CSCI-201: Principles of Software Development

Fall 2021

PA4: SalEats FrontEnd

Concepts Covered:

- HTML/CSS
- Databases/JDBC/SQL
- Java Servlets

Introduction:

Just in time! Magnificent job with the successful launch of the SalEats driver scheduling system, and as a result you have received special blessings from the director. “Perhaps it’s time to do something more modern?” Honestly, it’s not such a bad idea given the amount of success you’ve had. Don’t resist the modern call of duty - embrace it.

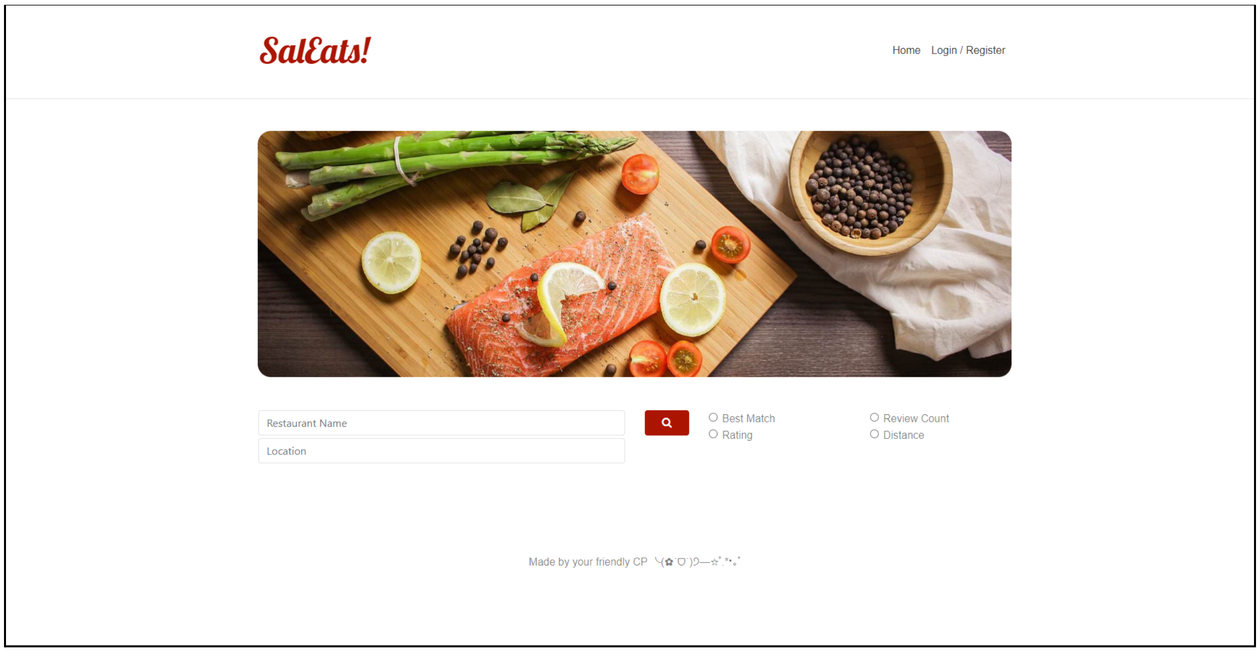
This time around, the Council of SAL is looking to expand their product line and make their new product more up-to-date and user-friendly.

In this assignment, you will be making a web application that will allow users to search for restaurants, add them to their list of favorites, and schedule reservations for these restaurants. You’ll need to use different APIs, as well as implement a database to keep track of user information. The rest of this document includes mockups of the web pages. Your web pages should look reasonably similar in style to the examples shown in the figures below. Try to get them as close as possible but don't fret over pixel-perfect accuracy. The user won't notice an image being a few pixels to the left but they WILL notice the search not working or errors popping up on their screen!

Home Page:

This will be your landing page. Users can use the header at the top to navigate back to the Home Page (this one so it would just refresh) or to Login / Register. The details of Login / Register are provided in the Login/Register Page section. The SalEats logo should also redirect the user back to the home page.

Users will be shown this screen with a page navbar at the top and an aesthetic banner in the main content section with search functionality below it. There is a field for restaurant name and another for location. There are also four sort options that can be used (Best match, Review Count, Rating, and Distance). The details of the search are outlined in the "Search Page" section.



