GROUP 86 Milestone 3 PDF

Project Description:

Our database models a food delivery app, and its purpose is to manage data regarding the users of the app. There are three different classes of users for our app: the customers, the restaurants, and the couriers. During Milestone 3, we created an application that connects users to our database in a way that allows them to access and/or modify certain information within the database using a set of queries accessible through a GUI. The GUI that we implemented allows users to easily run a specific set of queries without needing to use SQL. Each class of user has different queries available for them to execute when using our application: for example, a customer can view their individual order history, while a restaurant can view the information of all orders placed with that restaurant.

The functionality of our application is very similar to how we described it in Milestone 1, though a few ideas we had in Milestone 1 had to be scrapped due to implementation complications and time constraints. The updated description of our application's functionality is as follows:

1. Customers are able to register an account with the app, modify their some of their account information (such as their address, name, and phone number), delete their account, place orders, view their order history, buy gift cards, and redeem/use coupons.

2. Couriers are able to register an account with the app (as well as their vehicle, bicycle, or bus pass, depending on the type of courier they register as), modify some of their account information (such as their name and phone number), delete their account, and view information regarding orders they have delivered.

3. Restaurants are able to register an account with the app, modify some of their account information (such as their category and name), delete their account, and access information about orders that have been placed with their restaurant (such as the list of menu items ordered). Restaurants must also register their menu with the app, and they have the ability to add/update/delete dishes from their menu at any time.

Changes We Made Between Milestones:

During Milestone 1, we planned to create our application using Oracle, Java, and JDBC. Upon beginning Milestone 3, we changed our minds and decided to use Oracle and PHP instead. Using PHP proved to be a minor challenge since no one in our group had ever used it before, but with the guidance of both the CPSC 304 tutorials and Google, we were able to complete the project with relatively few difficulties.

After Milestone 1, we made the following changes to the ER diagram for Milestone 2:
1. Removed the Menu_Items ISA (because it wasn't meaningful).
2. Renamed the 'Driver' entity to 'Courier'.

3. Added an ISA to the Courier entity (with sub-entities Foot_Courier, Bicycle_Courier, and Vehicle_Courier).
4. Added some new attributes to the Orders entity: food_subtotal, delivery_fee, and courier_tip (to facilitate the normalization requirement in Milestone 2).

From Milestone 2 to Milestone 3, we did not make any major changes to the schema. We did make a few minor adjustments to the data types used (e.g. changed the data type of the "date_placed" attribute in the Orders entity from 'string' to 'DATE'). We also had to change the name of our "Order" entity to "Orders", because "Order" happened to be a reserved keyword in Oracle.

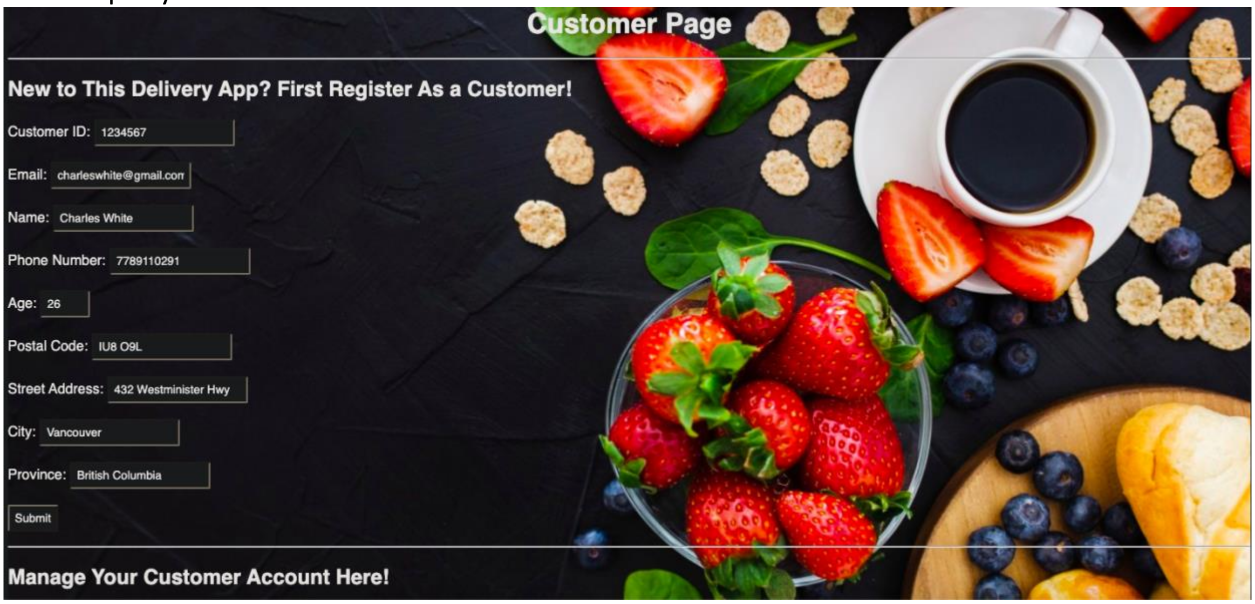
<u>SQL Script and Query Files</u>:
We used the databaseSetup.sql script to create and populate the tables for our database.
The SQL queries we used to satisfy the grading rubric can be found in the queries.sql file.

