SW Engineering CSC648/848 Spring 2021

Project title: "Campus Cantina"
Milestone 4
Team 04

Member Name	Member Role
Rajdeep Singh	Team Lead (rsingh12@mail.sfsu.edu)
Rinay Kumar	Backend Lead
Bhavani Goruganthu	Frontend Lead / Document Editor
Frederick White	Github Master
German Perez	Frontend Team
Henzon Zambrano	Backend Team

Revisions History:

Draft Number	Date Created/Revised	Date Submitted	Instructor Comments
Draft 1	May 08 2021	May 14 2021	

1) Product Summary

Name: Campus Cantina

Product URL: www.campuscantina.com

Final P1 List:

Unregistered Users / Guest:

1. Unregistered Users shall be able to search for local restaurants based on keyword or cuisine search.
2. Unregistered Users shall be able to view details of a particular restaurant and browse through their menu.
3. Unregistered Users shall be able to add items to the cart without logging in.
4. Login/Register screen is prompted when an Unregistered user attempts to checkout.
5. Unregistered Users shall register and login with SFSU credentials to order food.

Registered Users (SFSU Customers):

6. Registered Users have the same privileges as the unregistered users and the below additional ones.
7. SFSU faculty, students and staff shall register for seeking the delivery service of 'Campus Cantina' and shall login with their credentials.
8. SFSU Customers shall be able to enter their delivery address at checkout.
9. SFSU Customers shall be able to view a list of their previous orders.

Restaurant Owners:

10. Restaurant Owners shall be able to register to advertise their services.
11. Restaurant Owners shall await approval by the admin before their restaurants can go live.
12. Restaurant Owners shall be able to add, edit or update menu items.
13. Restaurant Owners shall be able to view their profile & restaurant information.
14. Restaurant Owners shall be able to check the orders received and start preparation.

Driver:

15. Drivers shall be able to register and pick a restaurant to work for.
16. Drivers shall be able to login and check the orders assigned to them..
17. Drivers shall be able to see customer order items & Delivery Instructions.
18. Drivers shall be able to see the customer drop-off location on a map.

Administrator:

19. The site Administrator shall be required to approve the restaurants before going live on the site.
20. The site Administrator shall be able to delete or remove inappropriate users, drivers, owners.

2) Usability Test Plan

Function to be tested for Usability : Search

a) Test Objectives:

- To ensure that the search feature is properly functioning and returns valid results.
- To ensure the feature is easy to use under a variety of different possible searches.
- To ensure the feature includes results from typing terms in the search bar and/or from the accompanying cuisine drop down menu adjacent to the search bar.
- To ensure when the user hits ‘enter’ in the search bar, the search function is executed and accurate restaurant results should be displayed.
- To ensure that the number of results returned is displayed clearly.
- To ensure the results are accurately returned with %LIKE% search pattern.

b) Test background and setup

i) **System Setup:** Working release of Campus Cantina deployed to AWS and available at campuscantina.com

ii) **Starting Point:** The home page, at campuscantina.com

iii) **Intended Users:** Guests, SFSU customers, drivers, and restaurant owners.

iv) **System URL to be tested:** campuscantina.com, search bar located at top of page in navbar.

v) **Measure User Satisfaction:** Satisfaction will be measured by user responses to ease of use of search, speed of search, and quality of search results.

c) **Usability Task description:**

- Dropdown Search : Search for all restaurant cuisines
- Dropdown Search : Search for a specific cuisine
- Search Bar : Search for a restaurant with name
- Search Bar: Search for a restaurant with a keyword
- Search Bar & Cuisine Dropdown : Search using both the dropdown & a keyword

d) **Evaluation of Effectiveness:** Is the search performance fast, accurate, and easy to use.

e) **Evaluation of efficiency:** Correctness of search results, time took to generate and display results, ease of use to conduct searches.

f) **Evaluation of user satisfaction:**

Questionnaire for User Satisfaction Evaluation:

1. Search GUI is easy to use.

🔵Strongly disagree 🔵Disagree 🔵Neither agree or disagree 🔵Agree 🔵Strongly agree

2. Cuisines dropdown has options and is selectable.

🔵Strongly disagree 🔵Disagree 🔵Neither agree or disagree 🔵Agree 🔵Strongly agree

3. Keyword Search is working with hitting ‘enter’

🔵Strongly disagree 🔵Disagree 🔵Neither agree or disagree 🔵Agree 🔵Strongly agree

4. Search results are accurate and match with the keyword entered and/or the cuisine selected.

🔵Strongly disagree 🔵Disagree 🔵Neither agree or disagree 🔵Agree 🔵Strongly agree

5. There are no usability issues with the overall experience for searching a restaurant.

🔵Strongly disagree 🔵Disagree 🔵Neither agree or disagree 🔵Agree 🔵Strongly agree

3) QA Test Plan

a) **Test Objectives:** The objective of the test is to determine if the search bar and search functionality is performing correctly & returns accurate restaurant results with a variety of search combinations including the search input bar and cuisine drop down menu.