Home Page

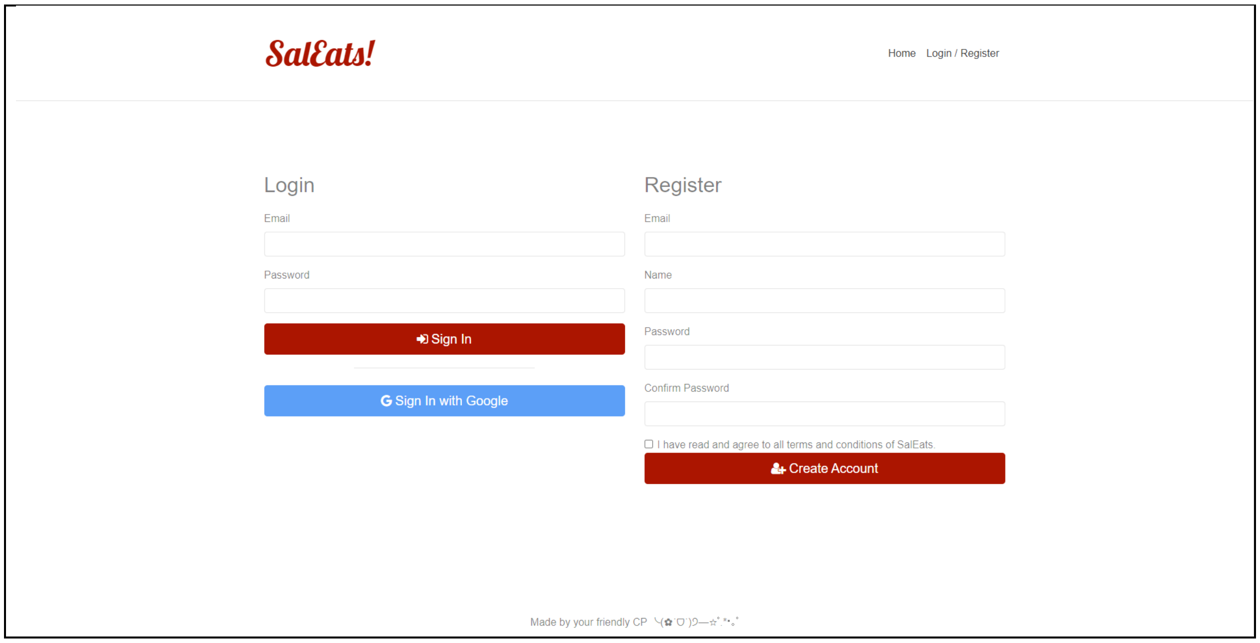
Login / Register Page:

Your application should allow the user to create an account and also allow users to sign in with their Google account. In other words, users will have two different login options, as shown below. Users can choose to login either with their credentials or with Google Sign-in. You can learn more about the Google Sign-in API here: https://developers.google.com/identity/sign-in/web/sign-in.

Here are some possible errors to check for when creating the Login / Sign Up page:

- Data is missing
- Data is malformed (i.e., Email does not contain an @ and ends in .com, .net, .edu, etc.)
    - No need to go crazy on the email validation beyond the above. Check this link to see how complicated perfect validation would be!
- There is already an account associated with the email
- When registering user password and confirm password do not match.
- This list is not comprehensive so if you find anything else that could break the functionality, it is your job to handle it

When signing up, the user will enter their information in the required fields (be sure to display an appropriate error message if there are missing or improperly formatted fields). Upon a successful sign up, the user should automatically be logged and redirected back to the Home Page. For this assignment, the username and password strings can be stored directly in the database without any security measures.



Login/Register Page

Search Page:

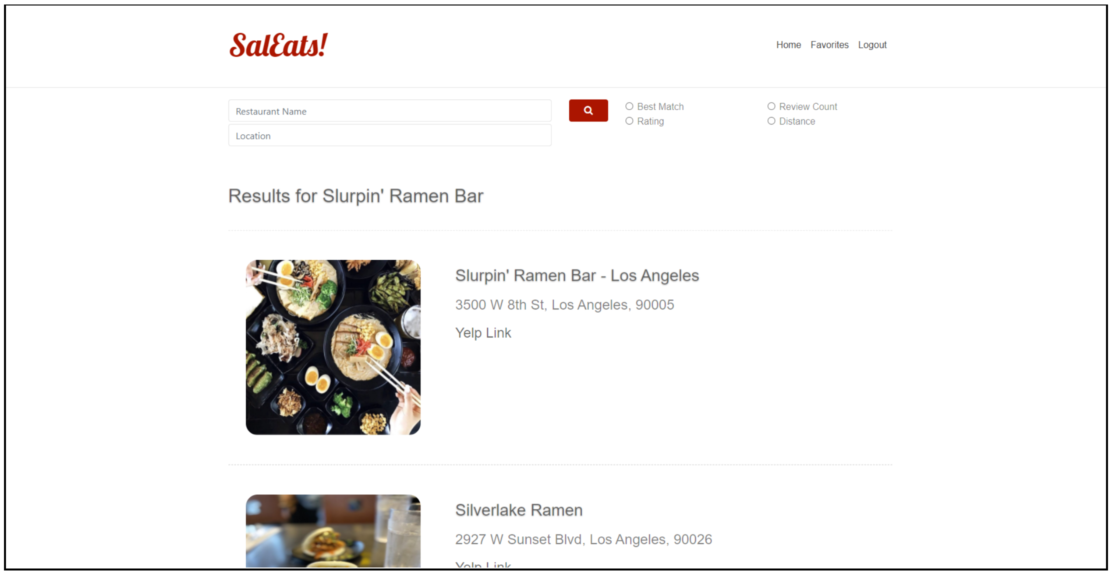
Users can run a search by entering a restaurant name, selecting the order of the search results (using the radio buttons), and then clicking on the magnifying glass icon (the search button). A successful search leads the user to the Search Results Page, as shown below.