(Project screenshots start on the following page.)

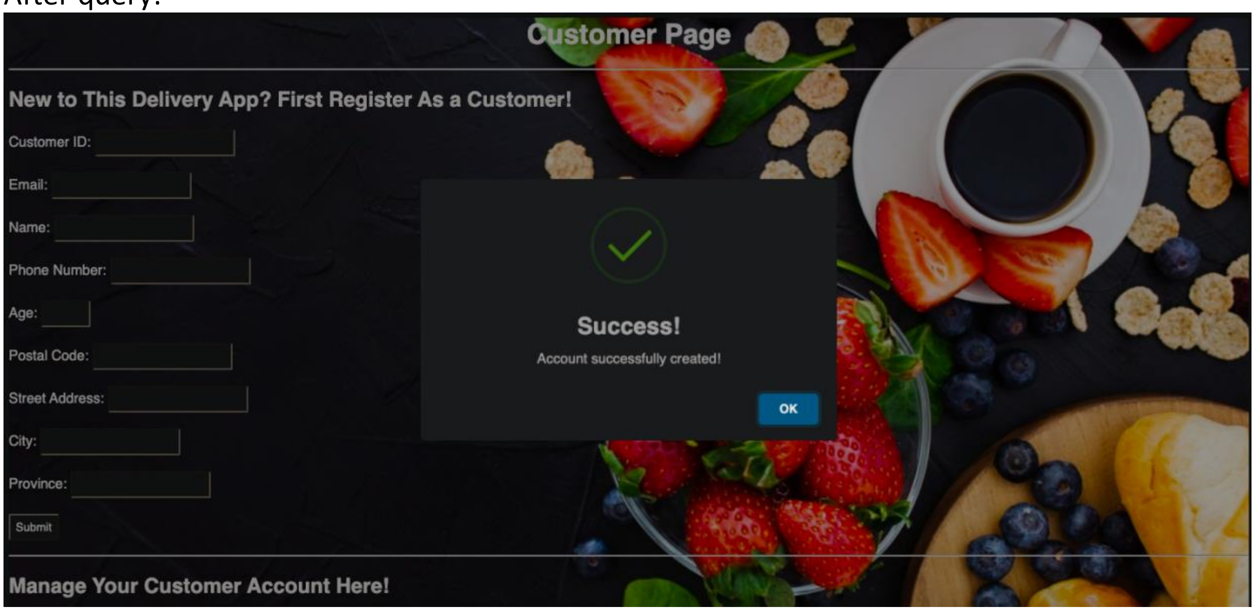
<u>SCREENSHOTS</u>:

1) Insertion Query

Before query:



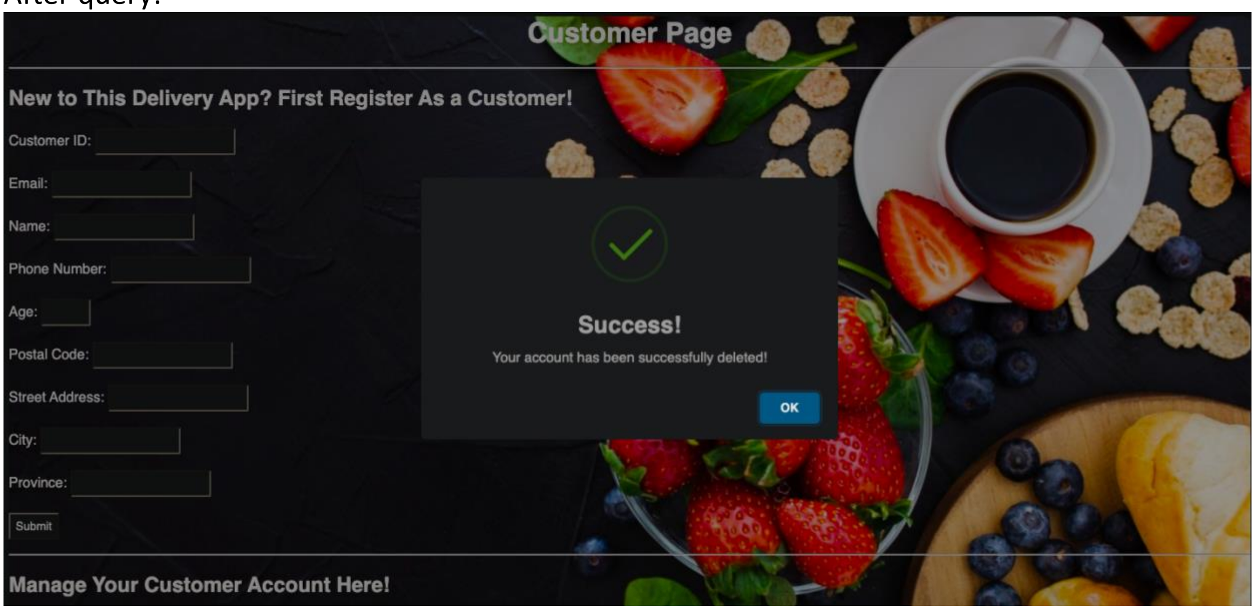
After query:



2) Deletion Query

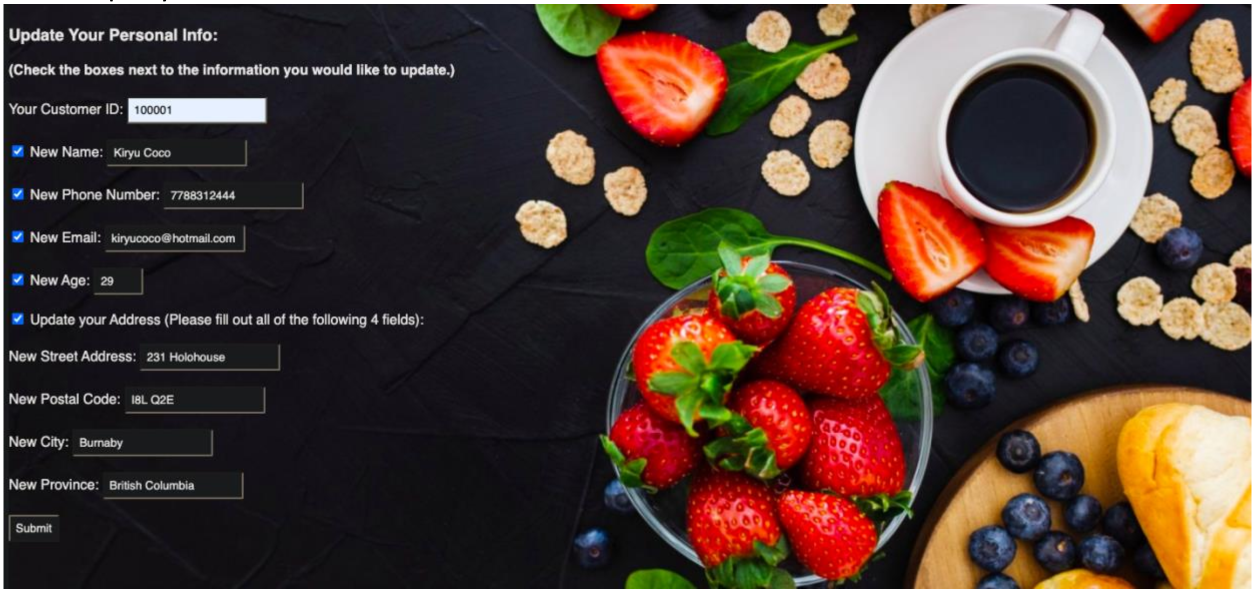
Before query:

After query:

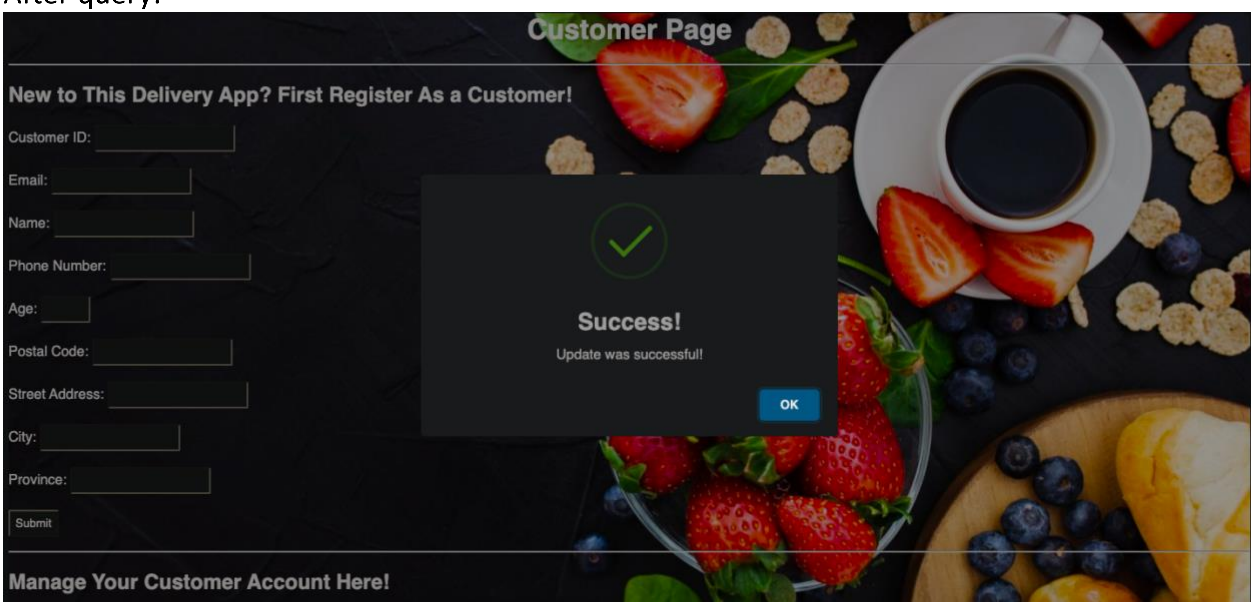


3) Update Query

Before query:

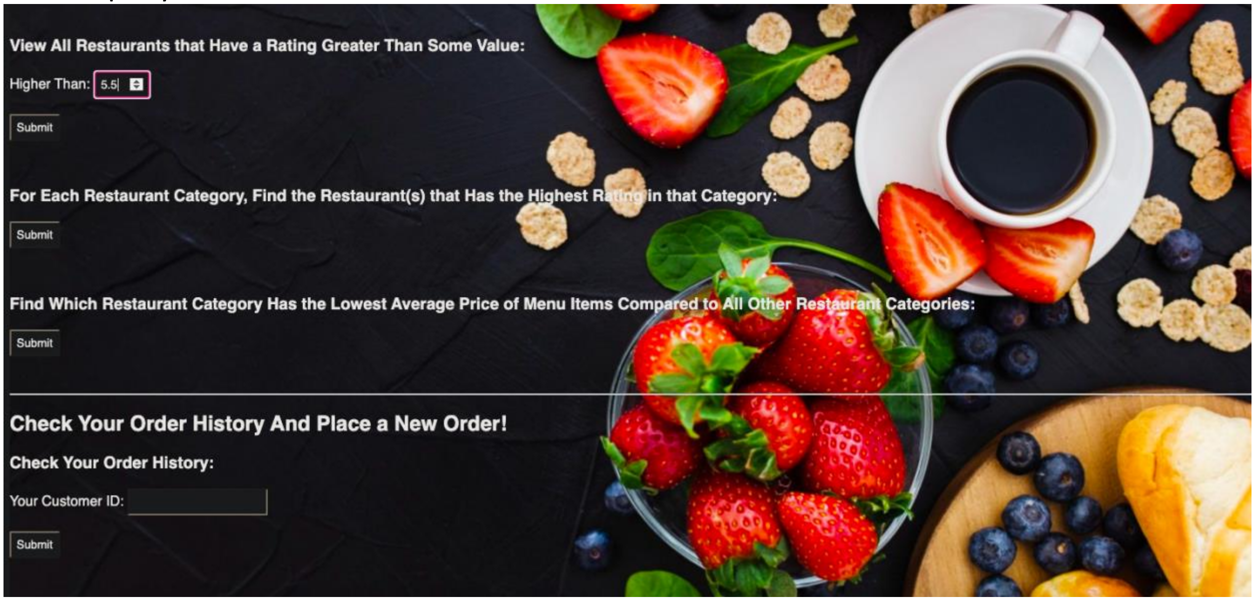


After query:



4) Selection Query

Before query:



After query:

5) Projection Query

Before query:

After query:

6) Join Query

Before query:

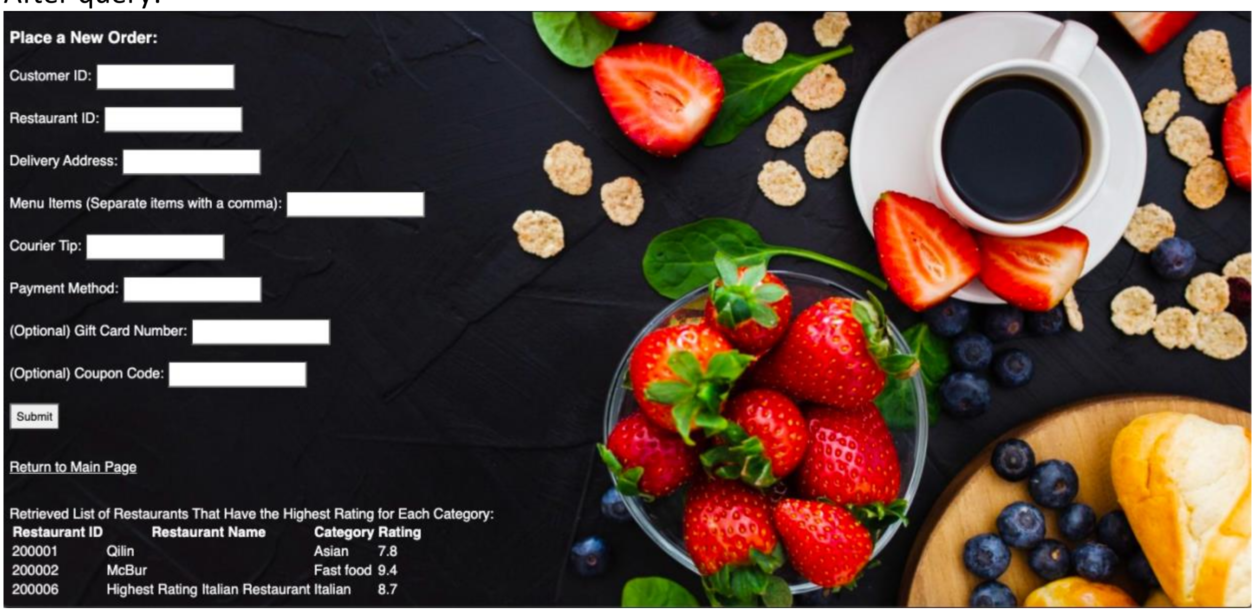
After query:

Average Food Subtotal	Restaurant ID	Name	Rating
22.445	200001	MIAMIA	7.8
68.74	200002	McBur	9.4
51.9725	200004	GrilledCk	8
38.0325	200005	Passione	6.7
38.575	200006	Highest Rating Italian Restaurant	8.7

7) Aggregation With Group By Query

Before query:

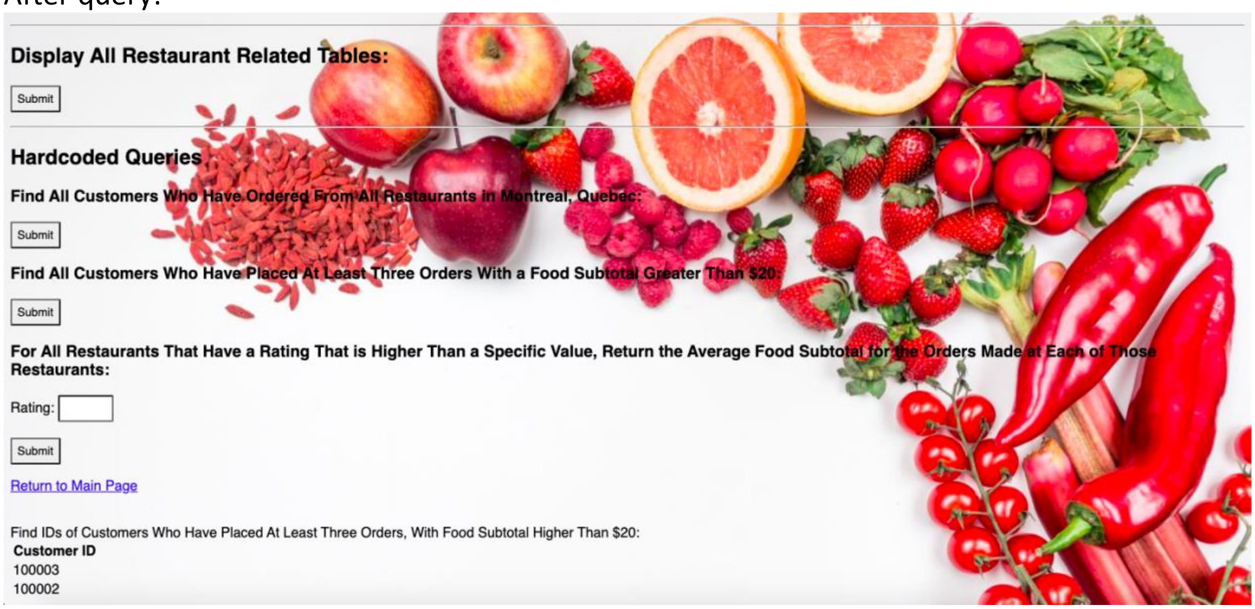
After query:



8) Aggregation With Having Query

Before query:

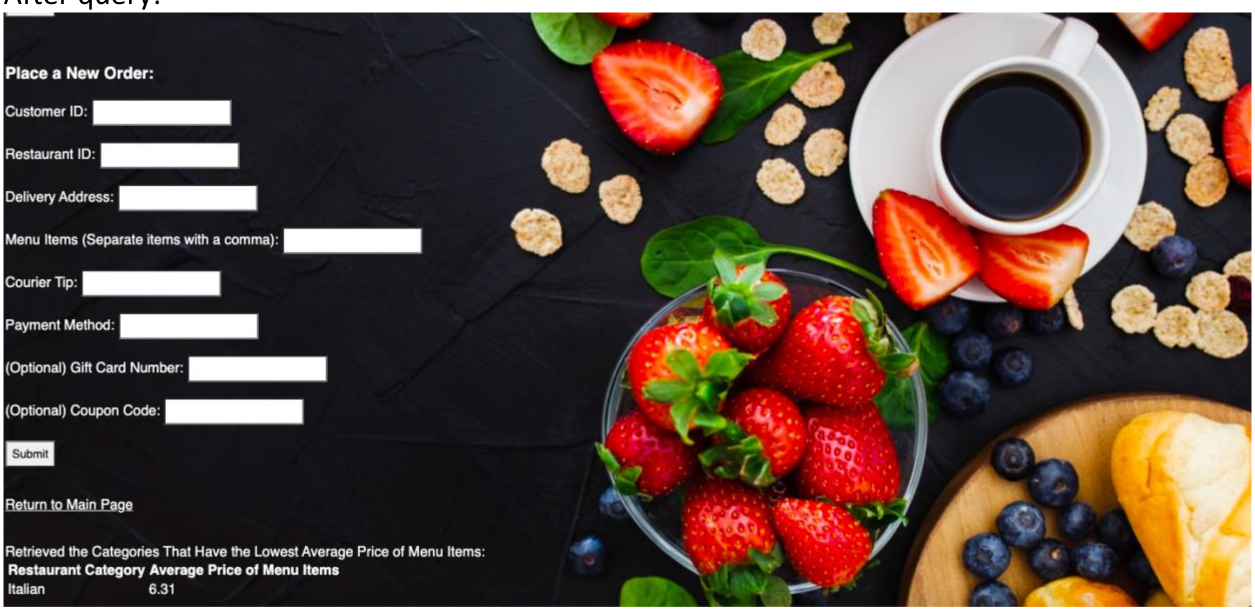
After query:



9) Nested Aggregation With Group By Query

Before query:

After query:



10) Division Query

Before query:

After query: