DBMS PROJECT REPORT

'JUST EAT IT' - An online food ordering and delivery system

Introduction and Problem Definition

We aimed to create a database application that would serve the purpose of an online food ordering and delivery system. Three groups of users can use the website: the customers, restaurants and agents. The user can browse through the different menus of the restaurants registered and order items but,only from a single restaurant at once.The restaurants and the agents are automatically updated in the database according to the order placed by the user

Miniworld

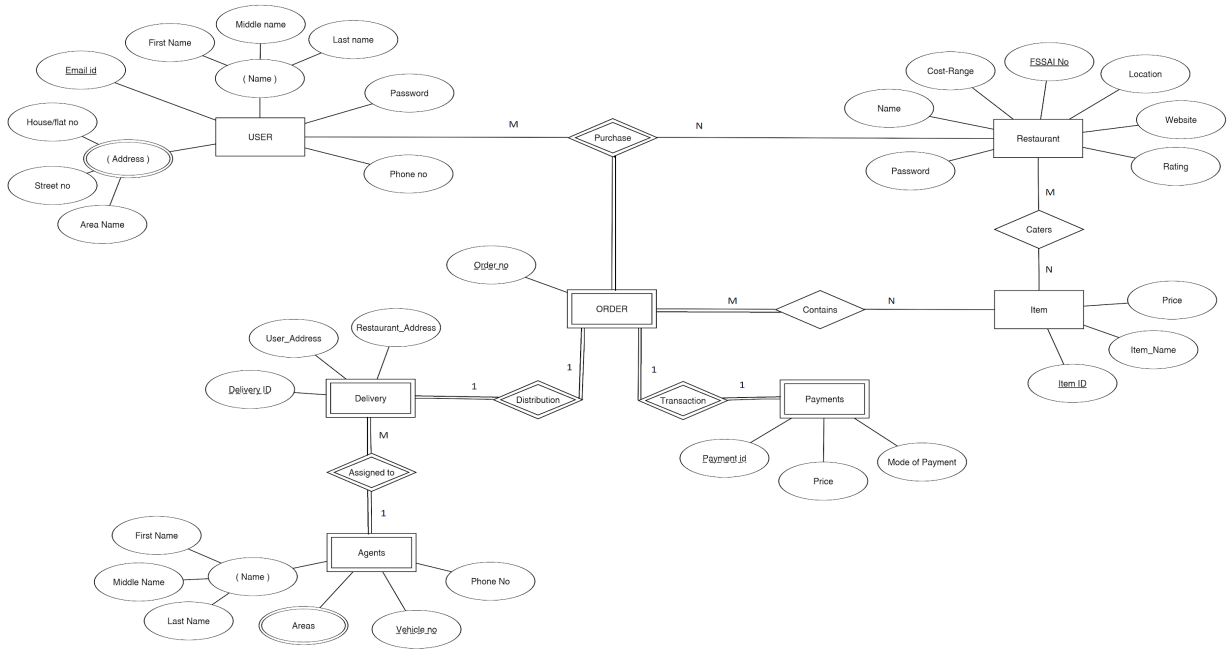
The miniworld consists of three main entities : the user, the restaurant and the agent. The user has the attributes first name,middle name,last name,email phone number and password with the primary key being the user's email. The restaurant has the attributes name,cost range,FSSAI number,location,rating password and website. The primary key is the restaurant's FSSAI number. The agent has the attributes vehicle number,phone number,first name,middle name and last name, with the primary key being the agent's vehicle number. The other entities include item,agent_areas,order,user_address,ordered_items,delivery and payment. The relationships in the miniworld are caters(restaurant and item) and prefers(user and item).