The user first arrives at this page after a successful query from the Home Page. Using the information from the Yelp API, your program should display the 10 best results based on the search query and sort option. Each row of information contains an image related to the restaurant, the restaurant name, the restaurant address, and a link to their Yelp page.

Note: In the case of Yelp page links being really long, the text displayed for the link can be changed to "Yelp Link" instead of the complete url text. You can just modify the <a> tag used for the link. Keep it consistent though. If this is done for one link, it should be done for all links.

Users can use the top menu bar to navigate between the Home Page by clicking on "Home," and the Login / Register Page by clicking on "Login / Register." If the user is already logged in, the menu bar should instead contain "Home," "Favorites," and "Logout." This menu bar should persist between all pages of this application.

Clicking on the logo (SalEats!) should redirect the user to the Home Page as well. This should apply to all the pages in this assignment.



Results for Slurpin' Ramen Bar

Slurpin' Ramen Bar - Los Angeles
3500 W 8th St, Los Angeles, 90005
Yelp Link

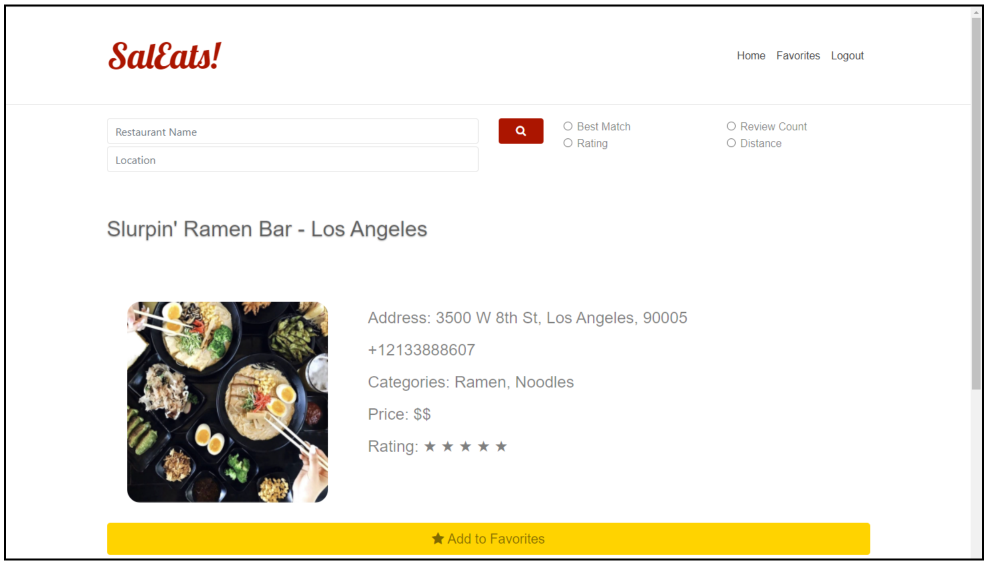
Silverlake Ramen
2927 W Sunset Blvd, Los Angeles, 90026

Search Page

Details Page:

This page contains additional details about the restaurant that was selected from the Search Results Page. This page should contain the restaurant name, address, same image from the Search Results Page, phone number, list of categories, price (in dollar signs), as well as star rating. Clicking on the image now redirects the user to the Yelp page of the restaurant.

Additionally, underneath the image should be one button: “Add to Favorites”. The former button will display either “Add to Favorites” or “Remove from Favorites” depending on whether the restaurant is already in the user’s favorites list or not. Once the button is clicked and the restaurant is added to the favorites list, the text and functionality should change without needing to refresh the entire webpage.



