Katrina Cwiertniewicz, Brittany Klose, Deare Abudushalamu

CSC Database Management

12/9/24

Final Project

Introduction

This project utilized HTML, JavaScript(JS), and MySQL languages along with Express, Node.js, Mysql 2, and Cors framework to connect and run everything. MySQL functionalities included: Select, Insert into, Delete, and Update. My team used Visual Studio Code(VSC) and Live Share to build out and collaborate on the website. The basis of our project was an online ordering website for a cafe called Java Cafe. Regarding the user story, customers can log in or create an account, browse the menu, search and select a cafe location, start the orders page for cart lists and item lists, users can add or remove items from their cart. Users can also see the cart list, and view and register for special events/promotions at the cafe. Below we will further break down the project by section and then go into performance evaluation by member.

The MySQL Database

We used Google Could to build our database for security purposes. The foundation for the database included the following entities and attributes:

- Customer(customer_id, username, email, password, phone, DOB)
- Location(location_id, location_address, town_city, state, zip code)
- Events(event_id, name, date_time)
- Event_Registration(fname, name, location, event_name)
- Occurring(location_id, event_id)
- Items(item_id, name, price, calories, category, description, url)

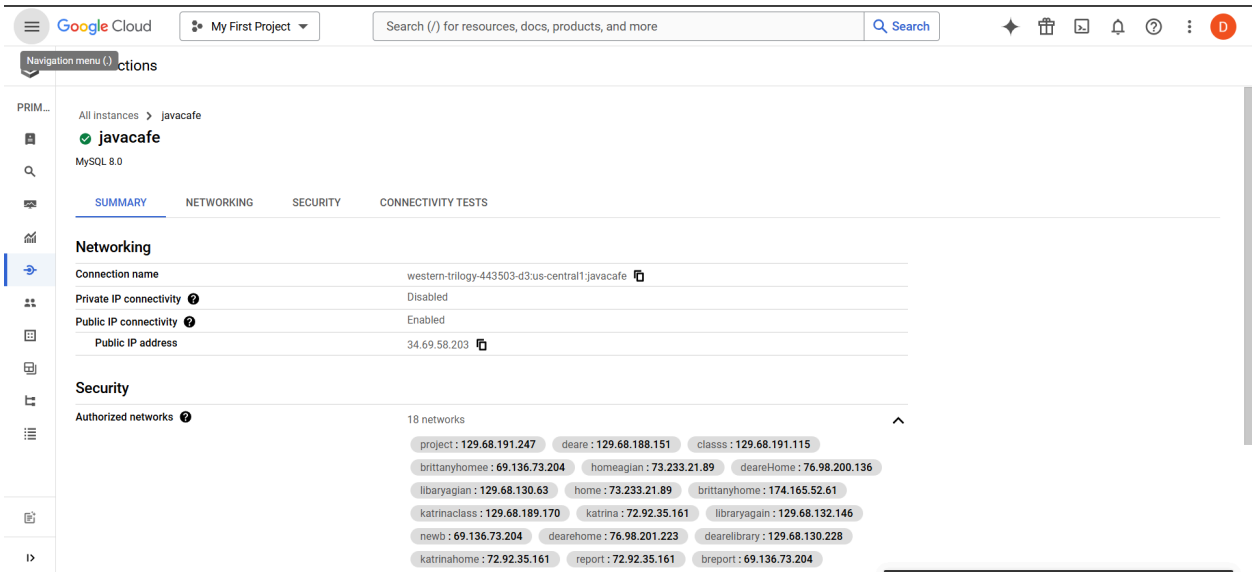
Our database also included the following entities below as well for extra functionality though we did not have time to utilize them on the back end. Our idea with rewards was to establish a count starting at 20 and each time a user made an order (clicked the order button) the count would

decrease by 1. The current number represented the number of visits/orders the user had left to earn a reward and could be seen when the user logged into their account. Additionally, we ended up creating a cart on the front end that could add and remove orders but were not able to connect it to the backend in time for our presentation. If we had been able to do so, it would have followed the setup below where a user had a cart connected to them by their user ID and a cart ID. It would store everything they ordered and the price. Then each order in the order table would be connected to a user, a cart, and a location for pick up or delivery.

- Rewards(customer_id, visits_til_reward)
- Order(order_id, customer_id, cart_id, location_id)
- Cart(cart_id, customer_id, item_id, item_name, item_price, quantity, total_price)

(ER Diagram here)- The photo won't load for me think we needed a new one anyway

Deare used her coupon to establish a database on Google Cloud. Then each member created a new MySQL connection with the details of Deare's original connection. I.e. name javacafe, root 34.69.58.3306, our unique IPV4 address, and our unique MySQL passwords. Once we each had the database we could then individually test queries if needed as long as Deare was running the database.



Ordering Process

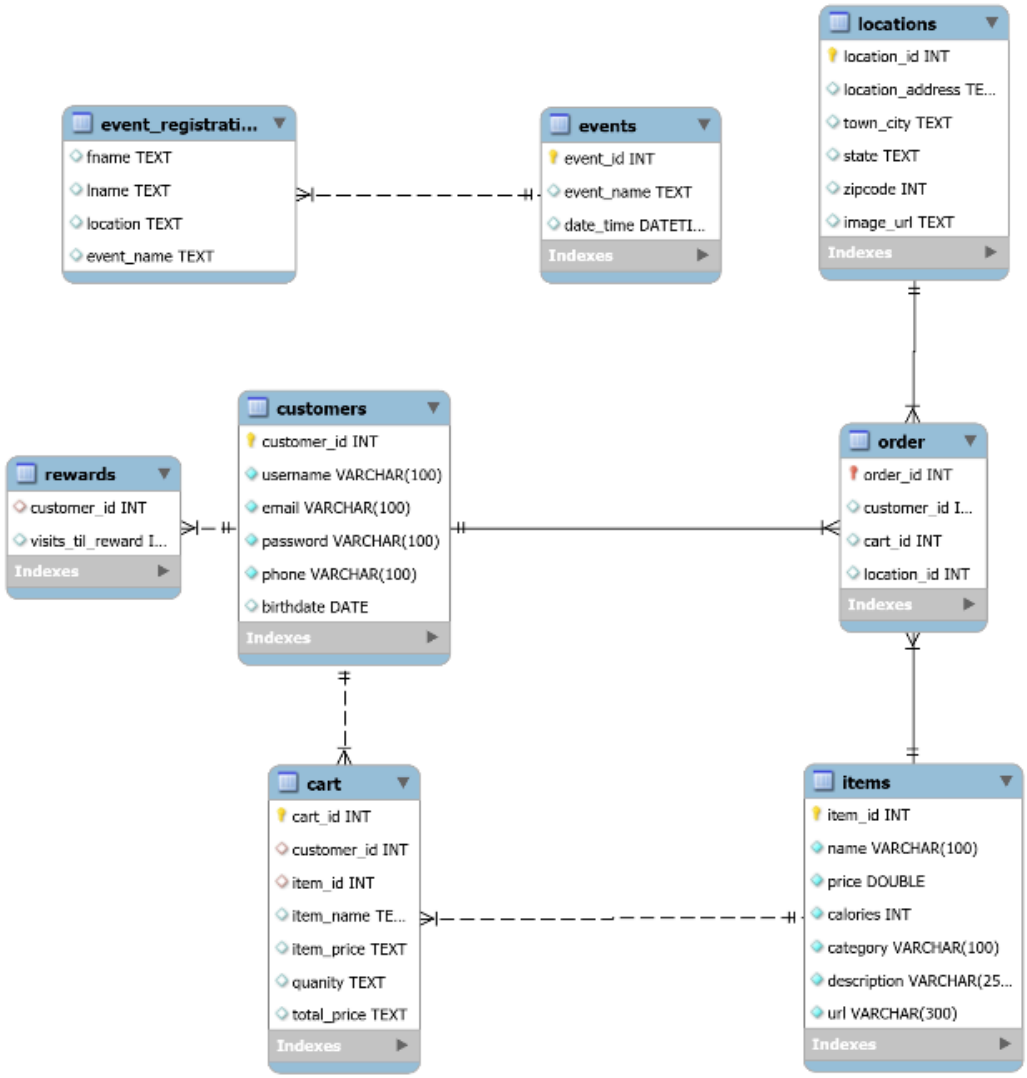
The ordering process begins with the user selecting a location. To do so the user selects the Location tab in the navigation bar and is brought to the location page. There the user will find cards for all locations which will include a photo of the cafe, and the address. A user can also use the search bar to look up a specific city. If their searched location is included in the database it will appear when the user hits enter. Next the user can click Start Order in the navigation or Cart in the top right corner. Either button will bring them to the ordering page where they can see the whole menu and click add to cart. Any items the user adds to their cart will be listed on the right hand side along with the price of the item, the quantity and then the total price for all items in the cart. As users add/remove items and or quantities the total price will appropriately increase or decrease. If a user completely removes an item it will be removed from the cart list. Once a user is satisfied with their order they can press the Order button and a pop up window will open letting them know their order has been submitted.

Connecting Database & Functionalities

The main functionality of our project is the interactions with frontend and backend. Our backend server running in port number 8081, and our front-end server running in port number 5051. The first interaction is I defined a route in our backend server in the database.js file named /search to fetch the locations based on city name, the route handles the GET HTML method to the '/search'

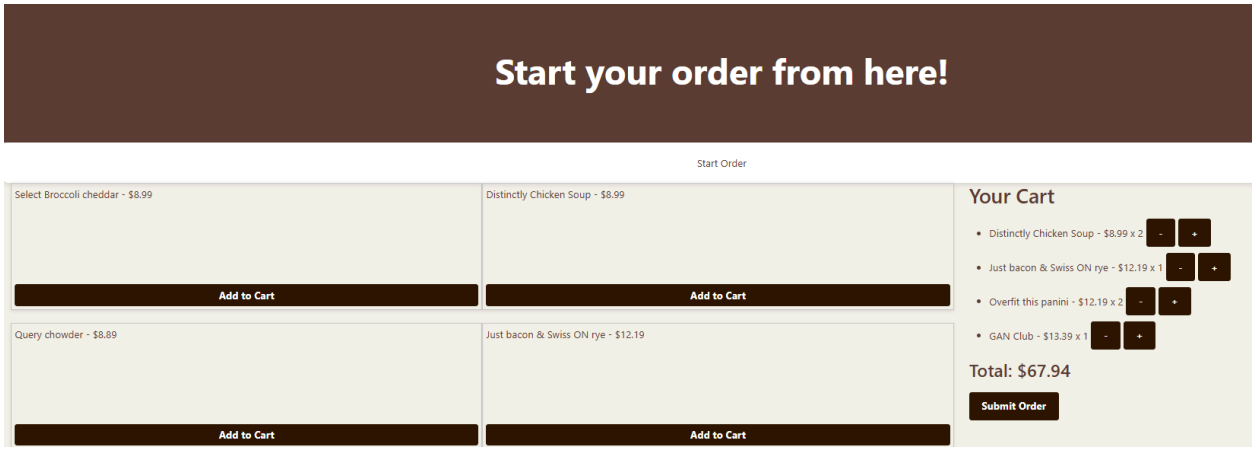
EER Model

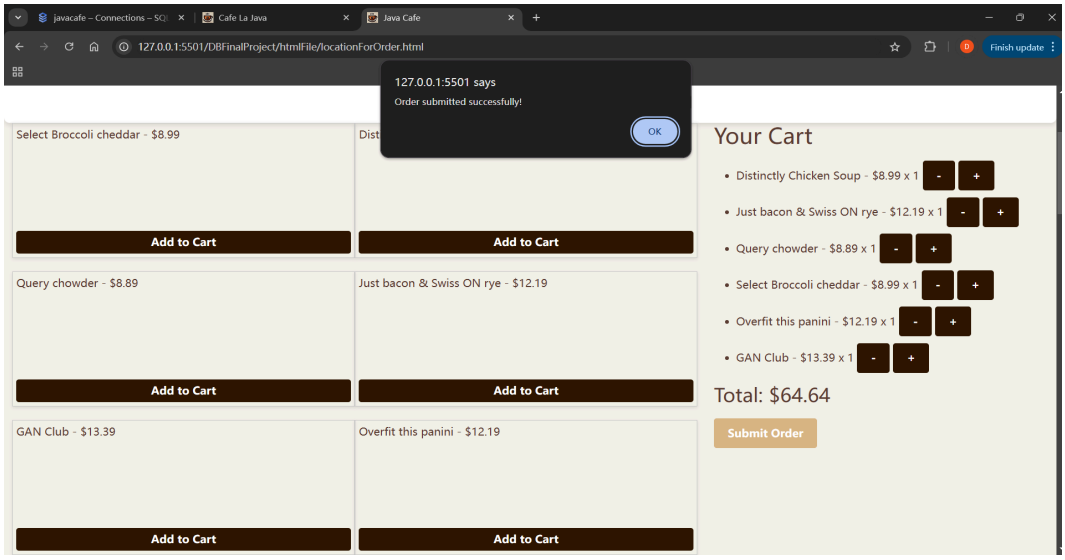
For our EER model, we wanted to include objects we considered using with our website. Rewards and orders were not used for the functionality but were a feature we considered adding to enhance our website. Event_registration was not connected in the EER model. Originally, we planned to connect event_registration to customers but used a username instead of first and last name for account creation. If we were to set it up differently we would incorporate our rewards order system and add first and last name to customer information.



Start an Order

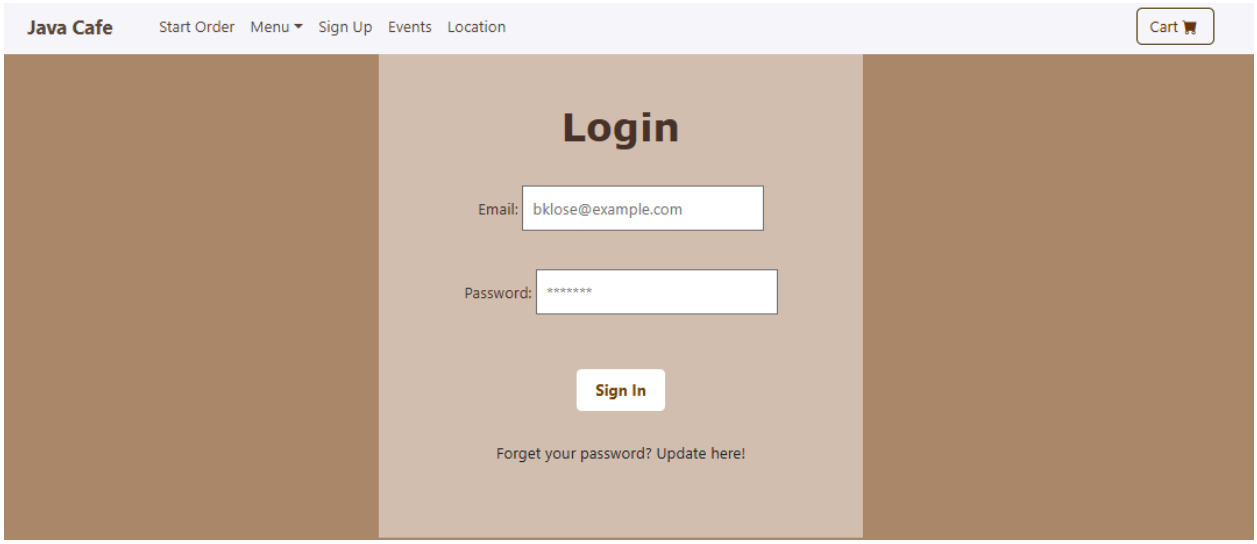
To start an order a user could select start an order on the navigation menu or select cart on the right hand side to get to the order page. On the order page users can add and remove items from their cart. Each item will be listed on the right hand side under the cart along with the price quantity and total. When the user is ready they hit submit order.

