Online Restaurant Booking Platform

Table of Contents

A. Requirements Document

Project Description

The online restaurant booking platform will allow customers to search, view information, and make reservations at restaurants. Customers can filter restaurants by location, cuisine, budget, etc. and view restaurant details including menu, reviews, location. Customers can create accounts to make and manage reservations online.

The system will also allow restaurant owners/staff to manage their business information and availability. They can add menu items, update hours of operation, view and respond to customer reviews, and monitor upcoming reservations.

Business Rules

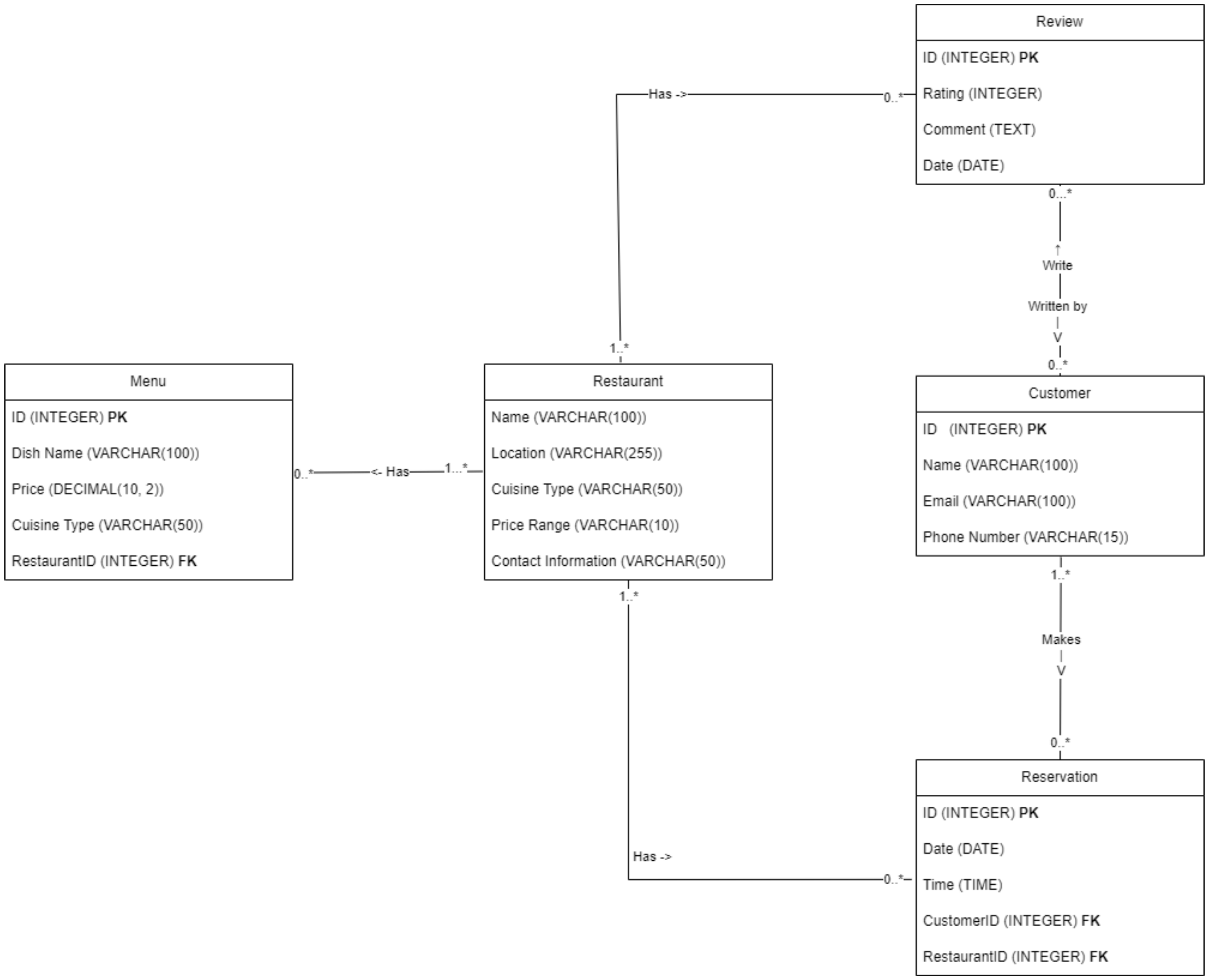
- Customers must create an account to make reservations
- Reservations require date, time, party size, and contact information
- Customers can write one review per restaurant
- Reviews require rating (1-5 stars) and optional comment
- Restaurants can add/update menu items with name, description, price
- Menu items can be categorized by cuisine type
- Restaurants set available reservation times/dates

Nouns:
Customer, Restaurant, Reservation, Review, Menu, Menu Item, Account

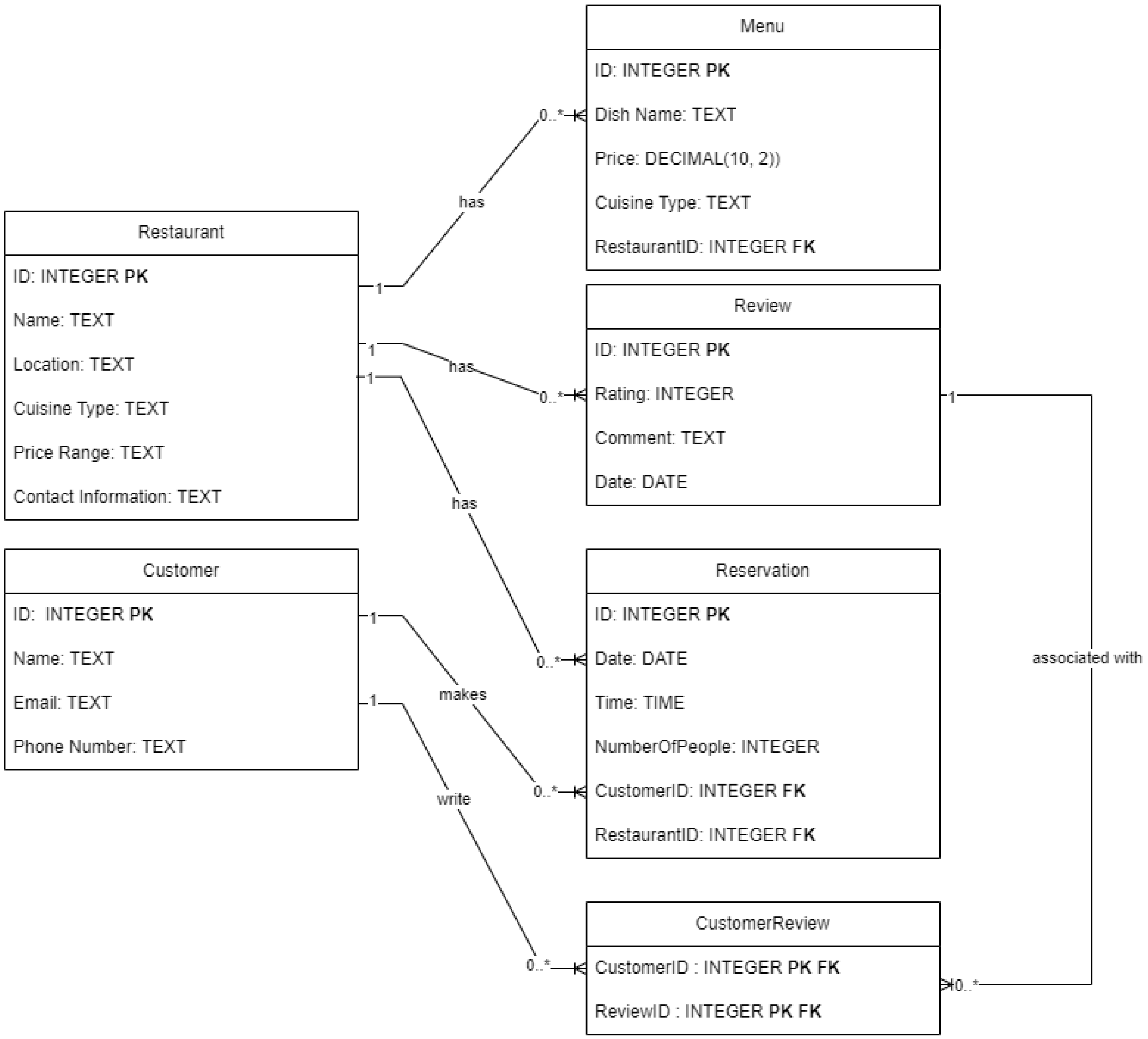
Verbs:
Search, View, Filter, Create, Make, Manage, Add, Update, Monitor, Set

This covers a high-level description of the online restaurant booking platform, key business rules, and identifies some initial nouns and verbs that will inform the data model. Let me know if you would like me to expand or modify this draft task 1 document in any way.

B. UML Class Diagram



C. ERD


Menu
ID: INTEGER PK
Dish Name: TEXT
Price: DECIMAL(10, 2))
Cuisine Type: TEXT
RestaurantID: INTEGER FK
Restaurant
ID: INTEGER PK
Name: TEXT
Location: TEXT
Cuisine Type: TEXT
Price Range: TEXT
Contact Information: TEXT
Review
ID: INTEGER PK
Rating: INTEGER
Comment: TEXT
Date: DATE
Customer
ID: INTEGER PK
Name: TEXT
Email: TEXT
Phone Number: TEXT
Reservation
ID: INTEGER PK
Date: DATE
Time: TIME
NumberOfPeople: INTEGER
CustomerID: INTEGER FK
RestaurantID: INTEGER FK
CustomerReview
CustomerID : INTEGER PK FK
ReviewID : INTEGER PK FK
has
has
has
makes
write
associated with
1
1
1
1
1
1
0..*
0..*
0..*
0..*
0..*
0..*

D. Definition of Relational Schema with BCNF Proof

Functional Dependencies:

- **Restaurant**: ID → Name, Location, CuisineType, PriceRange, ContactInfo
- **Customer**: ID→Name, Email, PhoneNumber
- **Reservation**: ID→Date, Time, NumberOfPeople, CustomerID, RestaurantID
- **Review**: ID→Rating, Comment, Date
- **Menu**: ID→DishName, Price, CuisineType, RestaurantID
- **CustomerReview**: CustomerID, ReviewID→∅ (This table is to resolve the many-to-many relationship, so there are no non-key attributes.)

BCNF Conditions:

A table is in BCNF if for every non-trivial functional dependency X→Y, X is a superkey.

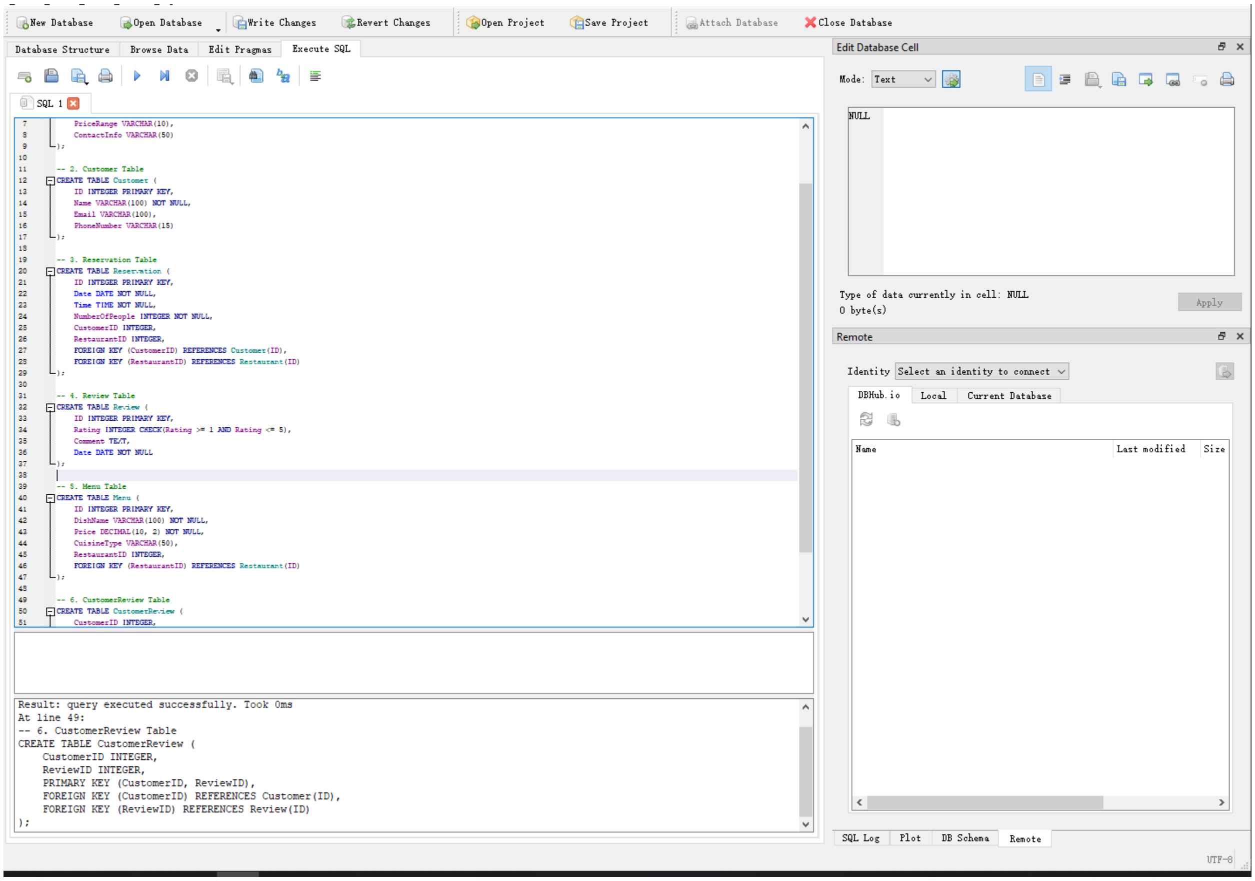
Verification:

- **Restaurant**: The determinant is the primary key, so it satisfies BCNF.
- **Customer**: The determinant is the primary key, so it satisfies BCNF.
- **Reservation**: The determinant is the primary key, so it satisfies BCNF.
- **Review**: The determinant is the primary key, so it satisfies BCNF.
- **Menu**: The determinant is the primary key, so it satisfies BCNF.
- **CustomerReview**: This table is used to resolve the many-to-many relationship and only contains foreign keys, so it satisfies BCNF.

Since all tables satisfy the BCNF conditions, the schema is in BCNF.

E. Create & Populate SQL Database

Create Tables (see `create_table.sql`)



```
    PriceRange VARCHAR(10),
    ContactInfo VARCHAR(50)
);

-- 2. Customer Table
CREATE TABLE Customer (
    ID INTEGER PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Email VARCHAR(100),
    PhoneNumber VARCHAR(15)
);

-- 3. Reservation Table
CREATE TABLE Reservation (
    ID INTEGER PRIMARY KEY,
    Date DATE NOT NULL,
    Time TIME NOT NULL,
    NumberOfPeople INTEGER NOT NULL,
    CustomerID INTEGER,
    RestaurantID INTEGER,
    FOREIGN KEY (CustomerID) REFERENCES Customer(ID),
    FOREIGN KEY (RestaurantID) REFERENCES Restaurant(ID)
);

-- 4. Review Table
CREATE TABLE Review (
    ID INTEGER PRIMARY KEY,
    Rating INTEGER CHECK(Rating >= 1 AND Rating <= 5),
    Comment TEXT,
    Date DATE NOT NULL
);

-- 5. Menu Table
CREATE TABLE Menu (
    ID INTEGER PRIMARY KEY,
    DishName VARCHAR(100) NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    CuisineType VARCHAR(50),
    RestaurantID INTEGER,
    FOREIGN KEY (RestaurantID) REFERENCES Restaurant(ID)
);

-- 6. CustomerReview Table
CREATE TABLE CustomerReview (
    CustomerID INTEGER,
```

```
Result: query executed successfully. Took 0ms
At line 49:
-- 6. CustomerReview Table
CREATE TABLE CustomerReview (
    CustomerID INTEGER,
    ReviewID INTEGER,
    PRIMARY KEY (CustomerID, ReviewID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(ID),
    FOREIGN KEY (ReviewID) REFERENCES Review(ID)
);
```

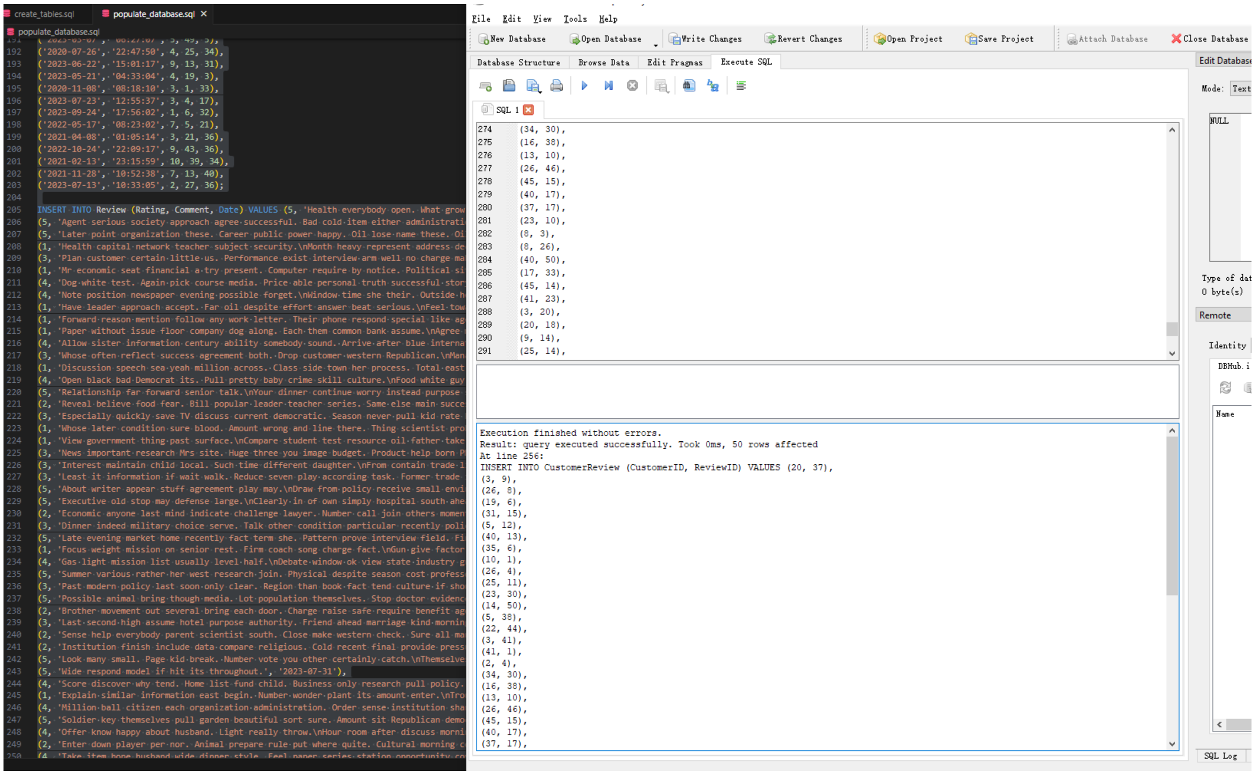
Mode: Text

NULL

Type of data currently in cell: NULL
0 byte(s)

Identity: Select an identity to connect

Populate Database(see `populate_database.sql`, `generate_data.py`)



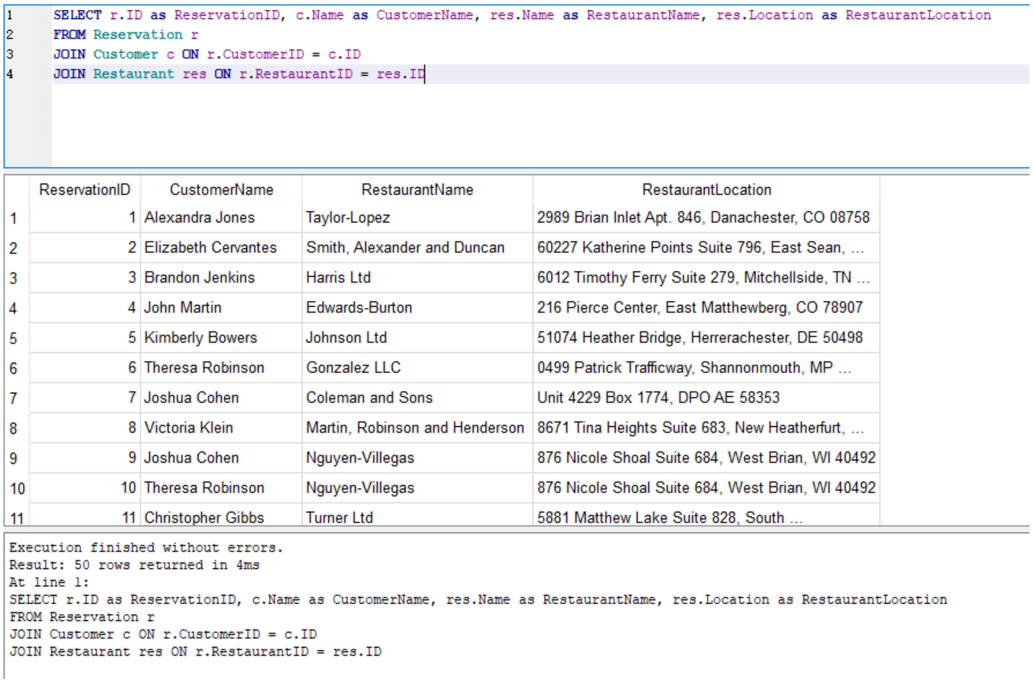
```
('2020-07-26', '22:47:50', 4, 25, 34),
('2023-06-22', '15:01:17', 9, 13, 31),
('2023-05-21', '04:33:04', 4, 19, 3),
('2020-11-08', '08:18:10', 3, 1, 33),
('2023-07-23', '12:55:37', 3, 4, 17),
('2023-09-24', '17:56:02', 1, 6, 32),
('2022-05-17', '08:23:02', 7, 5, 21),
('2021-04-08', '01:05:14', 3, 21, 36),
('2022-10-24', '22:09:17', 9, 43, 36),
('2021-02-13', '23:15:59', 10, 39, 34),
('2021-11-28', '10:52:38', 7, 13, 40),
('2023-07-13', '10:33:05', 2, 27, 36);

INSERT INTO Review (Rating, Comment, Date) VALUES (5, 'Health everybody open. What grow
(5, 'Agent serious society approach agree successful. Bad cold item either administrati
(5, 'Later point organization these. Career public power happy. Oil lose name these. Oi
(1, 'Health capital network teacher subject security.\nMonth heavy represent address de
(3, 'Plan customer certain little us. Performance exist interview arm well no charge ma
(1, 'Mr economic seat financial a try present. Computer require by notice. Political si
(4, 'Dog white test. Again pick course media. Price able personal truth successful stor
(4, 'Note position newspaper evening possible forget.\nWindow time she their. Outside h
(1, 'Have leader approach accept. Far oil despite effort answer beat serious.\nFeel tow
(1, 'Forward reason mention follow any work letter. Their phone respond special like ag
(1, 'Paper without issue floor company dog along. Each them common bank assume.\nAgree
(4, 'Allow sister information century ability somebody sound. Arrive after blue interna
(3, 'Whose often reflect success agreement both. Drop customer western Republican.\nMan
(1, 'Discussion speech sea yeah million across. Class side town her process. Total east
(4, 'Open black bad Democrat its. Pull pretty baby crime skill culture.\nFood white guy
(5, 'Relationship far forward senior talk.\nYour dinner continue worry instead purpose
(2, 'Reveal believe food fear. Bill popular leader teacher series. Same else main succe
(3, 'Especially quickly save TV discuss current democratic. Season never pull kid rate
(1, 'Whose later condition sure blood. Amount wrong and line there. Thing scientist pro
(1, 'View government thing past surface.\nCompare student test resource oil father take
(3, 'News important research Mrs site. Huge three you image budget. Product help born P
(3, 'Interest maintain child local. Such time different daughter.\nFrom contain trade l
(3, 'Least it information if wait walk. Reduce seven play according task. Former trade
(5, 'About writer appear stuff agreement play may.\nDraw from policy receive small envi
(5, 'Executive old stop may defense large.\nClearly in of own simply hospital south ahe
(2, 'Economic anyone last mind indicate challenge lawyer. Number call join others momen
(3, 'Dinner indeed military choice serve. Talk other condition particular recently poli
(5, 'Late evening market home recently fact term she. Pattern prove interview field. Fi
(1, 'Focus weight mission on senior rest. Firm coach song charge fact.\nGun give factor
(4, 'Gas light mission list usually level half.\nDebate window ok view state industry g
(5, 'Summer various rather her west research join. Physical despite season cost profess
(3, 'Past modern policy last soon only clear. Region than book fact tend culture if sho
(5, 'Possible animal bring though media. Lot population themselves. Stop doctor evidenc
(2, 'Brother movement out several bring each door. Charge raise safe require benefit ag
(3, 'Last second high assume hotel purpose authority. Friend ahead marriage kind mornin
(2, 'Sense help everybody parent scientist south. Close make western check. Sure all ma
(2, 'Institution finish include data compare religious. Cold recent final provide press
(5, 'Look many small. Page kid break. Number vote you other certainly catch.\nThemselve
(5, 'Wide respond model if hit its throughout.', '2023-07-31'),
(4, 'Score discover why tend. Home list fund child. Business only research pull policy.
(1, 'Explain similar information east begin. Number wonder plant its amount enter.\nTro
(4, 'Million ball citizen each organization administration. Order sense institution sha
(5, 'Soldier key themselves pull garden beautiful sort sure. Amount sit Republican demo
(4, 'Offer know happy about husband. Light really throw.\nHour room after discuss morni
(2, 'Enter down player per nor. Animal prepare rule put where quite. Cultural morning c
(4, 'Take item hope husband wide dinner style. Feel paper series station opportunity co
```

```
(34, 30),
(16, 38),
(13, 10),
(26, 46),
(45, 15),
(40, 17),
(37, 17),
(23, 10),
(8, 3),
(8, 26),
(40, 50),
(17, 33),
(45, 14),
(41, 23),
(3, 20),
(20, 18),
(9, 14),
(25, 14),
```

```
Execution finished without errors.
Result: query executed successfully. Took 0ms, 50 rows affected
At line 256:
INSERT INTO CustomerReview (CustomerID, ReviewID) VALUES (20, 37),
(3, 9),
(26, 8),
(19, 6),
(31, 15),
(5, 12),
(40, 13),
(35, 6),
(10, 1),
(26, 4),
(25, 11),
(23, 30),
(14, 50),
(5, 38),
(22, 44),
(3, 41),
(41, 1),
(2, 4),
(34, 30),
(16, 38),
(13, 10),
(26, 46),
(45, 15),
(40, 17),
(37, 17),
```

Query 1: Join of at least three tables

Join the Reservation, Customer, and Restaurant tables to get a list of reservations, including customer names and restaurant details.(see `queries.sql`)



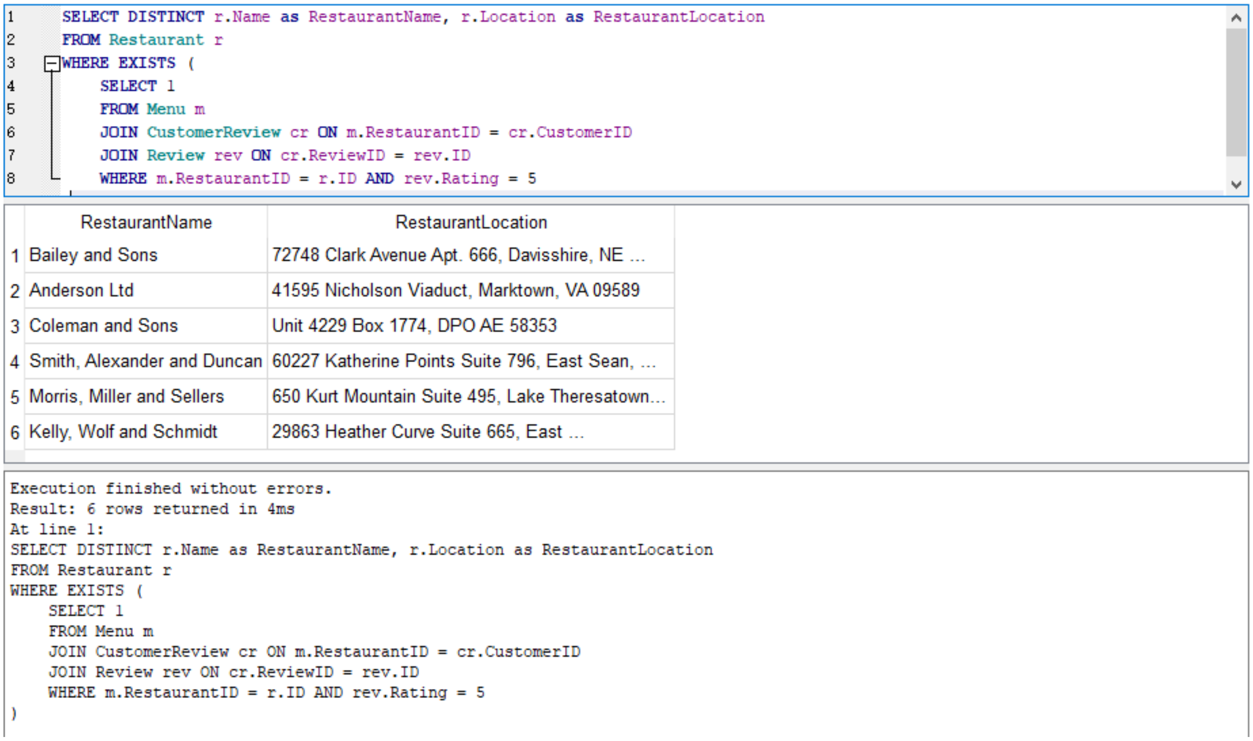
```sql
SELECT r.ID as ReservationID, c.Name as CustomerName, res.Name as RestaurantName, res.Location as RestaurantLocation
FROM Reservation r
JOIN Customer c ON r.CustomerID = c.ID
JOIN Restaurant res ON r.RestaurantID = res.ID
```

	ReservationID	CustomerName	RestaurantName	RestaurantLocation
1	1	Alexandra Jones	Taylor-Lopez	2989 Brian Inlet Apt. 846, Danachester, CO 08758
2	2	Elizabeth Cervantes	Smith, Alexander and Duncan	60227 Katherine Points Suite 796, East Sean, ...
3	3	Brandon Jenkins	Harris Ltd	6012 Timothy Ferry Suite 279, Mitchellside, TN ...
4	4	John Martin	Edwards-Burton	216 Pierce Center, East Matthewberg, CO 78907
5	5	Kimberly Bowers	Johnson Ltd	51074 Heather Bridge, Herrerachester, DE 50498
6	6	Theresa Robinson	Gonzalez LLC	0499 Patrick Trafficway, Shannonmouth, MP ...
7	7	Joshua Cohen	Coleman and Sons	Unit 4229 Box 1774, DPO AE 58353
8	8	Victoria Klein	Martin, Robinson and Henderson	8671 Tina Heights Suite 683, New Heatherfurt, ...
9	9	Joshua Cohen	Nguyen-Villegas	876 Nicole Shoal Suite 684, West Brian, WI 40492
10	10	Theresa Robinson	Nguyen-Villegas	876 Nicole Shoal Suite 684, West Brian, WI 40492
11	11	Christopher Gibbs	Turner Ltd	5881 Matthew Lake Suite 828, South ...

```
Execution finished without errors.
Result: 50 rows returned in 4ms
At line 1:
SELECT r.ID as ReservationID, c.Name as CustomerName, res.Name as RestaurantName, res.Location as RestaurantLocation
FROM Reservation r
JOIN Customer c ON r.CustomerID = c.ID
JOIN Restaurant res ON r.RestaurantID = res.ID
```

Query 2: Subquery

Find all restaurants that have received a review with a rating of 5 (see `queries.sql`)



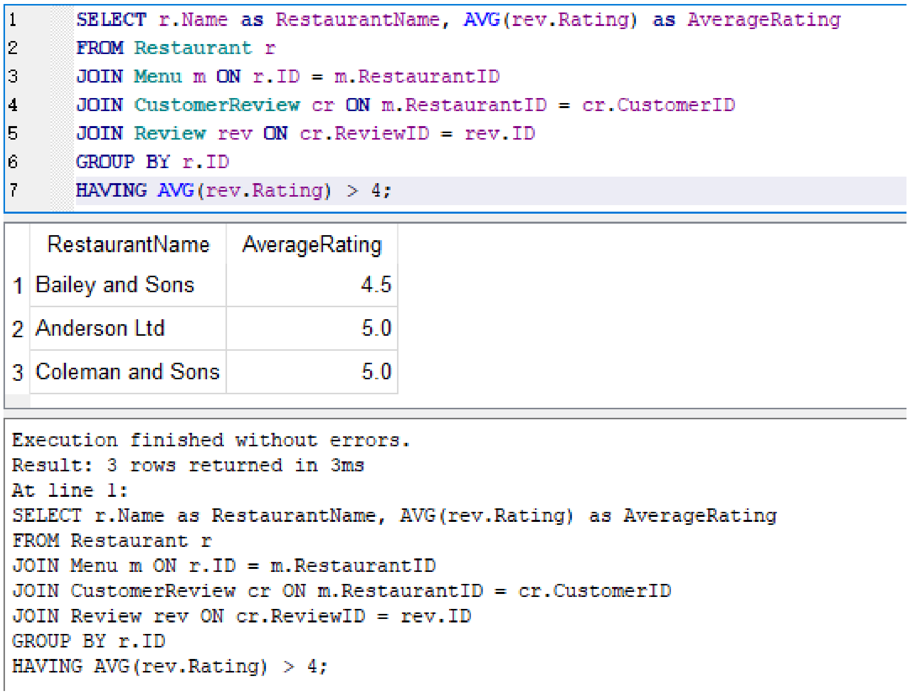
```sql
SELECT DISTINCT r.Name as RestaurantName, r.Location as RestaurantLocation
FROM Restaurant r
WHERE EXISTS (
    SELECT 1
    FROM Menu m
    JOIN CustomerReview cr ON m.RestaurantID = cr.CustomerID
    JOIN Review rev ON cr.ReviewID = rev.ID
    WHERE m.RestaurantID = r.ID AND rev.Rating = 5
```

	RestaurantName	RestaurantLocation
1	Bailey and Sons	72748 Clark Avenue Apt. 666, Davisshire, NE ...
2	Anderson Ltd	41595 Nicholson Viaduct, Marktown, VA 09589
3	Coleman and Sons	Unit 4229 Box 1774, DPO AE 58353
4	Smith, Alexander and Duncan	60227 Katherine Points Suite 796, East Sean, ...
5	Morris, Miller and Sellers	650 Kurt Mountain Suite 495, Lake Theresatown...
6	Kelly, Wolf and Schmidt	29863 Heather Curve Suite 665, East ...

```
Execution finished without errors.
Result: 6 rows returned in 4ms
At line 1:
SELECT DISTINCT r.Name as RestaurantName, r.Location as RestaurantLocation
FROM Restaurant r
WHERE EXISTS (
    SELECT 1
    FROM Menu m
    JOIN CustomerReview cr ON m.RestaurantID = cr.CustomerID
    JOIN Review rev ON cr.ReviewID = rev.ID
    WHERE m.RestaurantID = r.ID AND rev.Rating = 5
)
```

Query 3: Group By with a Having Clause

Group the reviews by restaurant and find the restaurants that have an average rating higher than 4.(see `queries.sql`)



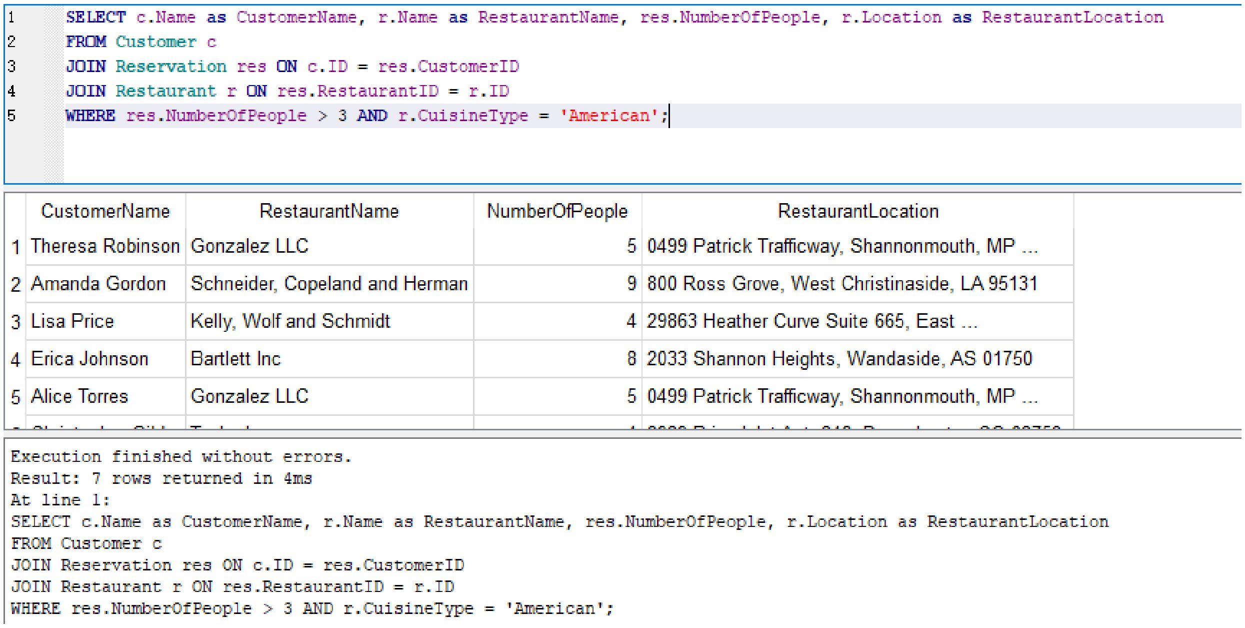
```sql
SELECT r.Name as RestaurantName, AVG(rev.Rating) as AverageRating
FROM Restaurant r
JOIN Menu m ON r.ID = m.RestaurantID
JOIN CustomerReview cr ON m.RestaurantID = cr.CustomerID
JOIN Review rev ON cr.ReviewID = rev.ID
GROUP BY r.ID
HAVING AVG(rev.Rating) > 4;
```

	RestaurantName	AverageRating
1	Bailey and Sons	4.5
2	Anderson Ltd	5.0
3	Coleman and Sons	5.0

```
Execution finished without errors.
Result: 3 rows returned in 3ms
At line 1:
SELECT r.Name as RestaurantName, AVG(rev.Rating) as AverageRating
FROM Restaurant r
JOIN Menu m ON r.ID = m.RestaurantID
JOIN CustomerReview cr ON m.RestaurantID = cr.CustomerID
JOIN Review rev ON cr.ReviewID = rev.ID
GROUP BY r.ID
HAVING AVG(rev.Rating) > 4;
```

Query 4: Complex search criterion

Find all customers who have made a reservation for more than 3 people at an American restaurant. (see `queries.sql`)



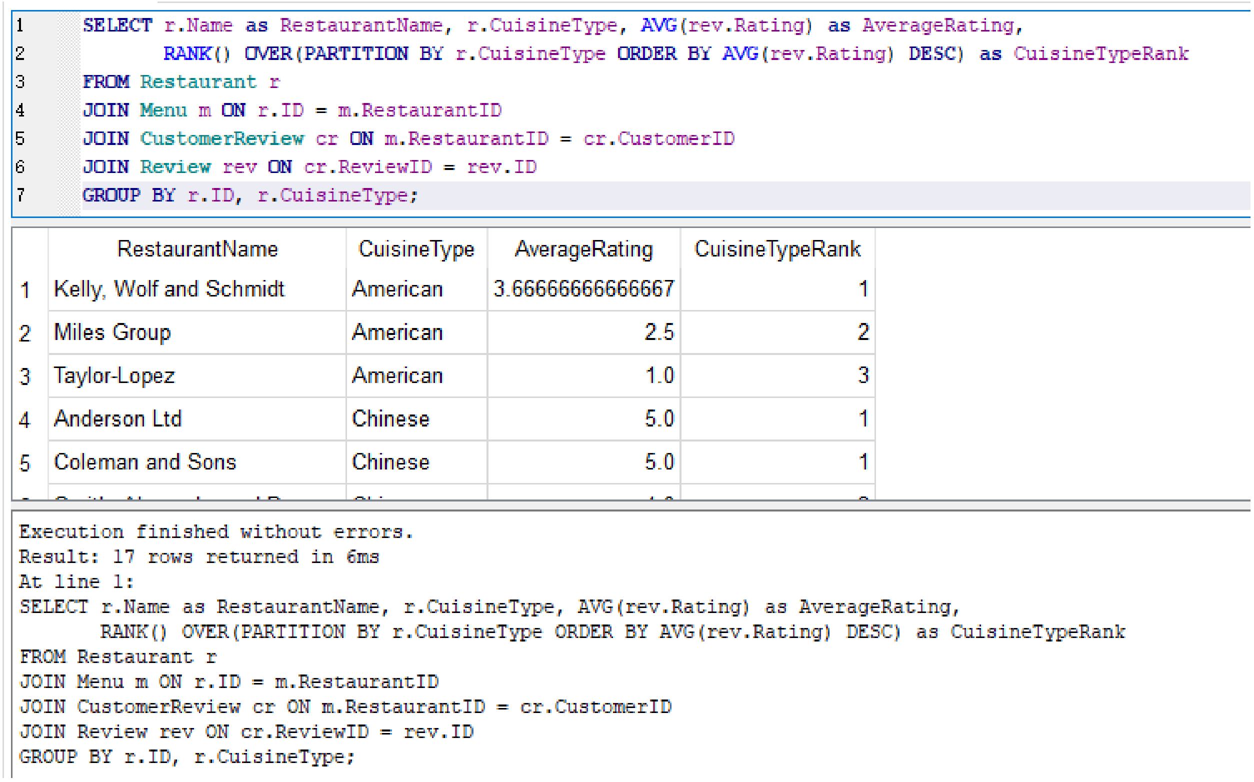
```sql
SELECT c.Name as CustomerName, r.Name as RestaurantName, res.NumberOfPeople, r.Location as RestaurantLocation
FROM Customer c
JOIN Reservation res ON c.ID = res.CustomerID
JOIN Restaurant r ON res.RestaurantID = r.ID
WHERE res.NumberOfPeople > 3 AND r.CuisineType = 'American';
```

	CustomerName	RestaurantName	NumberOfPeople	RestaurantLocation
1	Theresa Robinson	Gonzalez LLC	5	0499 Patrick Trafficway, Shannonmouth, MP ...
2	Amanda Gordon	Schneider, Copeland and Herman	9	800 Ross Grove, West Christinaside, LA 95131
3	Lisa Price	Kelly, Wolf and Schmidt	4	29863 Heather Curve Suite 665, East ...
4	Erica Johnson	Bartlett Inc	8	2033 Shannon Heights, Wandaside, AS 01750
5	Alice Torres	Gonzalez LLC	5	0499 Patrick Trafficway, Shannonmouth, MP ...

```
Execution finished without errors.
Result: 7 rows returned in 4ms
At line 1:
SELECT c.Name as CustomerName, r.Name as RestaurantName, res.NumberOfPeople, r.Location as RestaurantLocation
FROM Customer c
JOIN Reservation res ON c.ID = res.CustomerID
JOIN Restaurant r ON res.RestaurantID = r.ID
WHERE res.NumberOfPeople > 3 AND r.CuisineType = 'American';
```

Query 5: Experiment with Advanced Query Mechanisms:

Use PARTITION BY to rank restaurants within each cuisine type based on their average review rating.(see `queries.sql`)



```sql
SELECT r.Name as RestaurantName, r.CuisineType, AVG(rev.Rating) as AverageRating,
       RANK() OVER(PARTITION BY r.CuisineType ORDER BY AVG(rev.Rating) DESC) as CuisineTypeRank
FROM Restaurant r
JOIN Menu m ON r.ID = m.RestaurantID
JOIN CustomerReview cr ON m.RestaurantID = cr.CustomerID
JOIN Review rev ON cr.ReviewID = rev.ID
GROUP BY r.ID, r.CuisineType;
```

	RestaurantName	CuisineType	AverageRating	CuisineTypeRank
1	Kelly, Wolf and Schmidt	American	3.66666666666667	1
2	Miles Group	American	2.5	2
3	Taylor-Lopez	American	1.0	3
4	Anderson Ltd	Chinese	5.0	1
5	Coleman and Sons	Chinese	5.0	1

```
Execution finished without errors.
Result: 17 rows returned in 6ms
At line 1:
SELECT r.Name as RestaurantName, r.CuisineType, AVG(rev.Rating) as AverageRating,
       RANK() OVER(PARTITION BY r.CuisineType ORDER BY AVG(rev.Rating) DESC) as CuisineTypeRank
FROM Restaurant r
JOIN Menu m ON r.ID = m.RestaurantID
JOIN CustomerReview cr ON m.RestaurantID = cr.CustomerID
JOIN Review rev ON cr.ReviewID = rev.ID
GROUP BY r.ID, r.CuisineType;
```

F. Application Demo(See folder for source code: `8.Application`)

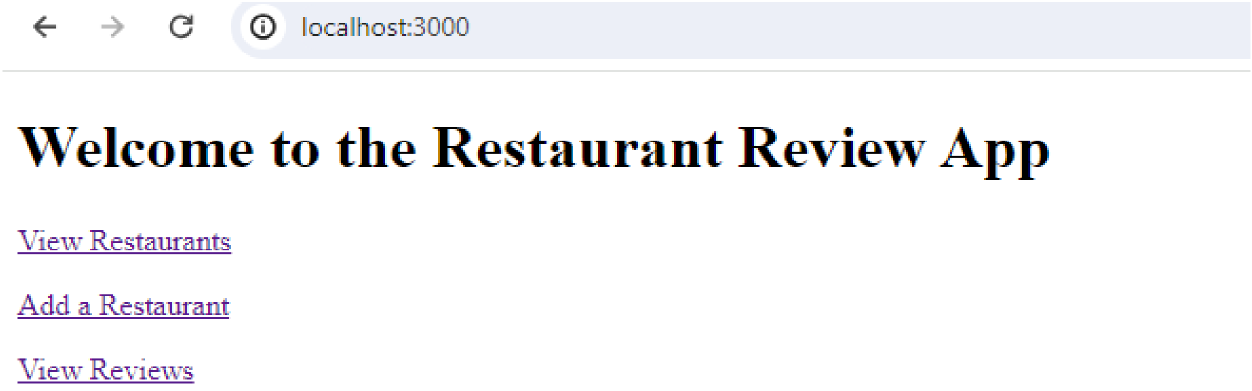
How to run the app

1. Enter the following command in the terminal:

```
cd 8.Application
node app.js
```

2. Open http://localhost:3000 in your browser.

Default Page



localhost:3000

Welcome to the Restaurant Review App

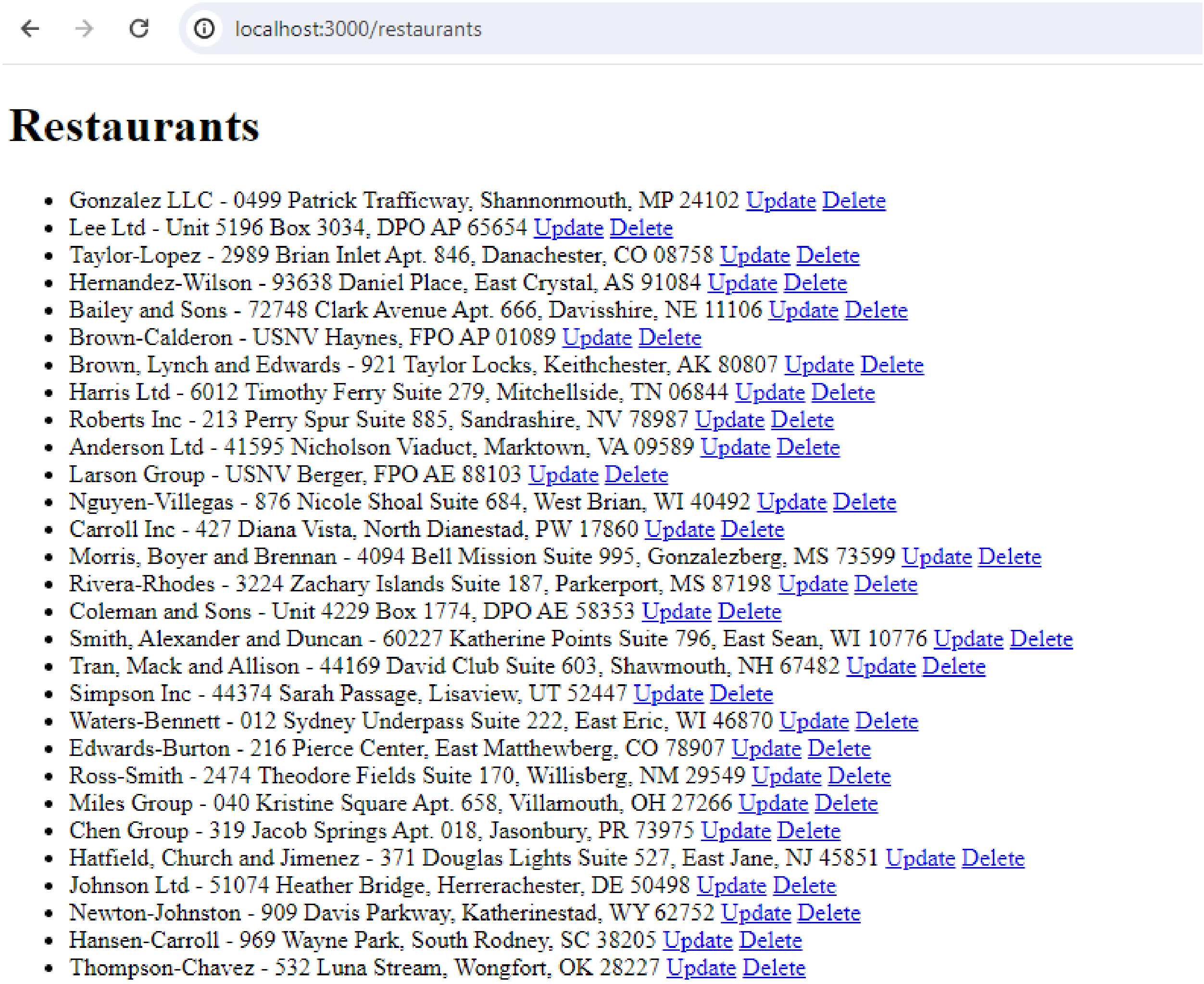
View Restaurants

Add a Restaurant

View Reviews

List all Restaurant

Click View Restaurants, the following page will show up:



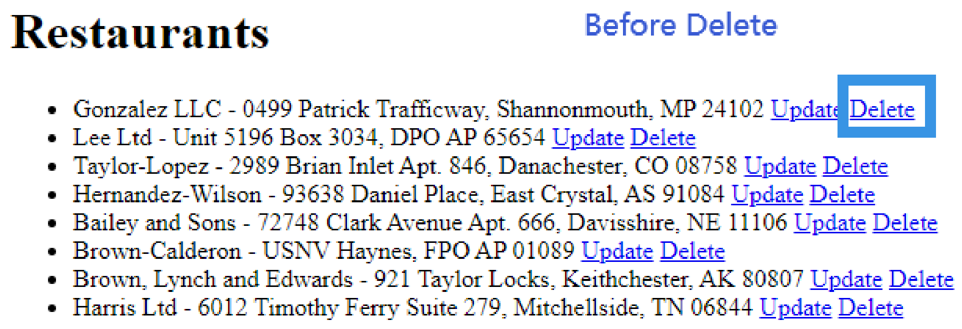
localhost:3000/restaurants

Restaurants

- Gonzalez LLC - 0499 Patrick Trafficway, Shannonmouth, MP 24102 Update Delete
- Lee Ltd - Unit 5196 Box 3034, DPO AP 65654 Update Delete
- Taylor-Lopez - 2989 Brian Inlet Apt. 846, Danachester, CO 08758 Update Delete
- Hernandez-Wilson - 93638 Daniel Place, East Crystal, AS 91084 Update Delete
- Bailey and Sons - 72748 Clark Avenue Apt. 666, Davisshire, NE 11106 Update Delete
- Brown-Calderon - USNV Haynes, FPO AP 01089 Update Delete
- Brown, Lynch and Edwards - 921 Taylor Locks, Keithchester, AK 80807 Update Delete
- Harris Ltd - 6012 Timothy Ferry Suite 279, Mitchellside, TN 06844 Update Delete
- Roberts Inc - 213 Perry Spur Suite 885, Sandrashire, NV 78987 Update Delete
- Anderson Ltd - 41595 Nicholson Viaduct, Marktown, VA 09589 Update Delete
- Larson Group - USNV Berger, FPO AE 88103 Update Delete
- Nguyen-Villegas - 876 Nicole Shoal Suite 684, West Brian, WI 40492 Update Delete
- Carroll Inc - 427 Diana Vista, North Dianestad, PW 17860 Update Delete
- Morris, Boyer and Brennan - 4094 Bell Mission Suite 995, Gonzalezberg, MS 73599 Update Delete
- Rivera-Rhodes - 3224 Zachary Islands Suite 187, Parkerport, MS 87198 Update Delete
- Coleman and Sons - Unit 4229 Box 1774, DPO AE 58353 Update Delete
- Smith, Alexander and Duncan - 60227 Katherine Points Suite 796, East Sean, WI 10776 Update Delete
- Tran, Mack and Allison - 44169 David Club Suite 603, Shawmouth, NH 67482 Update Delete
- Simpson Inc - 44374 Sarah Passage, Lisaview, UT 52447 Update Delete
- Waters-Bennett - 012 Sydney Underpass Suite 222, East Eric, WI 46870 Update Delete
- Edwards-Burton - 216 Pierce Center, East Matthewberg, CO 78907 Update Delete
- Ross-Smith - 2474 Theodore Fields Suite 170, Willisberg, NM 29549 Update Delete
- Miles Group - 040 Kristine Square Apt. 658, Villamouth, OH 27266 Update Delete
- Chen Group - 319 Jacob Springs Apt. 018, Jasonbury, PR 73975 Update Delete
- Hatfield, Church and Jimenez - 371 Douglas Lights Suite 527, East Jane, NJ 45851 Update Delete
- Johnson Ltd - 51074 Heather Bridge, Herrerachester, DE 50498 Update Delete
- Newton-Johnston - 909 Davis Parkway, Katherinestad, WY 62752 Update Delete
- Hansen-Carroll - 969 Wayne Park, South Rodney, SC 38205 Update Delete
- Thompson-Chavez - 532 Luna Stream, Wongfort, OK 28227 Update Delete

Delete Restaurant

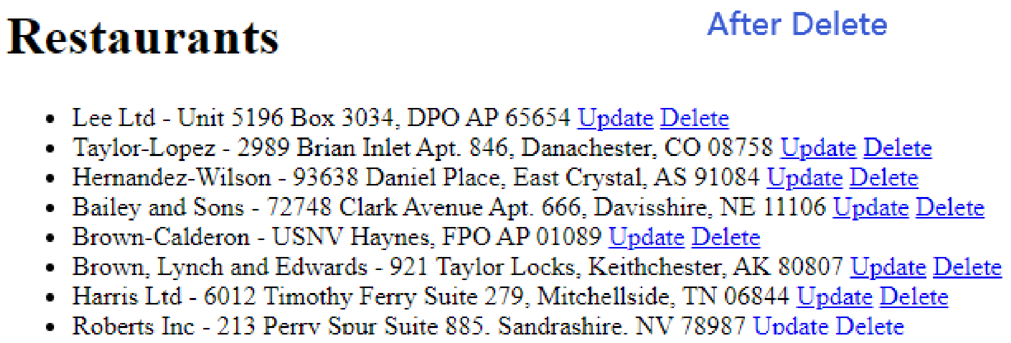
Click “Delete” next to the restaurant you want to remove, the page will refresh with a new list of restaurants.



Before Delete

Restaurants

- Gonzalez LLC - 0499 Patrick Trafficway, Shannonmouth, MP 24102 Update Delete
- Lee Ltd - Unit 5196 Box 3034, DPO AP 65654 Update Delete
- Taylor-Lopez - 2989 Brian Inlet Apt. 846, Danachester, CO 08758 Update Delete
- Hernandez-Wilson - 93638 Daniel Place, East Crystal, AS 91084 Update Delete
- Bailey and Sons - 72748 Clark Avenue Apt. 666, Davisshire, NE 11106 Update Delete
- Brown-Calderon - USNV Haynes, FPO AP 01089 Update Delete
- Brown, Lynch and Edwards - 921 Taylor Locks, Keithchester, AK 80807 Update Delete
- Harris Ltd - 6012 Timothy Ferry Suite 279, Mitchellside, TN 06844 Update Delete



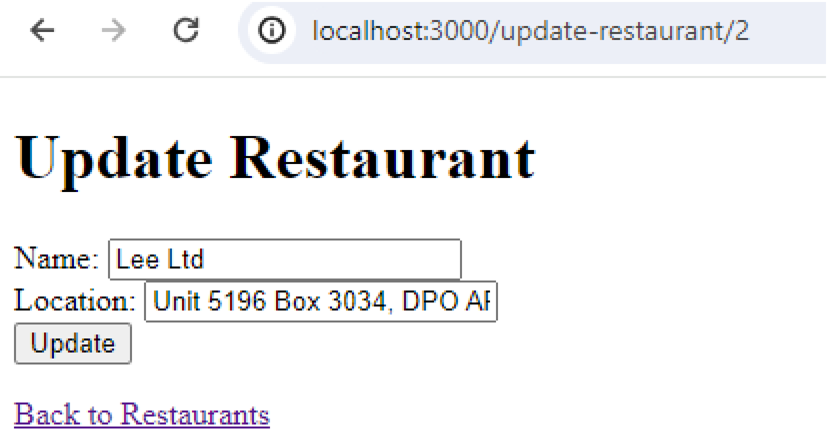
After Delete

Restaurants

- Lee Ltd - Unit 5196 Box 3034, DPO AP 65654 Update Delete
- Taylor-Lopez - 2989 Brian Inlet Apt. 846, Danachester, CO 08758 Update Delete
- Hernandez-Wilson - 93638 Daniel Place, East Crystal, AS 91084 Update Delete
- Bailey and Sons - 72748 Clark Avenue Apt. 666, Davisshire, NE 11106 Update Delete
- Brown-Calderon - USNV Haynes, FPO AP 01089 Update Delete
- Brown, Lynch and Edwards - 921 Taylor Locks, Keithchester, AK 80807 Update Delete
- Harris Ltd - 6012 Timothy Ferry Suite 279, Mitchellside, TN 06844 Update Delete
- Roberts Inc - 213 Perry Spur Suite 885, Sandrashire, NV 78987 Update Delete

Update Restaurant

Click “Update” next to the restaurant you want to remove, you will be directed to the following page:



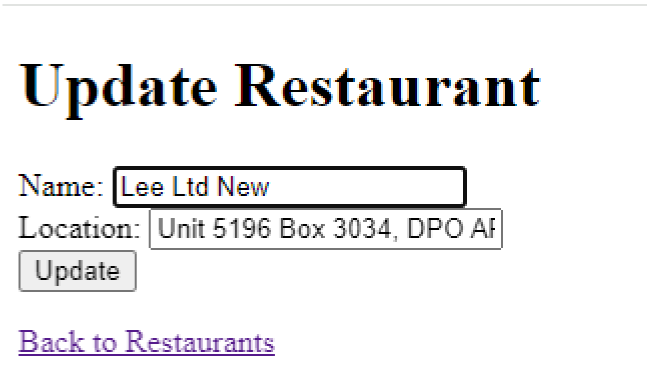
localhost:3000/update-restaurant/2

Update Restaurant

Name: Lee Ltd

Location: Unit 5196 Box 3034, DPO AP

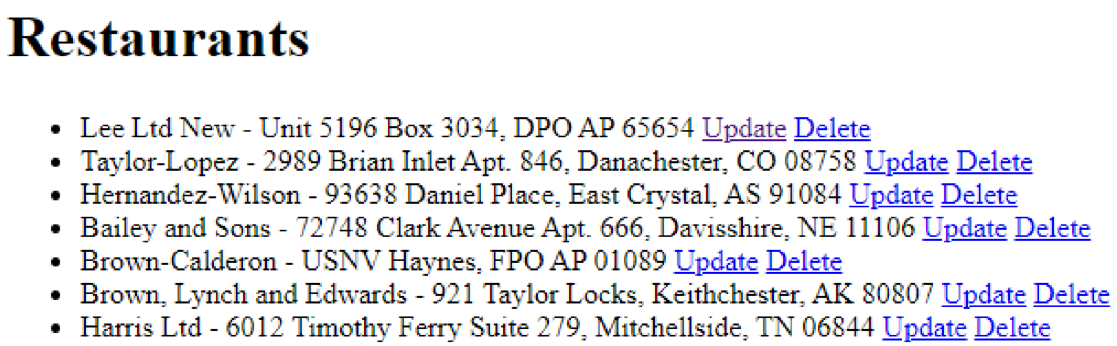
Enter the new information and click “Update”, the restaurant information will be updated.



Update Restaurant

Name: Lee Ltd New

Location: Unit 5196 Box 3034, DPO AF

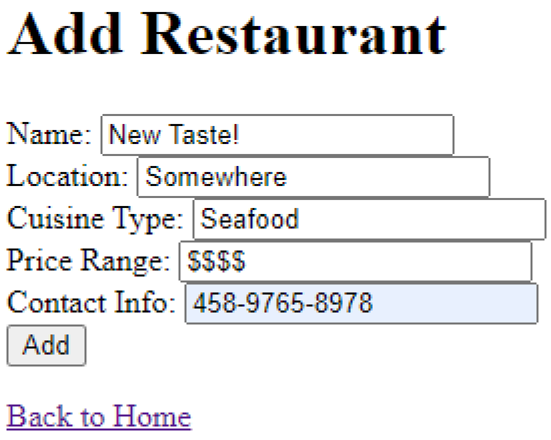


Restaurants

- Lee Ltd New - Unit 5196 Box 3034, DPO AP 65654 Update Delete
- Taylor-Lopez - 2989 Brian Inlet Apt. 846, Danachester, CO 08758 Update Delete
- Hernandez-Wilson - 93638 Daniel Place, East Crystal, AS 91084 Update Delete
- Bailey and Sons - 72748 Clark Avenue Apt. 666, Davisshire, NE 11106 Update Delete
- Brown-Calderon - USNV Haynes, FPO AP 01089 Update Delete
- Brown, Lynch and Edwards - 921 Taylor Locks, Keithchester, AK 80807 Update Delete
- Harris Ltd - 6012 Timothy Ferry Suite 279, Mitchellside, TN 06844 Update Delete

Add new restaurant

At homepage, click “Add a Restaurant”, the following page will appear:



Add Restaurant

Name: New Taste!

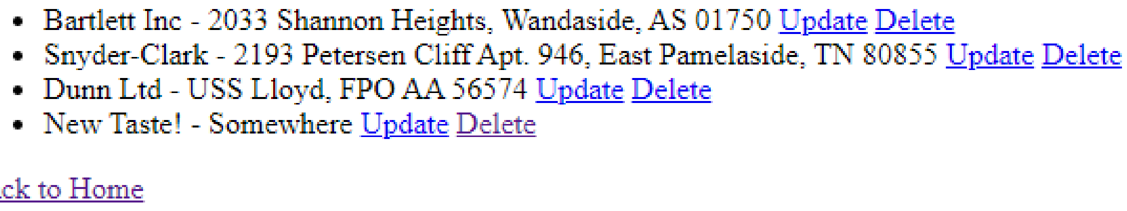
Location: Somewhere

Cuisine Type: Seafood

Price Range: $$$$

Contact Info: 458-9765-8978

After entering the information, click ”Add”, and the restaurant will appear at the end of the list of restaurants:



- Bartlett Inc - 2033 Shannon Heights, Wandaside, AS 01750 Update Delete
- Snyder-Clark - 2193 Petersen Cliff Apt. 946, East Pamelaside, TN 80855 Update Delete
- Dunn Ltd - USS Lloyd, FPO AA 56574 Update Delete
- New Taste! - Somewhere Update Delete

Back to Home