Slurpin' Ramen Bar - Los Angeles

Address: 3500 W 8th St, Los Angeles, 90005

+12133888607

Categories: Ramen, Noodles

Price: $$

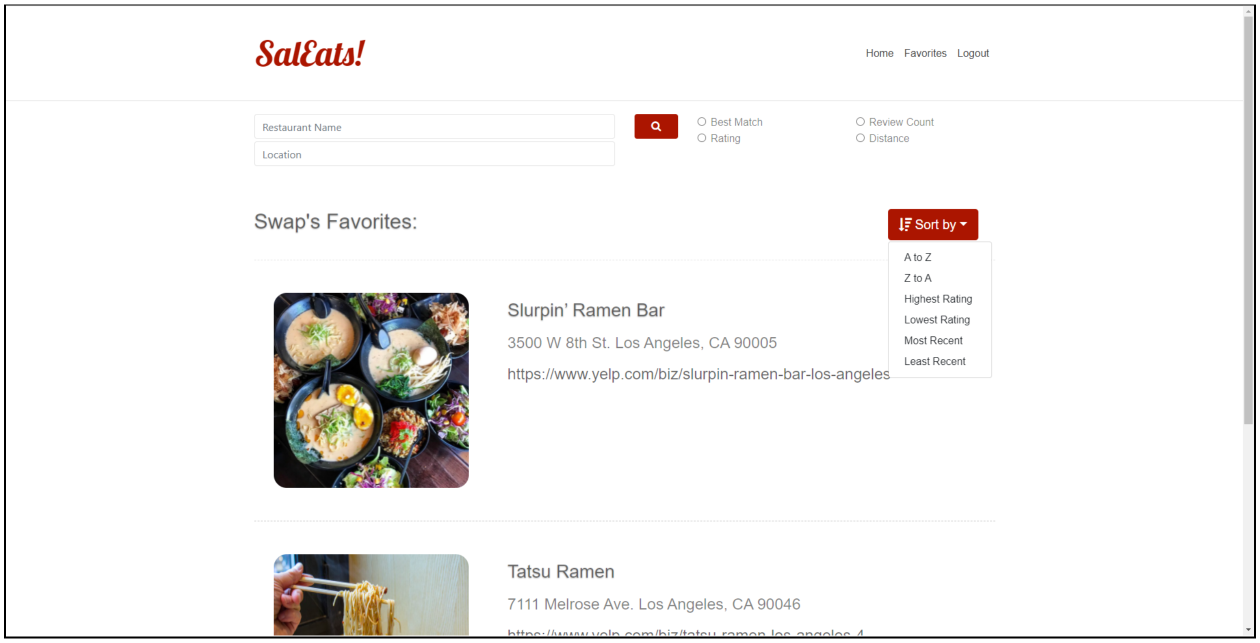
Rating: ★ ★ ★ ★ ★

Details Page

Favorites Page:

In this assignment, you will also need to track user information. At this point, you should have constructed a simple MySQL database with a table that stores usernames, passwords, and emails. You will now need a new table to store user favorites for each user as you will need to display this data on the Favorites page, shown below.

The favorites should be displayed chronologically (most recent first) by default.
The user should also be able to navigate to the Details Page of the restaurant by clicking on the image associated with that restaurant.



Swap's Favorites:

Slurpin' Ramen Bar
3500 W 8th St. Los Angeles, CA 90005
https://www.yelp.com/biz/slurpin-ramen-bar-los-angeles

Tatsu Ramen
7111 Melrose Ave. Los Angeles, CA 90046

Favorites Page

Extra credit: *The user can change the sorting order by using the "Sort by" dropdown menu on the page. Selecting a new sort order should reorder the favorites list without needing to refresh the entire page. Alphabetical is based on the name of the restaurant, rating is based on the star rating of the restaurant, and recency is based on the time the restaurant was added to the favorites list.*

Additional Implementation Notes:

- Please use Tomcat 10 for this assignment!
- The font used for the SalEats logo is called lobster and can be found here. There is a convenient css link on that page as well but you'll need to do some research on how to use custom fonts with css.
- The Yelp API should best be called from your back-end code and I believe you'll get errors if you don't. If you only call it from the front end via JS, that will be **at your own risk**.
    - You all have knowledge on how to parse JSON by this point. Given the details of the returned JSON from the Yelp API documentation parsing this via GSON should be fairly simple.
    - Check out this website for a good example:
        - https://switchswap.me/posts/2021-11-12-api-calls-with-java
- You can use JSP's to take your object set and dynamically server a web page
- Learn more about JSTL here: https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm
- Small code snippet example:

  // In servlet

  ```
  ArrayList<Restaurant> restaurants = // your restaurants;
  request.setAttribute("data", restaurants);
  ```

  // Pass data to jsp for it to fill out with

  ```
  RequestDispatcher dispatcher =
  getServletContext().getRequestDispatcher("/your-results-page.jsp");
  dispatcher.forward(request, response);
  ```

  // In JSP (Just one method of many)

  ```
  // Parse using JSTL tags:
  <c:forEach var="restaurant" items="${data}">
      // Your code here
  </c:forEach>
  ```