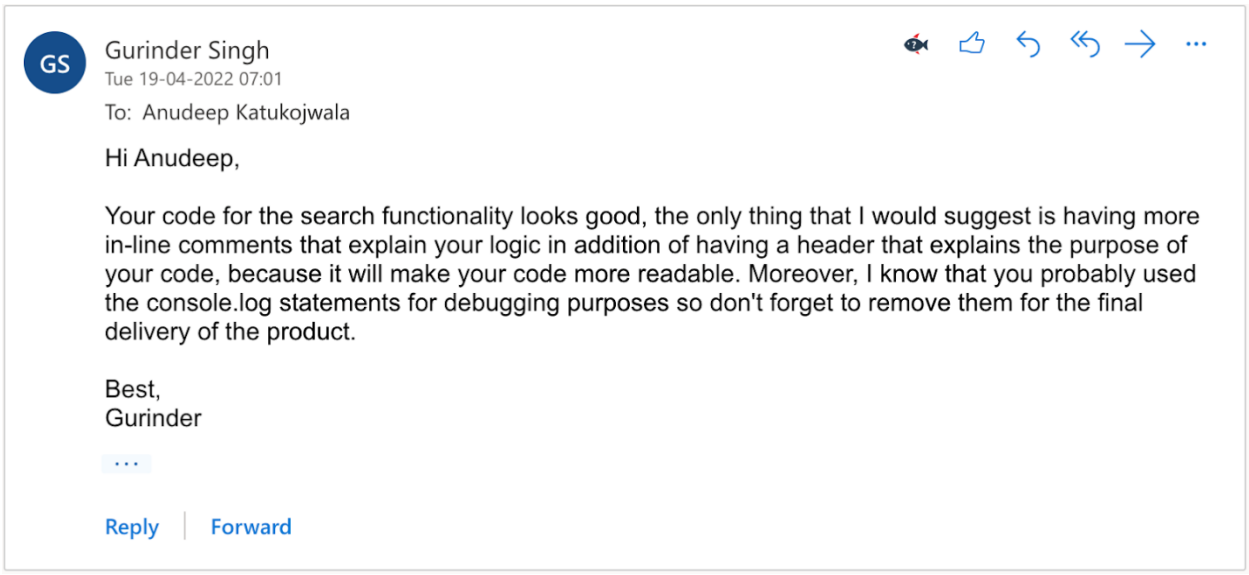
Hi Gurinder,

Please review the code that was highlighted, in the below link:
https://github.com/CSC-648-SFSU/csc648-02-sp22-team02/blob/713efc0b99f0fc0b5a44374593c41c40a4930164/application/backend/index.js#L61-L80

Please let me know if you have any comments on the code I have written.

Thanks and Regards,
Anudeep Katukojwala.

Comments by reviewer



GS

Gurinder Singh
Tue 19-04-2022 07:01
To: Anudeep Katukojwala

Hi Anudeep,

Your code for the search functionality looks good, the only thing that I would suggest is having more in-line comments that explain your logic in addition of having a header that explains the purpose of your code, because it will make your code more readable. Moreover, I know that you probably used the console.log statements for debugging purposes so don't forget to remove them for the final delivery of the product.

Best,
Gurinder

...

Reply | Forward

Self-Check on best Practices for Security

Major assets to be protected:

1. Emails
    a. Part of the instances
2. Passwords
    a. Encryption of passwords in the database
3. SQL Query
    a. Can only be accessed by us.
4. Servers
    a. Ssh access keys and managed by github.
5. Instances
    a. Part of the servers

Confirmation that password encryption:

```
INSERT INTO `basicDev`.`company` (`name`, `passwordHash`, `description`) VALUES ('Wilkerson Ltd', 'pO8HyjvXCEOQYUzmGPk8YQ==', 'Synchronized solution-oriented core');
Actual password: Wilkerson Ltd
INSERT INTO `basicDev`.`company` (`name`, `passwordHash`, `description`) VALUES ('Smith, Harris and Bates', 'njVqyzqT87Br2lXFJFz0b3CPGSGZHHG1iwbsUl5WF5g=', 'Multi-tiered uniform open system
Actual password: Smith, Harris and Bates
INSERT INTO `basicDev`.`company` (`name`, `passwordHash`, `description`) VALUES ('Scott, Trujillo and Chen', 'h/deNWJEV7dmEy+9ZmmaQN20öwvdyPNP1qWAyHtUM8I=', 'Stand-alone zero tolerance capa
Actual password: Scott, Trujillo and Chen
INSERT INTO `basicDev`.`company` (`name`, `passwordHash`, `description`) VALUES ('Quinn PLC', 'J/EY8OQGXve/fw47cvkx1w==', 'Managed systemic alliance');
Actual password: Quinn PLC
INSERT INTO `basicDev`.`company` (`name`, `passwordHash`, `description`) VALUES ('Smith-Allen', 'X3dbLUn9g/1uMa8aEnc8PA==', 'Centralized web-enabled challenge');
Actual password: Smith-Allen
```

Confirmation of Input data validation:

```
if (freeSearch.length <= 40) {
  if (freeSearch.match(/^$|^[0-9a-zA-Z ]+$/)) {
    console.log(
      "Search term length is less than or equal to 40 and all characters are alphanumeric"
    );
    validRequest = true;
  } else {
    console.log("Entered search term is not alphanumeric");
    alert("Please ensure that the search term is alphanumeric");
  }
} else {
  console.log("Search term length is greater than 40 characters");
  alert("Please ensure that the search term's length is <= 40 characters");
}
```

Self-Check Adherence to Original Non-Functional Specs – Performed by Team Leads

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).
    - Frontend: Complete
    - Backend: Complete
2. Application shall be optimized for standard desktop/laptop browsers e.g., must render correctly on the two latest versions of two major browsers.
    - Frontend: Complete
    - Backend: Complete
3. Selected application functions must render well on mobile devices
    - Frontend: Work in Progress (On Track)
    - Backend: Complete
4. Data shall be stored in the team's chosen database technology on the team's deployment server.
    - Frontend: Complete
    - Backend: Complete

5. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
    - Frontend: Work in Progress (On Track)
    - Backend: Work in Progress (On Track)
6. The Language used shall be English.
    - Frontend: Complete
    - Backend: Complete
7. Application shall be very easy to use and intuitive.
    - Frontend: Work in Progress (On Track)
    - Backend: Work in Progress (On Track)
8. Google maps and analytics shall be added
    - Frontend: Work in Progress (On Track)
    - Backend: Work in Progress (On Track)
9. No email client shall be allowed. You shall use webmail.
    - Frontend: Work in Progress (On Track)
    - Backend: Work in Progress (On Track)

10. Pay functionality, if any paying for goods and services shall not be implemented nor simulated in UI.

    - Frontend: Work in Progress (On Track)
    - Backend: Complete

11. Site security

    - Frontend: Work in Progress (On Track)
    - Backend: Work in Progress (On Track)

12. Modern SE processes and practices shall be used as specified in class. This includes collaborative and continuous SW development.

    - Frontend: Work in Progress (On Track)
    - Backend: Work in Progress (On Track)

13. The website shall prominently display the following exact text on all pages: “SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration only”.

    - Frontend: Work in Progress (On Track)
    - Backend: Work in Progress (On Track)