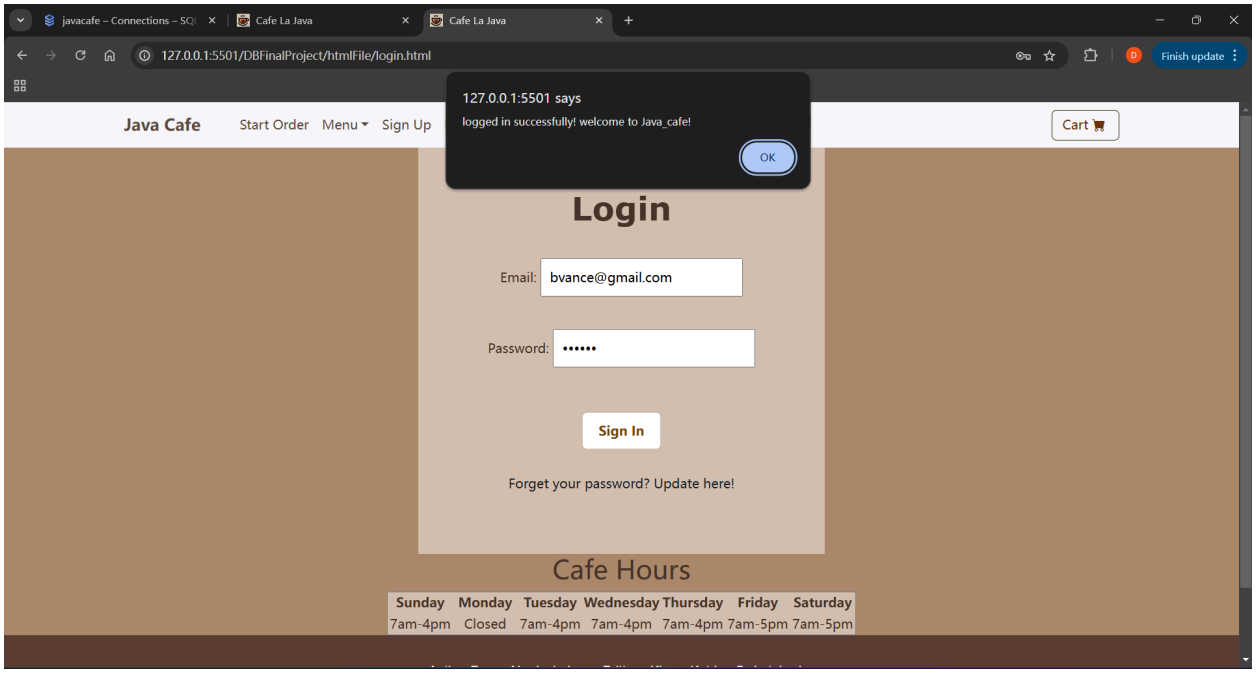




Login

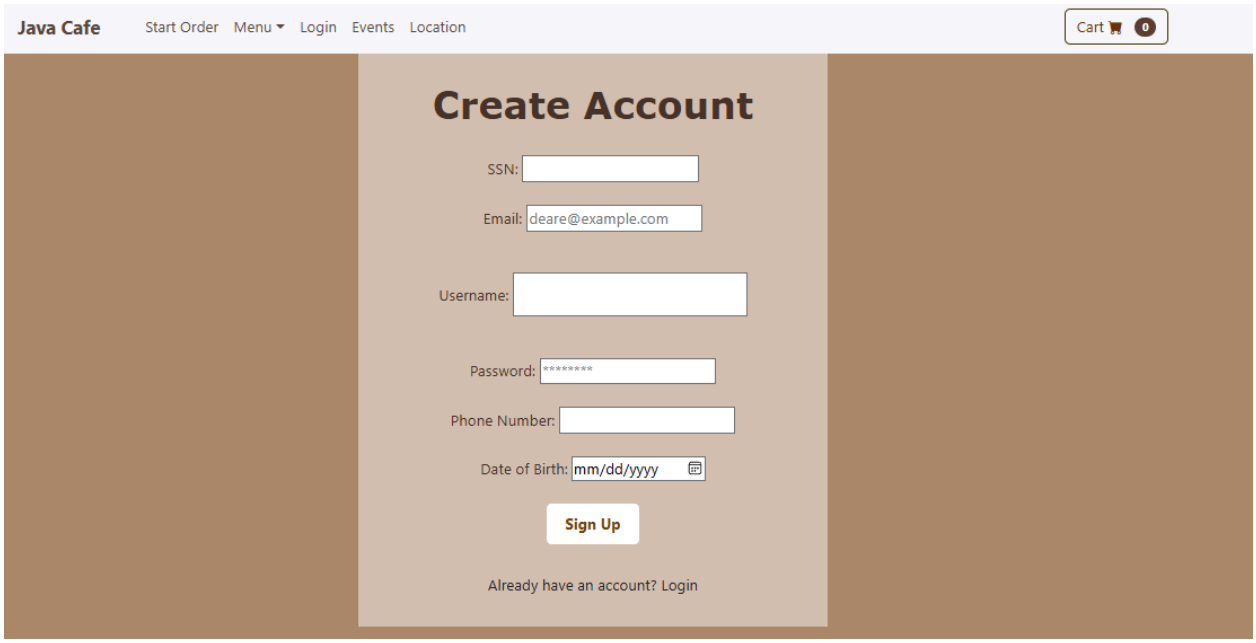
To login a user simply navigates to the login page via the navigation menu. When they are in the login page it will disappear from the navigation. The user can input in their email and password and hit sign in. The database will then search the customer table for a user with that matching password and email and return success if the match is found.

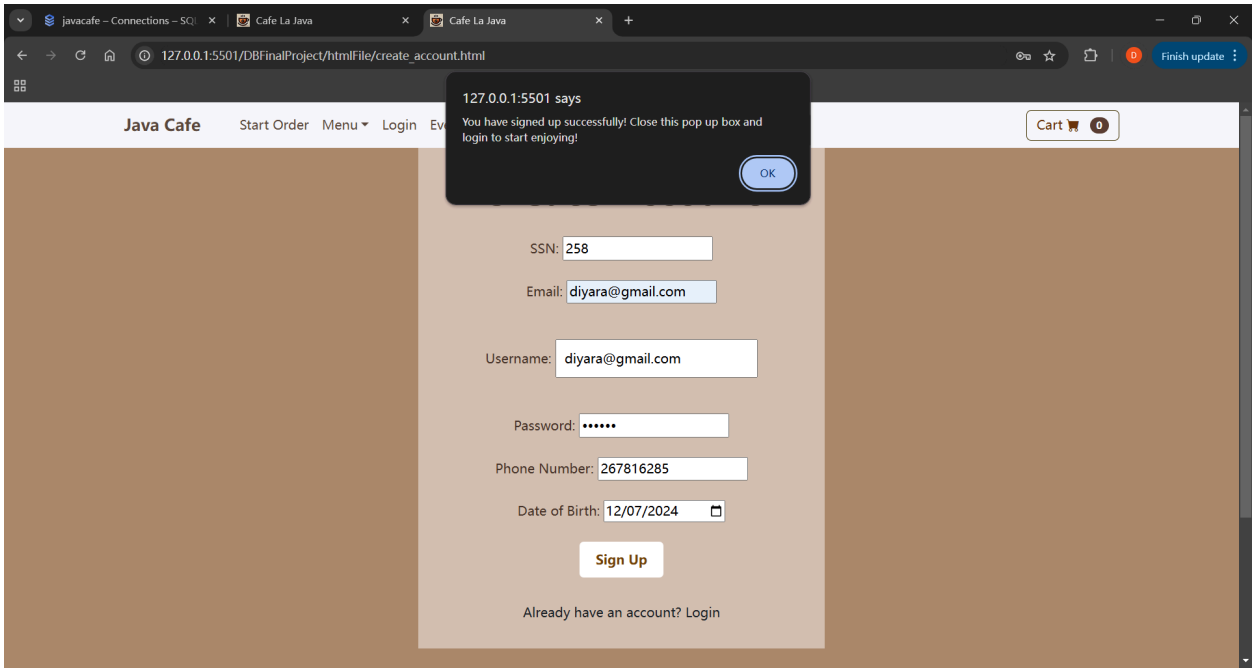




Create Account

To create a new account a user will navigate to sign up via the navigation tab. From there they will input their information into all the required fields and hit sign up. This new account will be added to the database in the customers table. They can then head to login and sign in to their new account.