b) **HW and SW setup:** The tests will be conducted on the previous release of Campus Cantina already deployed on AWS at campuscantina.com. The tester to visit the site and perform the tests.

c) **Feature to be tested:** The feature being tested is the search bar on the top of the page in the navbar, which is a key aspect of the web application.

d) **QA Test plan**

Test #	Test Title	Test Description	Test Input	Expected Output	Test Results
1	**Search bar search for 'pizza'**	Test the keyword search functionality and the results when Search icon is clicked	Type Search term 'pizza' in the search bar and hit enter or click on search icon	Search results showing the 'Domino Hut' restaurant, with image and description should be displayed.	**PASS**
2	**Dropdown search for 'Mexican'**	Test the Dropdown search functionality and the results displayed when the Search Icon is clicked.	Click on the cuisine dropdown menu and select 'Mexican' and click on the search icon.	Search results showing three restaurant cards for 'Chipotoplay', 'Taco Shell', and 'Chapo Tacos', with accompanying images and description should be displayed.	**PASS**
3	**Keyword search for the term 'good'**	Test the %LIKE% functionality for the keyword entered in the search bar.	Enter the keyword 'good' in the search bar and hit Return or Enter key on the keyboard with the cursor on the search bar.	Search results showing two restaurant cards for 'Goodburger' and 'Goodfellas', with accompanying images and details should be displayed.	**PASS**
4	**Cuisine Dropdown & Keyword search for 'Mexican' & 'nacho' respectively**	Test the Cuisine Dropdown search functionality along with keyword search.	Click on the cuisine dropdown menu and select 'Mexican' & enter the keyword 'nacho'	Search results showing 'Chapo Tacos' restaurant with accompanying image and description should be displayed. Chapo Tacos should	**PASS**

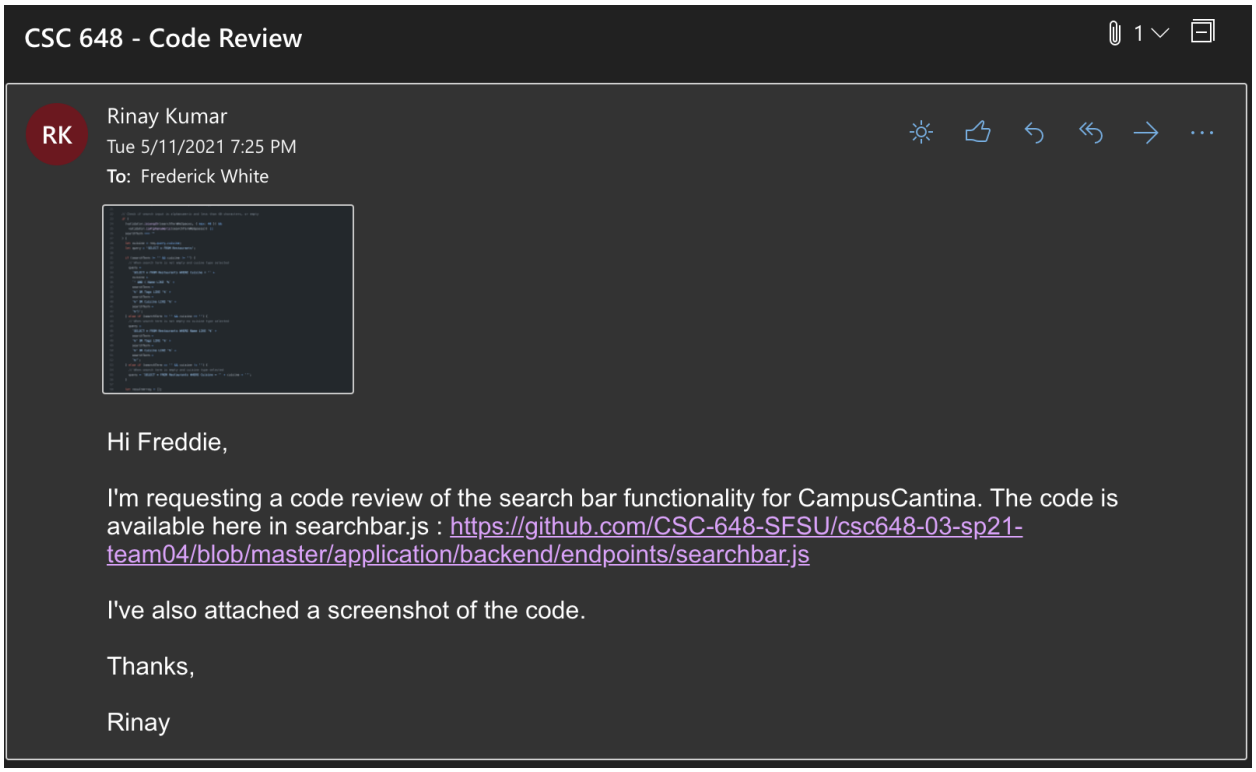
			and click on the search icon.	have a description of ‘nachos’	
5	**Invalid search term test**	Test the Search functionality with a keyword which has no results.	Enter the keyword ‘123#’ in the search bar and hit Return or Enter key on the keyboard with the cursor on the search bar.	A results page showing the text ‘0 stores nearby’ and the text ‘No Results Found’ right below it.	**PASS**

4) Code Review

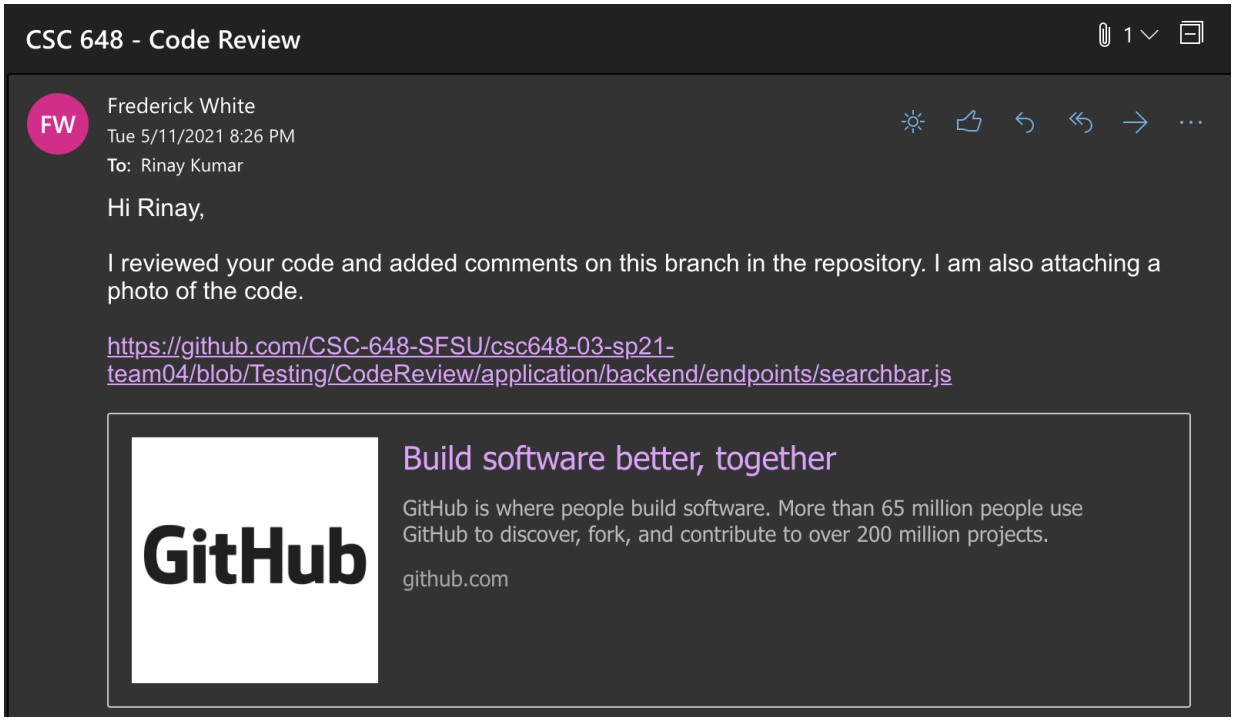
Chosen Feature: Search Bar

Peer Review screenshots:

Rinay sent an email to Freddie requesting a code review of the search bar functionality. Included in the email was a link directly to the code:



Freddie reviewed the code and replied to Rinay via email, including a link to the GitHub page where comments were added.



A key point mentioned in the comments on GitHub was to include a longer description as a header in the file.

```
/**
 * Reviewed by Frederick White
 * 05/11/2021
 * A longer description for what the file does could go here.
 */

//What functions are included in the file
/* Endpoints needed for Search bar and drop down */
```

Freddie also suggested expanding each of the comments so that it is clear what each function is doing, with required parameters and expected results.

```
/**
 * Comments for each function could be more descriptive on what it does.
 * The current comments only say what the function does
 * Comments can include what is being called, what is being returned,
 * and what parameters are required
 *
 * */
```

Commented Code:

The following is the complete code with comments from the reviewer:

```
1  /**
2   * Reviewed by Frederick White
3   * 05/11/2021
4   * A longer description for what the file does could go here.
5   */
6
7  //What functions are included in the file
8  /* Endpoints needed for Search bar and drop down */
9
10
11 const express = require('express');
12 const router = express.Router();
13 const database = require('../db'); //*What is this file for
14 const validator = require('validator'); // Used for input validation
15
16 // API call to populate cuisine drop-down list
17 router.get('/cuisines', (req, res) => {
18   database.query('SELECT * FROM Food_Cuisines', (err, result) => {
19     console.log('Called cuisines endpoint');
20     res.send(result);
21   });
22 });
23
24 // API call to search database
25 router.get('/search', (req, res) => {
26   let searchTerm = req.query.searchTerm;
27   searchTerm = searchTerm.trim();
28   let searchTermNoSpaces = searchTerm.replace(/\s+/g, '');
29
```

```
30    // Check if search input is alphanumeric and less than 40 characters, or empty
31    if (
32      (validator.isLength(searchTermNoSpaces, { max: 40 }) &&
33        validator.isAlphanumeric(searchTermNoSpaces)) ||
34      searchTerm === ''
35    ) {
36      let cuisine = req.query.cuisine;
37      let query = 'SELECT * FROM Restaurants';
38
39      if (searchTerm != '' && cuisine != '') {
40        // When search term is not empty and cusine type selected
41        query =
42          `SELECT * FROM Restaurants WHERE Cuisine = '` +
43          cuisine +
44          `' AND ( Name LIKE '%` +
45          searchTerm +
46          `%' OR Tags LIKE '%` +
47          searchTerm +
48          `%' OR Cuisine LIKE '%` +
49          searchTerm +
50          `%')`;
51      } else if (searchTerm != '' && cuisine == '') {
52        // When search term is not empty no cuisine type selected
53        query =
54          `SELECT * FROM Restaurants WHERE Name LIKE '%` +
55          searchTerm +
56          `%' OR Tags LIKE '%` +
57          searchTerm +
58          `%' OR Cuisine LIKE '%` +
59          searchTerm +
60          `%'`;
61      } else if (searchTerm == '' && cuisine != '') {
62        // When search term is empty and cuisine type selected
63        query = `SELECT * FROM Restaurants WHERE Cuisine = '` + cuisine + `'`;
64      }
```

```
    let resultArray = [];

    database.query(query, (err, result) => {
      console.log('Called search endpoint');

      // Loop through query results and add only approved restaurants to resultArray
      result.forEach((restaurant) => {
        if (restaurant.Approved === 1) {
          resultArray.push(restaurant);
        }
      });

      res.send(resultArray);
    });
  } else {
    // Send invalid as response when search term vaidation fails
    res.send('Invalid');
  }
});

module.exports = router;

/**
 * Comments for each function could be more descriptive on what it does.
 * The current comments only say what the function does
 * Comments can include what is being called, what is being returned,
 * and what parameters are required
 *
 * */
```

5) Self-check on best practices for security

PW Encryption:

All passwords are encrypted using the ‘bcrypt’ module and the hashed password is stored in the database.

Input Data Validation:

- Validated the Search bar Input field for upto 40 alphanumeric characters.
- Validated SFSU Login / Register forms for the “sfsu.edu” domain emailID.

Major Assets requiring Security:

Index	Asset to be protected	Types of possible/expected attacks	Your strategy to mitigate/protect the asset
1	User Passwords	Passwords can be breached and exposed if stored in plain text	All passwords are encrypted using the 'bcrypt' module and the hashed password is stored in the database.
2	Database Contents	SQL Injection	Validated the Search bar Input field.

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. Application delivery shall be from the chosen cloud server. **-- 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 must render well on mobile devices (specifics to be developed in consultation with users e.g. Petkovic) **-- DONE**
4. Ordering and delivery of food shall be allowed only for SFSU students, staff and faculty. **-- DONE**
5. Data shall be stored in the database on the team's deployment cloud server. **-- DONE**
6. No more than 50 concurrent users shall be accessing the application at any time. **-- ON TRACK**
7. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **-- DONE**
8. The language used shall be English (no localization needed) **-- DONE**
9. Application shall be very easy to use and intuitive. **-- DONE**
10. Application should follow established architecture patterns. **-- DONE**

11. Application code and its repository shall be easy to inspect and maintain. **-- DONE**
12. Google analytics shall be used **-- DONE**
13. No email clients shall be allowed. **-- DONE**
14. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **-- DONE**
15. Site security: basic best practices shall be applied (as covered in the class) for main data items. **-- DONE**
16. Application shall be media rich (images, maps etc.). Media formats shall be standard as used in the market today **-- DONE**
17. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development. **-- DONE**
18. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2021 For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application). **-- DONE**