CSC 648/848 Software Engineering – Fall 2022
Milestone 4

JobHunter

Submitted by:

Team 04
Muzaffar Sharapov - Team Lead
Jiayi Gu - Front End Developer
Julian Manaois - Database & Github check
Chengkai Yang - Front End Developer
Yagiz Saatci - Back End Developer

Revision History

Nov 1, 2022	Document initialized
Nov 6, 2022	Update URL, other sections
Nov 8, 2022	Update QA test

Table of Contents

1. Product Summary

Application name: JobHunter

URL: ec2-54-67-53-87.us-west-1.compute.amazonaws.com

1. Have a working application to help SFSU students to find their dream jobs
2. The web application must allow Tech companies to post jobs in 9 areas.
3. It will allow us to search & filter jobs based on tech areas, job positions, and skills.
4. Users can get alerts for matching job interests, and Users can turn off alerts if needed.
5. The site will have administrator capabilities to trigger the matching alerts
6. The site will have login and sign-up pages for users.
7. Each user will have a profile page.
8. The application will focus on desktop/laptop browsers
9. The site will utilize a user database set for storing all user information and postings.
10. Recruiters will be able to view profiles and post job listings
11. Users can change or edit their search on the same page.
12. Users can search job listings without registering or logging into their account.
13. User credentials will be encrypted.
14. Users can navigate the Helpful Links page without an account.
15. Users can apply directly for job listings without filling out a third party application.
16. Users can delete their account.

Our app is unique from other competitors because you do not need an account to search for listings, and has a helpful links page that can help job-searchers with tips for interviewing and other skills.

2. Usability Test Plan

Test objectives: We will be testing the Search & filter function of our website for functionality and cross-platform usability. We are testing this specific function because it is the main feature of our site and the most important. This feature allows users to search for specific jobs and the site has no use otherwise.

Test background and setup:

System will be set up as multiple browsers running the URL of our website. We will test on two browsers, Google Chrome and Safari.

The intended users will be users who are searching for specific job titles or fields on our website.

The starting point will be our website's home page, which is the URL provided: ec2-54-67-53-87.us-west-1.compute.amazonaws.com

For this usability test, we will test user satisfaction and efficiency to the desired output.

Usability Tasks

Task	Description
Task	Search 'Engineer'
Machine State	Default home page
Successful completion criteria	Search results containing 'engineer' appear
Benchmark	<1 minute

Task	Description
Task	Filter search ‘Web dev’
Machine State	Default home page
Successful completion criteria	Search results containing ‘web dev’ category appear
Benchmark	<1 minute

To measure effectiveness, we will ask if the UI is easy to use, if results are easy to read, and if they are easily accessible.

To measure efficiency, we will see how long it takes for the task to be completed compared to the benchmark, and see how many clicks it takes to get to desired output.

Lickert Subjective Test

1. I found the search feature easy to use.

Strongly Disagree Disagree Neither Agree Strongly Agree

2. The search results were easy to navigate through.

Strongly Disagree Disagree Neither Agree Strongly Agree

3. The search results that appeared were what I expected.

Strongly Disagree Disagree Neither Agree Strongly Agree

3. **QA Test Plan**

Test objectives: Functionality & cross-platform testing

HW & SW Setup:

Feature to be tested: Search & filter feature

Browser 1: Chrome

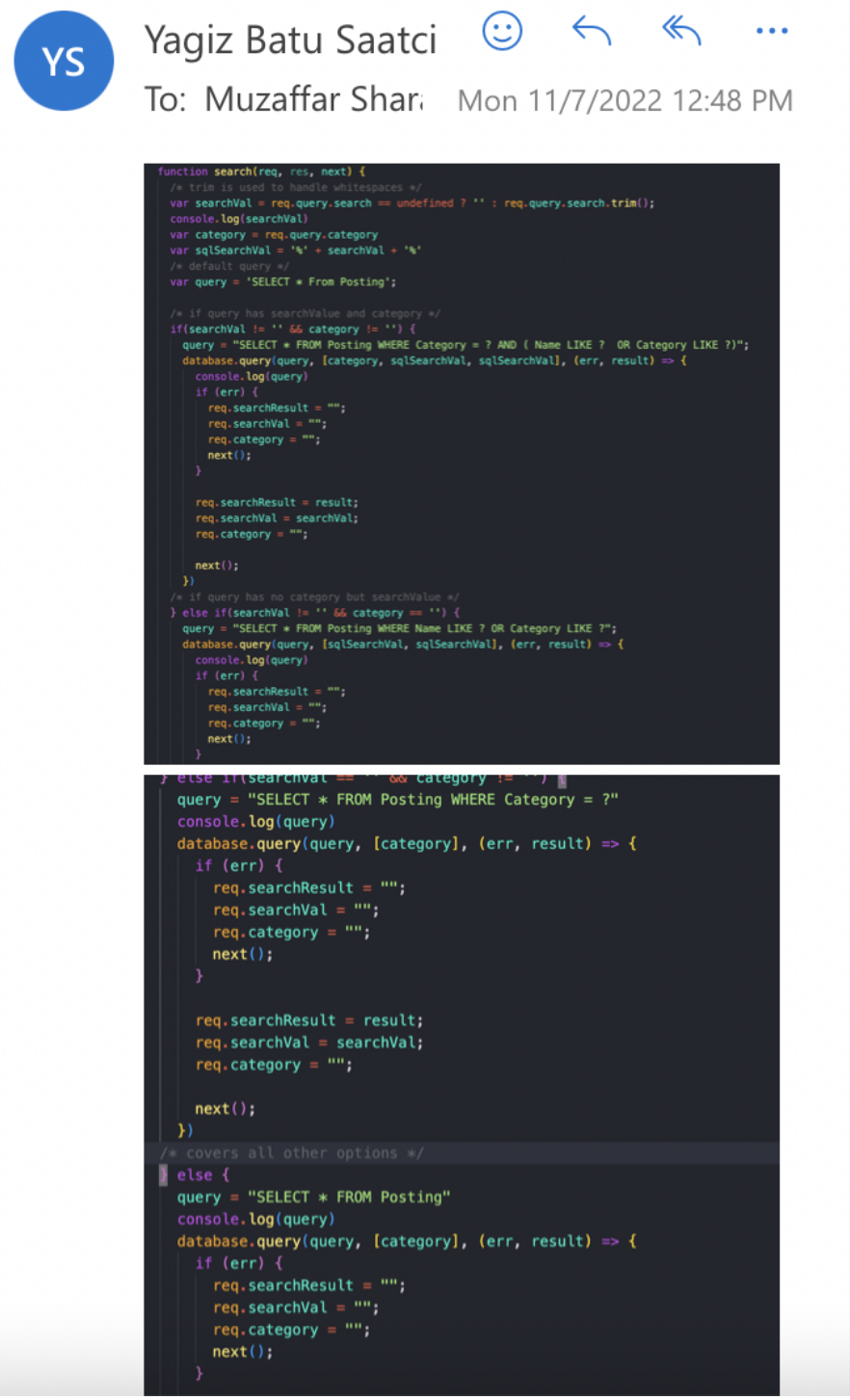
Test #	Title	Test Input	Expected Correct Output	Test Results
1	Test LIKE in search for name field	“engineer”	Get 6 results, all with “engineer” in the name field	PASS
2	Test LIKE in search for category field	Change category to “Software dev”	Get 2 results, all with “Software dev” category	PASS
3	Test LIKE in search for name field	“UX”	Get 2 results, all with “UX” in the name field	PASS

Browser 2: Safari

Test #	Title	Test Input	Expected Correct Output	Test Results
1	Test LIKE in search for name field	“engineer”	Get 6 results, all with “engineer” in the name field	PASS
2	Test LIKE in search for category field	Change category to “Software dev”	Get 2 results, all with “Software dev” category	PASS
3	Test LIKE in search for name field	“UX”	Get 2 results, all with “UX” in the name field	PASS

4. Code Review

The code style used for the application is CamelCase. Below are the screenshots of the peer code review conducted for our team.



code review

```
    req.searchResult = result;
    req.searchVal = searchVal;
    req.category = "";

    next();
  })
/* covers all other options */
} else {
  query = "SELECT * FROM Posting"
  console.log(query)
  database.query(query, [category], (err, result) => {
    if (err) {
      req.searchResult = "";
      req.searchVal = "";
      req.category = "";
      next();
    }

    req.searchResult = result;
    req.searchVal = searchVal;
    req.category = "";

    next();
  })
}
};
```

Muzaffar Sharapov

To: Yagiz Batu Saat Mon 11/7/2022 1:10 PM

Hi Yagiz,

The code looks good to me.

Best,

Muzaffar

• • •

Reply Forward

5. Self-check on best practices for security

Assets being protected:

Passwords - User passwords are encrypted in the database.

Emails - User emails are not publicly displayed. Database is not prone to SQL injection.

Input data - Input data is not prone to SQL injection or XSS. We do not have comment fields for XSS, and for SQL injection, we validate the user's information and parametrize the queries.

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

Non-functional requirement	Status
The application shall be developed, tested, and deployed using tools and servers approved by the 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 the class CTO).	Done
The application shall be optimized for standard desktop/laptop browsers, e.g., must render correctly on the two latest versions of two major browsers	Done
Selected application functions must render well on mobile devices	Done
Data shall be stored in the team's chosen database technology on the team's deployment server.	Done
The privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.	On track
The language used shall be English.	Done

The application shall be very easy to use and intuitive.	On track
Google maps and analytics shall be added.	On track
No email clients shall be allowed. You shall use webmail.	Done
Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.	Done
Site security: basic best practices shall be applied (as covered in the class)	Done
Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.	On track
The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Fall 2022. For Demonstration Only"* **at the top of the WWW page. (Important so as not to confuse this with a real application).**	On track