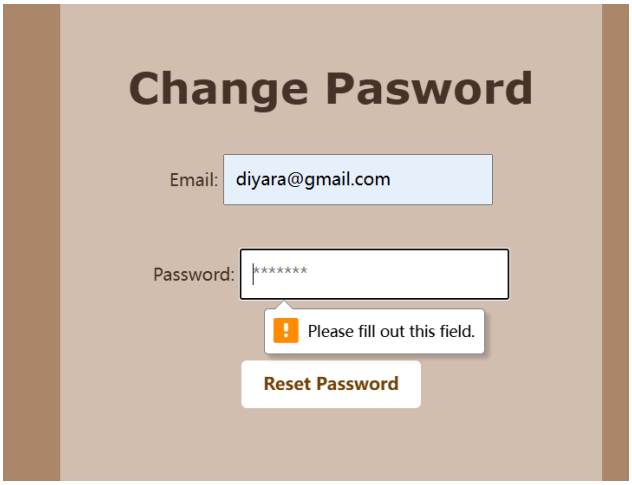


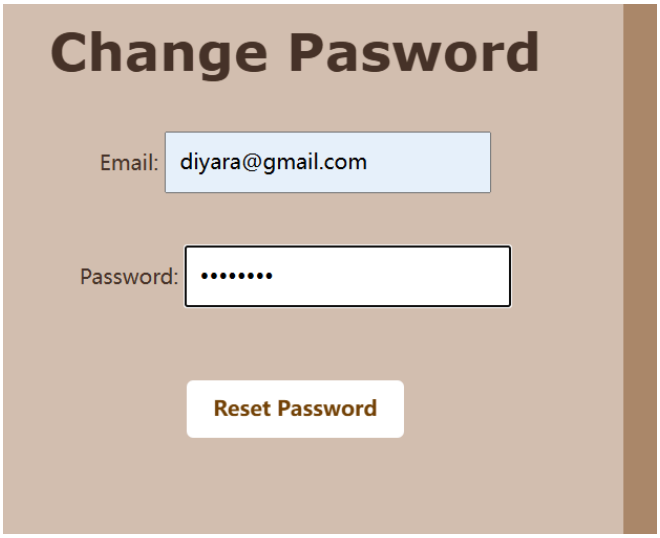


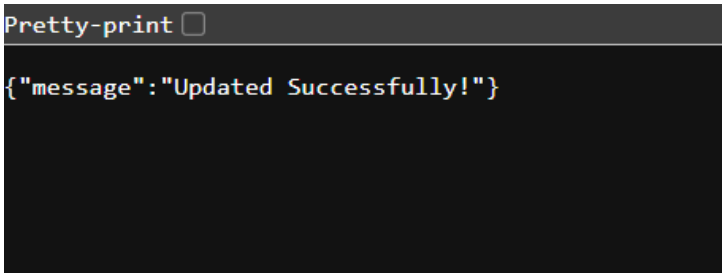
14	zeldayeahh	zelda@yahoo.com	omg123124se	3333333333	2024-12-14
15	lisaballsssss	lisa@gmail.com	2342342rufv	8888883333	2024-12-17
17	nightlordheehee	Rhysand@hotmail.c...	2309r2utfofs	3338848383	2024-12-18
18	jiljillll	jill@Yahoo.com	232wfefef	1112228374	2024-12-24
22	jwalshto	walsh@hotmail.com	3w54bhdtdg	5553337878	2024-12-25
235	graceiscool	grace@hotmail.com	43854hods	7774448585	2024-12-05
258	diyara@gmail.com	diyara@gmail.com	123456	267816285	2024-12-07
1341351351	kcwiert	kc10804372@sju.edu	eageragqergwr	24524523523	2024-12-10

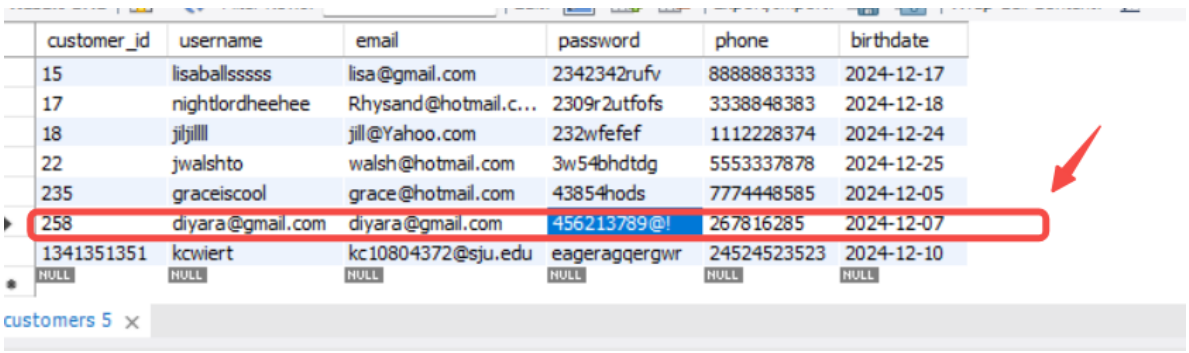
Change Password

If a user wants or needs to reset their password they can click ‘forgot password’ on the login page and be brought to the change password form. From there a user will input their email and the new password and hit reset. Then when a user goes to login again they can use their new password.









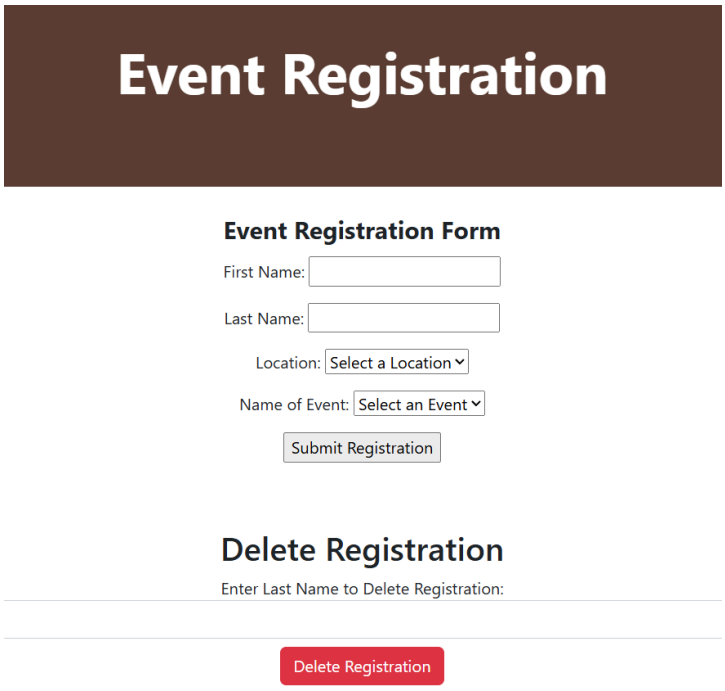
customer_id	username	email	password	phone	birthdate
15	lisaballsssss	lisa@gmail.com	2342342rufv	8888883333	2024-12-17
17	nightlordheehee	Rhysand@hotmail.c...	2309r2utfofs	3338848383	2024-12-18
18	jiljillll	jill@Yahoo.com	232wfefef	1112228374	2024-12-24
22	jwalshto	walsh@hotmail.com	3w54bhdtdg	5553337878	2024-12-25
235	graceiscool	grace@hotmail.com	43854hods	7774448585	2024-12-05
258	diyara@gmail.com	diyara@gmail.com	456213789@!	267816285	2024-12-07
1341351351	kcwiert	kc10804372@sju.edu	eageragqergwr	24524523523	2024-12-10
NULL	NULL	NULL	NULL	NULL	NULL

customers 5

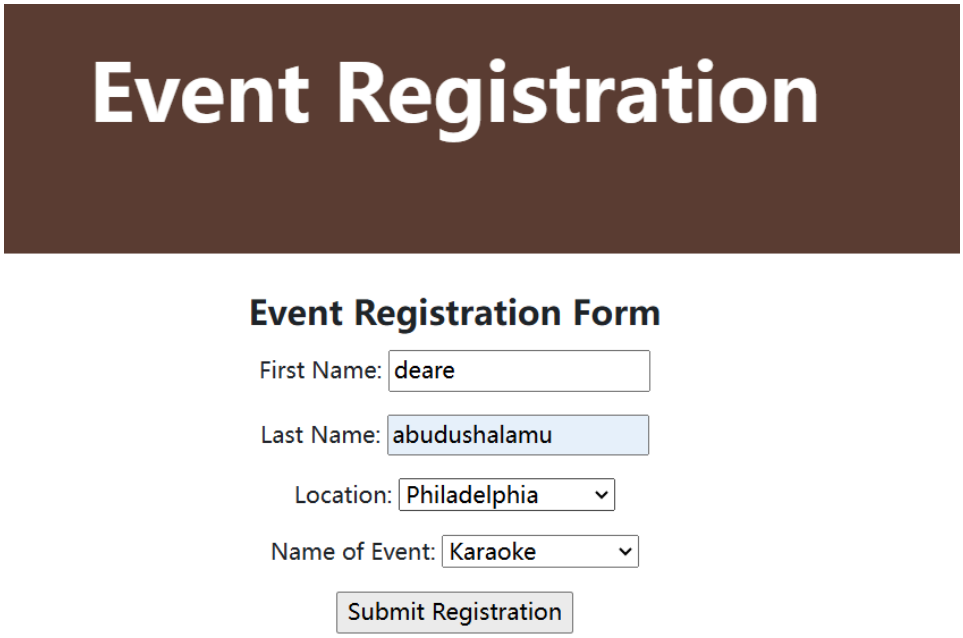
Event Registration

For event registration, the user clicks on events from the navigation bar. This brings them to a calendar with all available events. All events have a hyperlink attached to them. Once an event is clicked this will bring the user to the event registration page.

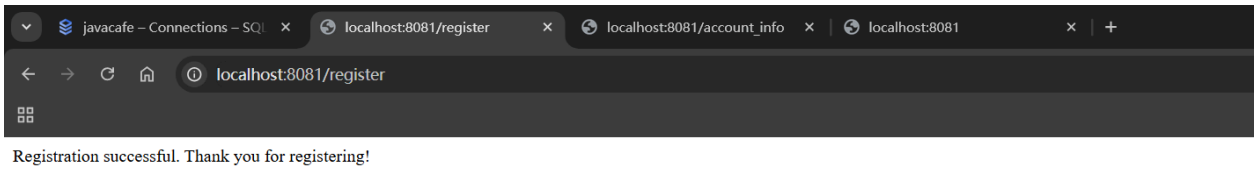
The user will input their First and Last name and choose from the drop-down options for Location and Name of Event.



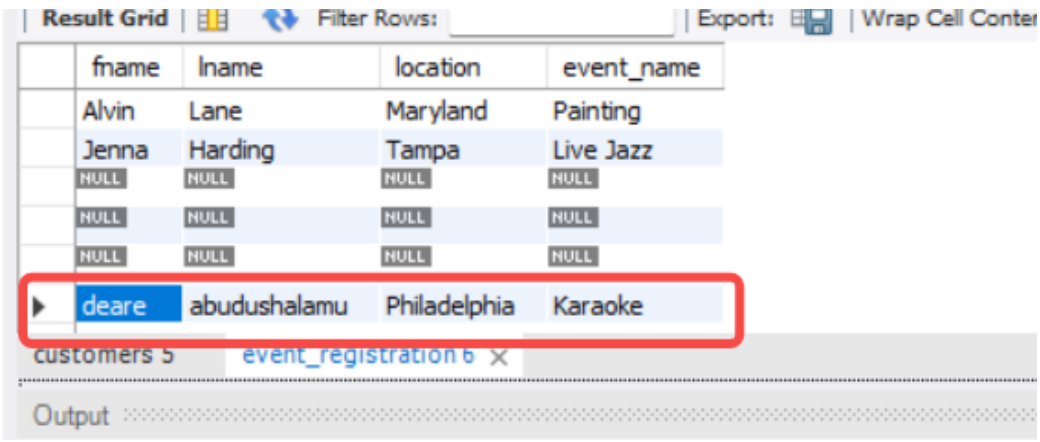
For error checking, a user is required to add a field for each section before being able to submit the registration.



Once submitted successfully, the user will receive a confirmation message that says, “Registration successful. Thank you for registering!”.

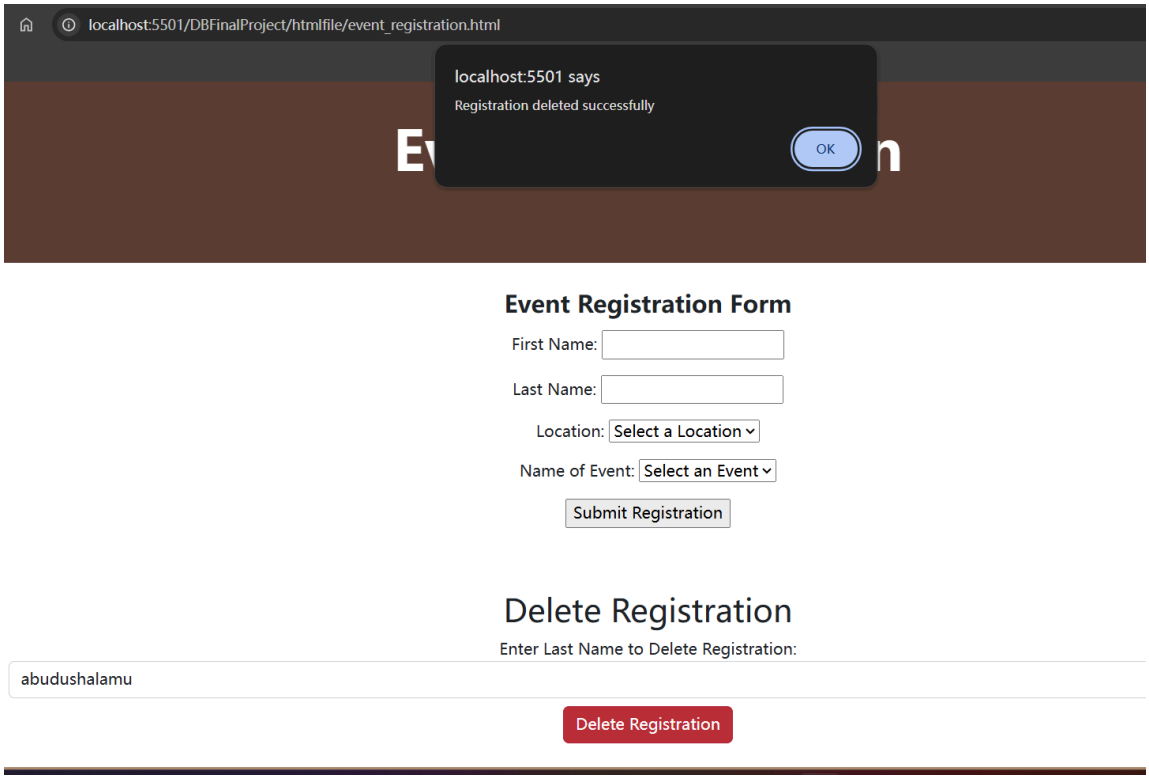


Once the registration is submitted successfully, the information is added to the table event_registration.

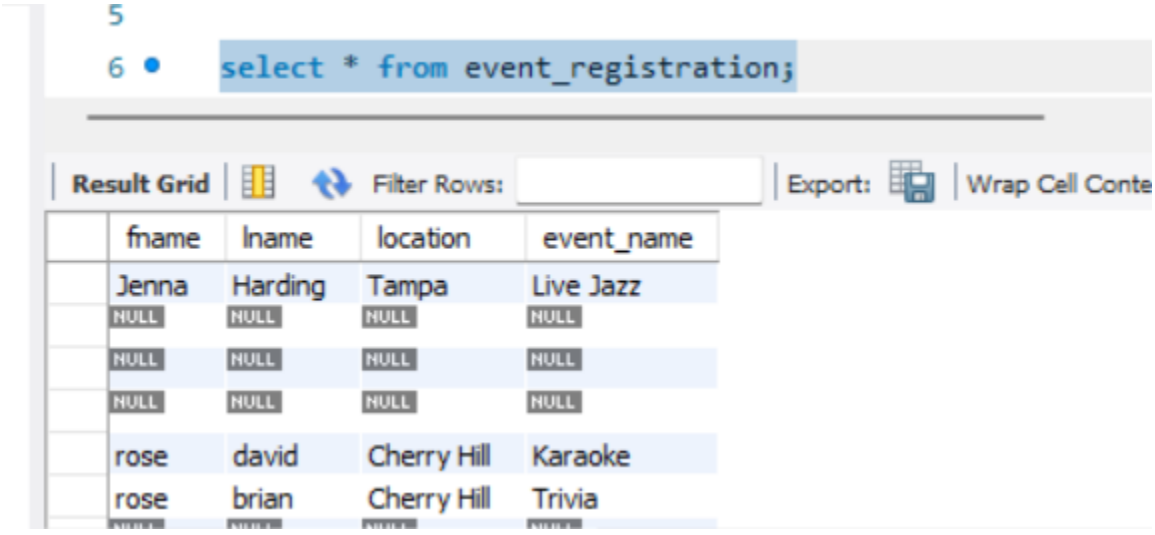


Delete Registration

To delete a registration, the user reloads the page and adds their last to the text bar. If the user's last name is a part of the event_registration table, the user will be deleted and receive a message that the "Registration deleted successfully."

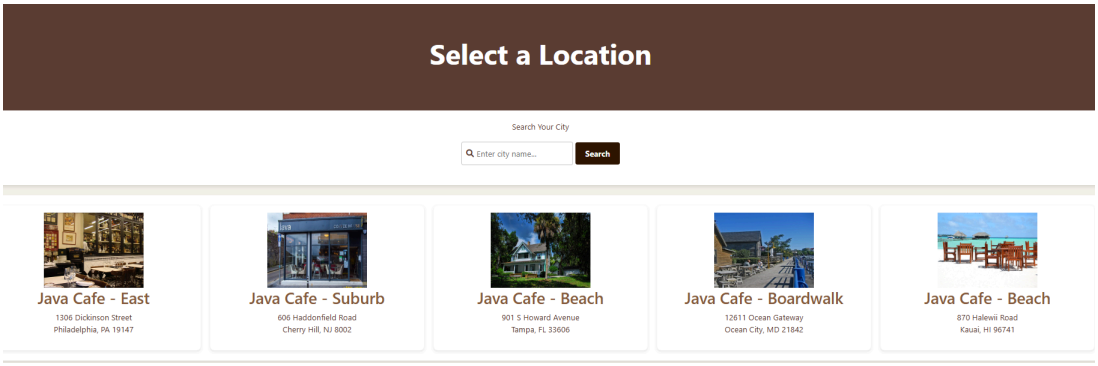


Once the registration is deleted successfully, the information is deleted from the table event_registration.

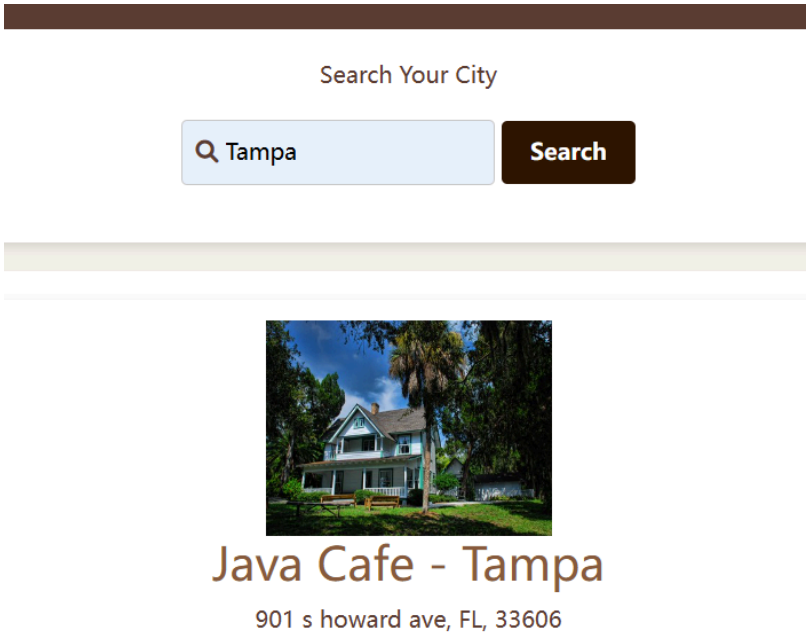


Locations page

The locations page was for the user to select the cafe closest to them for delivery or pick up. To select a location, the user types in the name of a city. If a city matches one of the city locations of our cafes, the cafes are filtered to only display that one location. If the location is not one of the listed cities, all locations stay on screen.



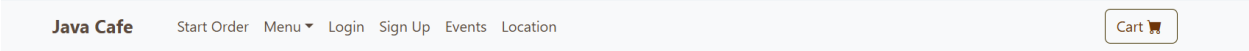
For example, if the city “Tampa” is searched for, only the Java Cafe in Tampa will appear.



Additional Features

Navigation Bar

Our navigation bar includes links that connect to Main Page, Start Order, Menu, Login, Sign Up, Events, Location, and Cart(which are connected to Start Order)

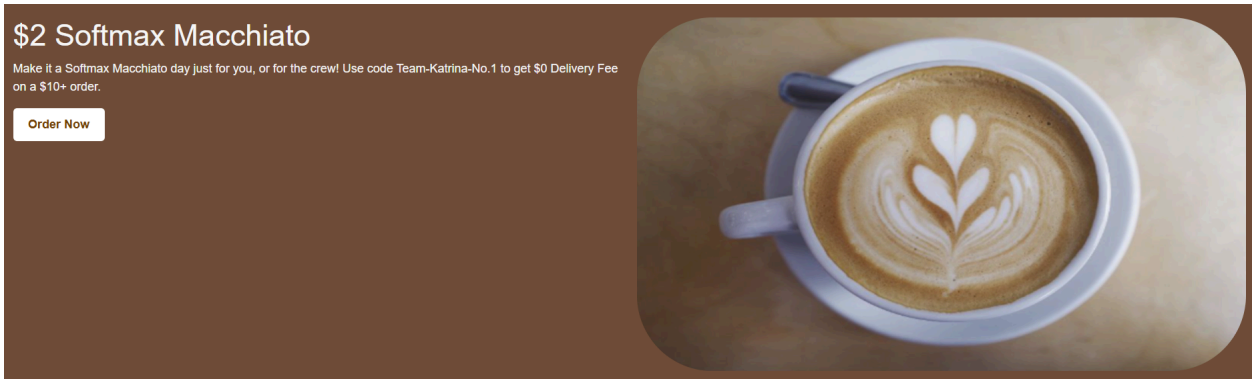


Main Page

The main page includes a video of our cafe and music that can be turned on and off with a button action while browsing the main page. If either video or music does not work on the browser a message will appear to show that the browser does not support this functionality.

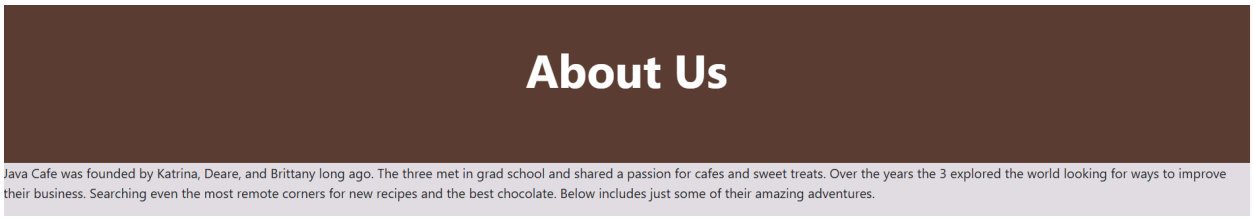
Promotional Deal

Our promotional deal includes an Order Now button. If clicked, the user is sent to the Start and Order page.



About Us

At the bottom of the homepage includes a section detailing the stories of the owners with fun images of adventures they took to help run the cafe.

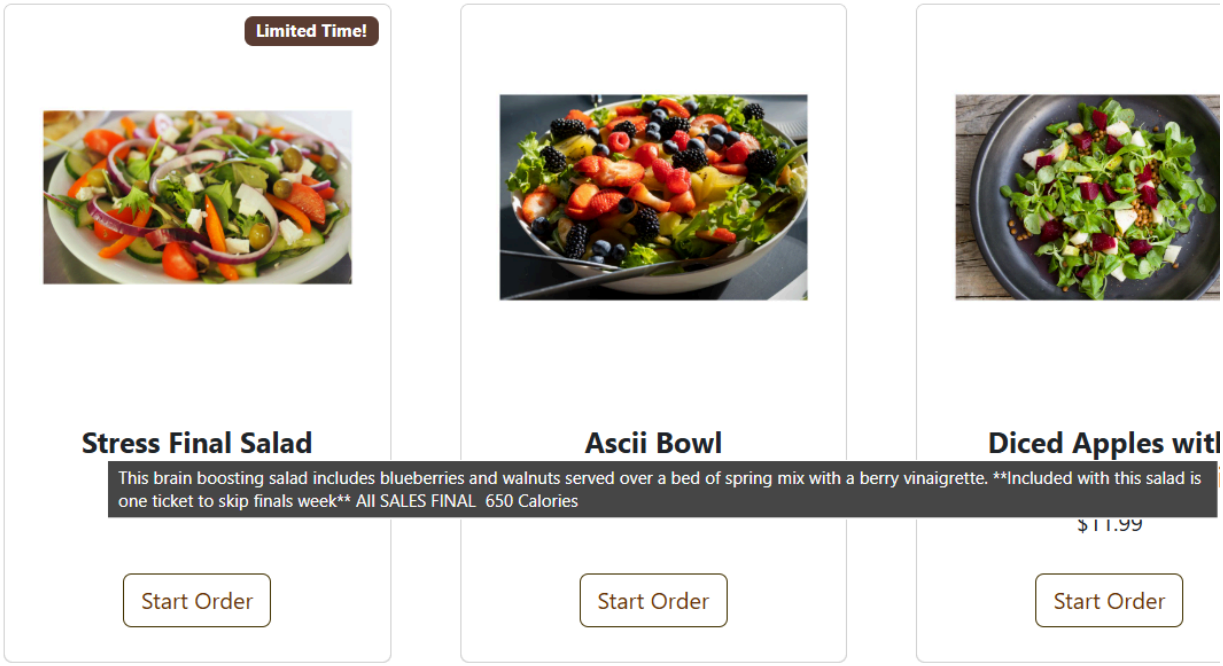


Cafe Hours

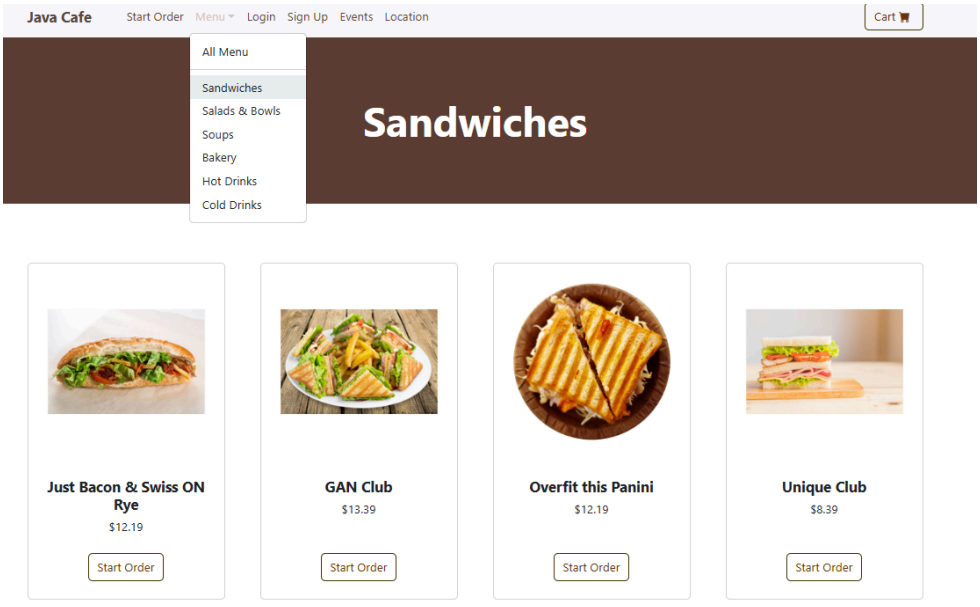
Our cafe hours are displayed at the bottom of every page

Menu Pages

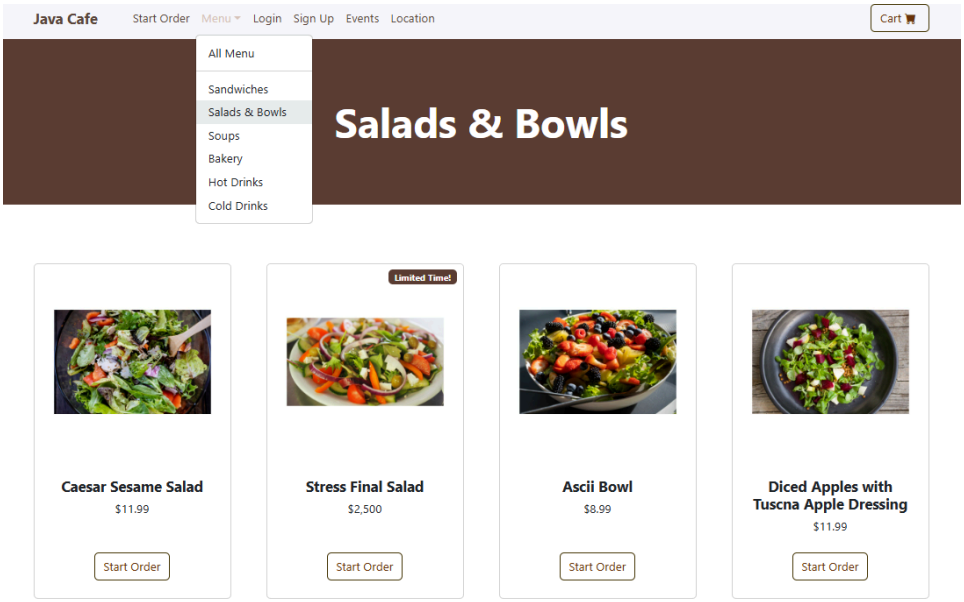
Each menu page includes all products available from our item table. Each item includes the name, image, and price. The description and calories can be found by hovering over the name of the item. A user can view the whole menu or select a specific category. Below includes images of each of our menu category pages.



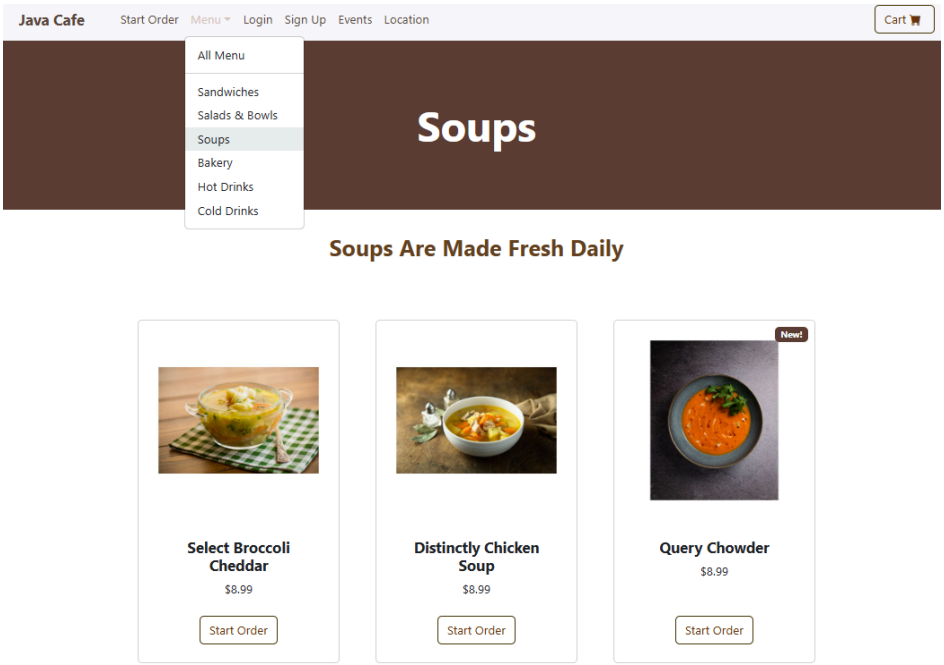
Menu- Sandwiches



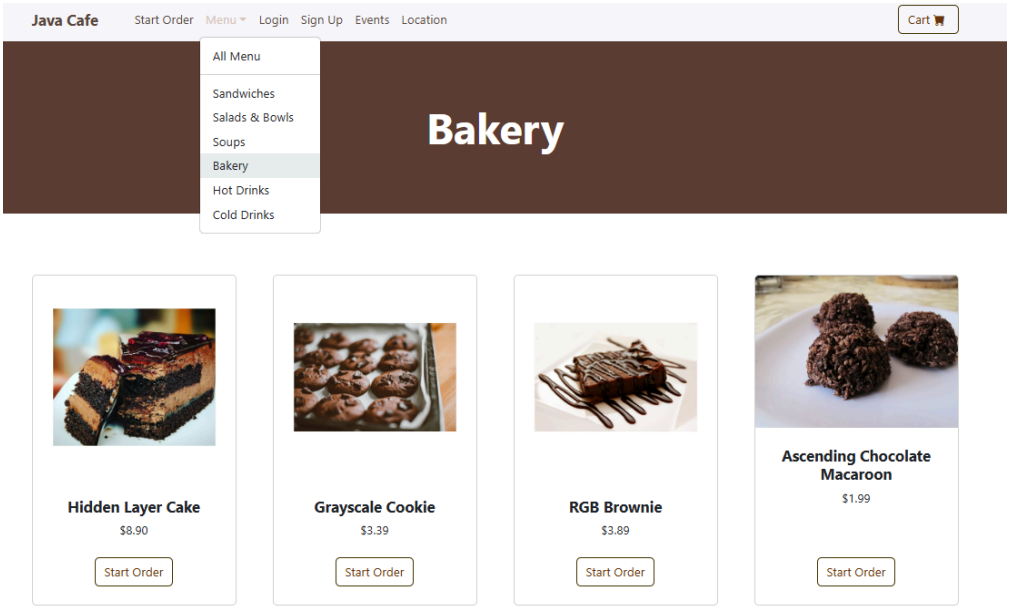
Menu- Salads & Bowls



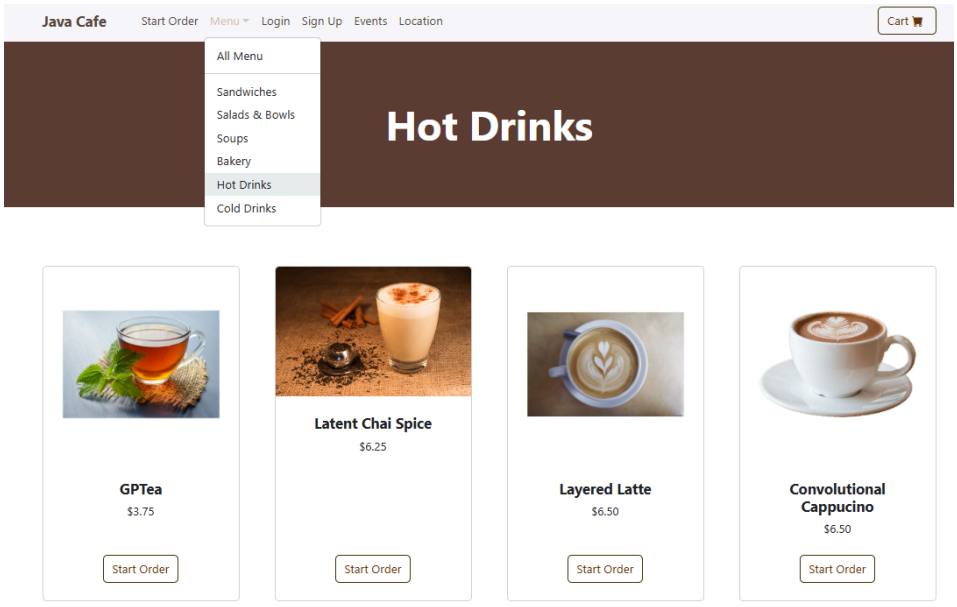
Menu- Soups



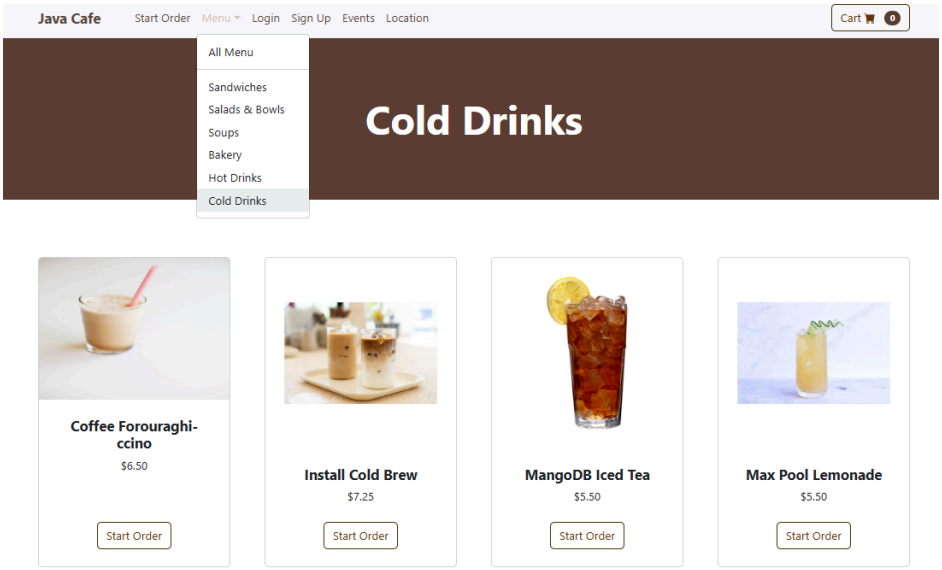
Menu- Bakery



Menu- Hot Drinks



Menu- Cold Drinks



Ranking Statements

Member 1: Katrina Cwiertniewicz

My teammates and I agree that I handled 33% of the overall project. My specific tasks included:

Task 1: I designed and created the events calendar page and hyperlinked events to event registration.

Task 2: I wrote the SQL Schema, the SQL Script, the ER model, and the Relationship Model.

Task 3: Implemented images, price, name, and hypertext descriptions for all menu items.

Task 4: Incorporated music and video with the main page using action buttons and creating a music function feature.

Task 5: I helped with the main page design, CSS color palette, and added images to the location page.

Task 6: I helped troubleshoot login and events registration and attempted to use bycrypt for password hashing.

Task 7: Co-wrote the Final Project Summary and helped troubleshoot in group meetings.

Member 2: Brittany Klose

My teammates and I agree that I handled 33% of the overall project. My specific tasks included:

Task 1: I designed and implemented the sign up page and functionality. This included setting up the HTML page(create_account.html), CSS page(create_account.css), and a javascript page(signup.js). The later two were linked in the HTML page to connect the files. I then wrote out the back end post call on database.js where we had connected our MySQL database. The layout of sign up as seen in the photos above was a form made on HTML with boxes for the user to input a username, password, phone number, and DOB. These were the attributes tied to the customer entity table in our database. When the user inputted their data and clicked submit it triggered an event listener that grabbed all the inputs and then made a POST call to add the new user to the database. If it was a success the website would pop up a box that told the user they had successfully created an account. I originally had some issues getting signup to work. I realized the issue came from the lack of a customer ID. I tried importing a random ID generator but we had some trouble with it, so in the end I simply made another input box for customer ID. It was labeled as SSN mostly for fun as in a real site for a simple cafe we would not ask customers for such sensitive information and if so it would be encrypted. This worked and we were able to sign up multiple new customers with the SQL script: INSERT INTO customers (customer_id, username, email, password, birthdate, phone) VALUES (?, ?, ?, ?, ?, ?)'. With the question marks representing the values the user would input.

Task 2: I designed and implemented the login page and functionality. Similar to sign up I created the HTML page(login.html) and javascript page(login.js). I then linked the javascript page and a CSS page(account.css) made by my team in the HTML, and then wrote the POST call on the backend on database.js. The login page was also a form in HTML though it only included email and password. When a user hit submit it would enable the POST call and the database would search in the customers table for a match and return a success message in a pop up box. The sign in form like with sign up was given an event listener grabbed through its ID using document.getElementbyId. The SQL script for the post call was `SELECT email, password

FROM customers WHERE email = ? AND password = ?. Enabling the database to search for a match.

Task 3: I helped create the initial relationship model set up for entities and attributes and seeded all the import files for the MySQL database. As discussed above we came up with a starting layout for the database for the entities: customers, orders, cart, items, events, event registration, reward, and locations. The customer table would hold all the information about a user we would need for sign in and sign up. The items table would hold all of the information about each menu item and so on. I then created a master spreadsheet that held all the individual sheets for each table that I downloaded as csv files and imported into VScode and MySQL Workbench. We particularly had fun naming all the menu items after coding terminology. It was definitely one of my favorite parts of the project.

Task 4: I collected and uploaded all the photos for menu items and attached all the URLs to the HTML page. I have some past experience with canva, a platform used for presentation and marketing to make ads and social posts. I created a folder for item images and loaded free stock photos from canva. I appended the folder in our VScode workspace and then added the pathway links to the front end HTML files to display on the menu pages.

Task 5: I wrote and created the about us section. This was primarily just for fun. We initially weren't sure what we wanted on our home page besides a welcome to the cafe banner, and didn't want the home page to just be the menu. So I offered to create an about us section that would include photos of us and our backstory with creating and running the cafe. Deare and Katrina later did an awesome job with the home page adding in videos and music and special items, but we decided to keep the about us section alongside for fun and were excited to show the class my makeshift photoshopping and storytelling. I used canva again to upload the selfie of me, Katrina, and Deare simply removing the background and adding in new backgrounds or cropping our faces onto other images. The stories I made up during a car ride over Thanksgiving break. Some of which were based on popular stories like Harry Potter, Game of thrones, Avatar the last airbender. Overall, just a fun little side thing for the website.

Member 3: Deare Abudushalamu

My teammates and I agree that I handled 33% of the overall project. My specific tasks included:

Task 1: I designed and implemented the location page with a search bar that can search location, start order page, navigation bar of the home page and its functionality, built Google Could database, database connection, event registration, event registration deletion, and update password functionality. First thing before developing the website I came up with the idea of using Vscode's live share extension which can live share all the front-end and back-end servers with team members to code together. I built the database on the Google Could Called cafe, whenever we try to connect with the database we will update our IP address for security purposes. I created a back-end server called database.js which includes an express framework for handling HTTP requests in node.js, MySQL2 client for Node.js is for connecting with the database to provide database interaction. Cors for web security used to maintain or relax cross-domain access restrictions, body-parser that parses request bodies in JSON formats. After importing the required modules, I created an instance of express and Enabled CORS for the frontend, also I created this backend server in port number 8081, and then I created a connection to the MySQL database using the hostname, user name, password, database name that I already created in Google Could. After that, I defined a route for the root URL, defined get HTML methods of a route '/Search' using SQL 'SELECT location_address, town_city, state, zipcode, image_url FROM locations WHERE town_city LIKE ?' to fetch locations based on the city name and I used this API to fetch the city name from location table design it return the in HTML format if the typed in information matches. If the city name does not match it will return an empty internal HTML. I define an endpoint to handle POST requests to '/register' to handle the form submissions for event registration using the SQL `INSERT INTO event_registration (fname, lname, location, event_name) VALUES (?, ?, ?, ?)` also there is an error checking for event registration if there no error send a success message to the client. I define an endpoint to handle POST requests to '/deleteRegistrationByLname' to handle the form deletion from the envent_registration table using the SQL 'DELETE FROM event_registration WHERE lname = ?' design to delete the event registration based on the user's last name who has registered event before. Define a route with endpoint '/account_info' that handles POST requests to update account passwords using the SQL `UPDATE customers SET password = ? where email = ?`

design to update user passwords based on their email address which is in our customers table. Also after completing the design, I debugged both the front end and back fixed all the bugs, and helped with every problem that team members encountered.

Task 2: I wrote the home pages the softmax coffee commercial content and the home page welcome content, and I commented on every part of the backend server to ensure everything was cleared to the members.

Task 3: In order to identify a solution to the error, I conducted research on the medium.com and Stack overflow websites. I also consulted a series of YouTube videos on the integration of the front-end with the back-end database. Having gained a deeper understanding of this process, I created a blog post on the medium.com to assist other developers who may encounter similar challenges. I created a navigation bar that included the Java Cafe logo that allows users to jump back to the home page and Start ordering the page with a cart list and item list, Menu navigation has its sub-navigation with all the categories of our menu and Location which allow user to search the location of our cafe, Event for user registration event and delete event from, log in and log up for user to sign in to our page.

Task 4: I integrated the front end with the back end using the library and frameworks that I mentioned in the earlier part of task 1.