Requirements

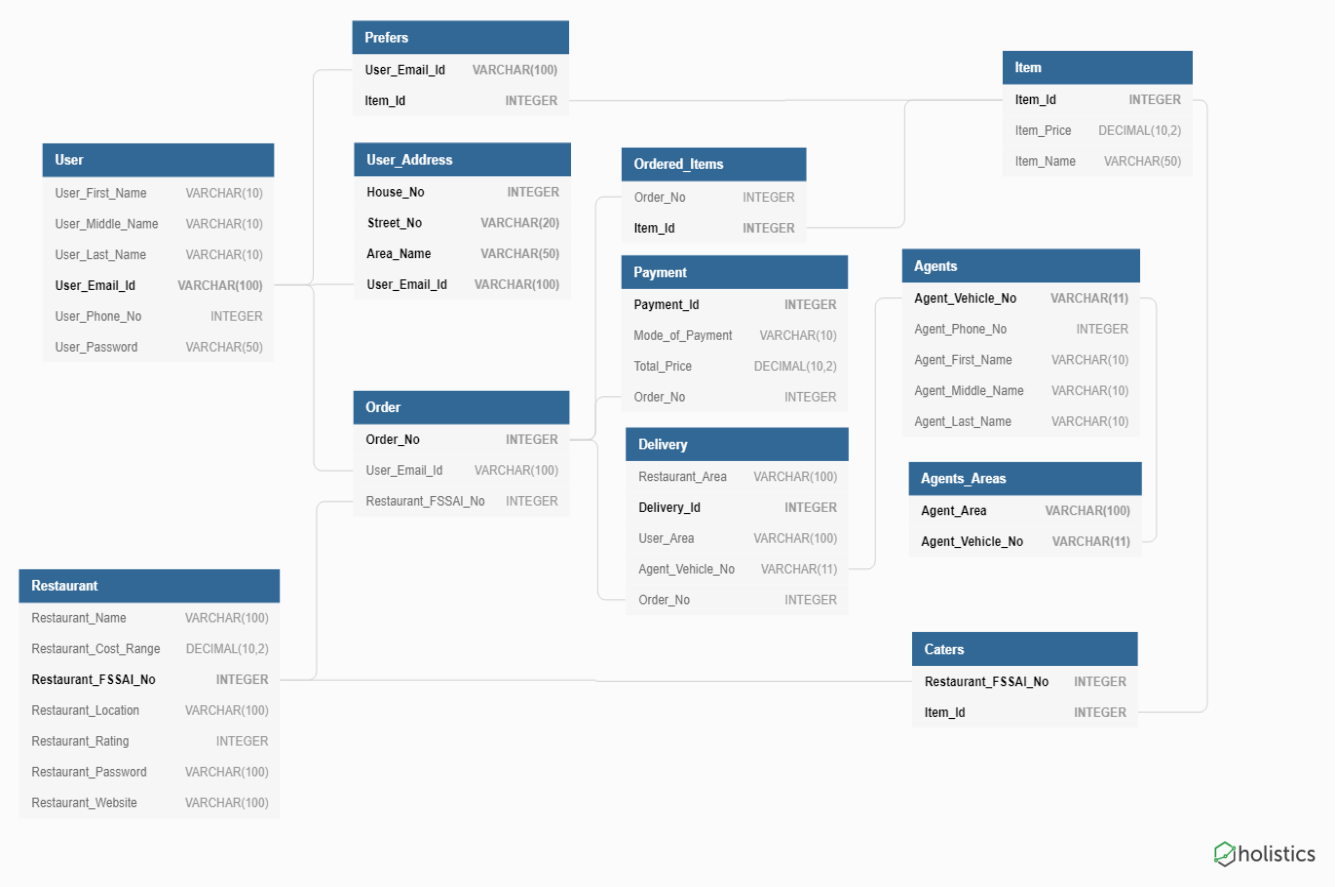
A user, on visiting the page is able to browse through the registered restaurants and their menus.To order food from the restaurant,the user must login(if previously registered) or sign up - this requirement results in a form to insert a user record. The user can then order food from the restaurant of his choice ,but he can only order items from one restaurant at once.

According to the future scope of this project,the restaurants can login to view their orders and the agents assigned to each order.If the restaurant is not registered,it has to be first registered -this requirement results in a form to insert restaurant details.The agents can also do so.

ER Diagram:



Schema diagram:



Create Statements:

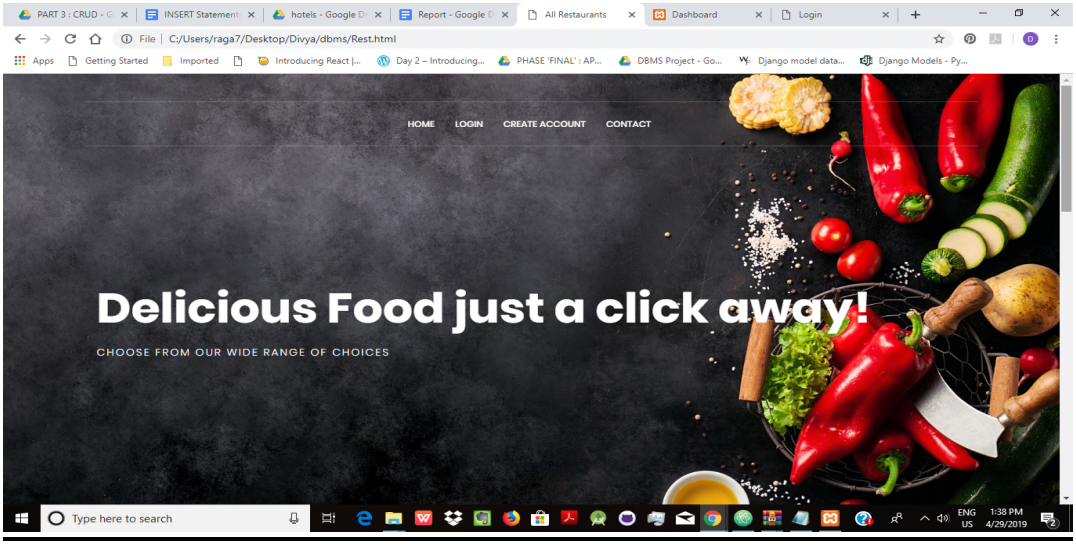
```
CREATE TABLE User
(
  User_First_Name VARCHAR(10) NOT NULL,
  User_Middle_Name VARCHAR(10),
  User_Last_Name VARCHAR(10) NOT NULL,
  User_Email_Id VARCHAR(100) NOT NULL,
  User_Phone_No INTEGER NOT NULL,
  User_Password VARCHAR(50) NOT NULL,
  PRIMARY KEY (User_Email_Id),
  UNIQUE (User_Phone_No)
);
```

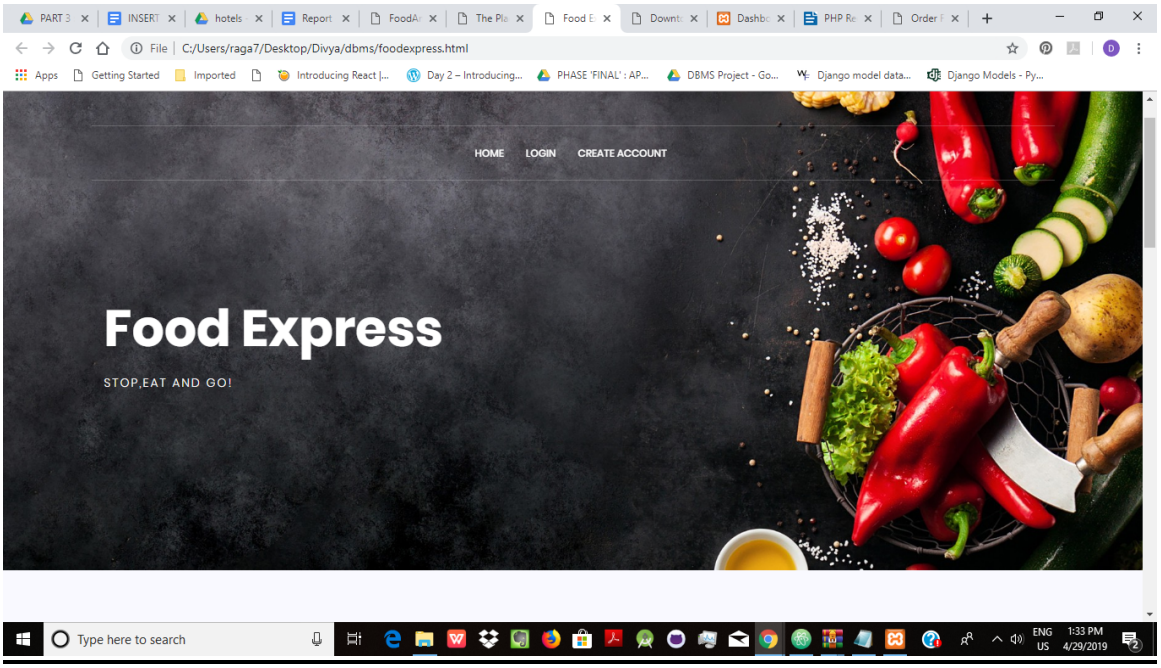
```
CREATE TABLE Restaurant
(
 Restaurant_Name VARCHAR(100) NOT NULL,
 Restaurant_Cost_Range DECIMAL(10,2) NOT NULL,
 Restaurant_FSSAI_No INTEGER NOT NULL,
 Restaurant_Location VARCHAR(100) NOT NULL,
 Restaurant_Rating INTEGER NOT NULL,
 Restaurant_Password VARCHAR(100) NOT NULL,
 Restaurant_Website VARCHAR(100) NOT NULL,
 PRIMARY KEY (Restaurant_FSSAI_No),
 UNIQUE (Restaurant_Website)
 CHECK (Restaurant_Cost_Range > 0),
 CHECK (Restaurant_Rating >= 0 AND Restaurant_Rating <= 5)
);

CREATE TABLE Item
(
 Item_Id INTEGER NOT NULL,
 Item_Price DECIMAL(10,2) NOT NULL,
 Item_Name VARCHAR(50) NOT NULL,
 PRIMARY KEY (Item_Id),
 CHECK (Item_Price > 0)
);

CREATE TABLE Agents
(
 Agent_Vehicle_No VARCHAR(11) NOT NULL,
 Agent_Phone_No INTEGER NOT NULL,
 Agent_First_Name VARCHAR(10) NOT NULL,
 Agent_Middle_Name VARCHAR(10),
 Agent_Last_Name VARCHAR(10) NOT NULL,
 PRIMARY KEY (Agent_Vehicle_No),
 UNIQUE (Agent_Phone_No)
);

CREATE TABLE Agents_Areas
(
 Agent_Area VARCHAR(100) NOT NULL,
 Agent_Vehicle_No VARCHAR(11) NOT NULL,
 PRIMARY KEY (Agent_Area, Agent_Vehicle_No),
 FOREIGN KEY (Agent_Vehicle_No) REFERENCES Agents(Agent_Vehicle_No)
);

CREATE TABLE Caters
```

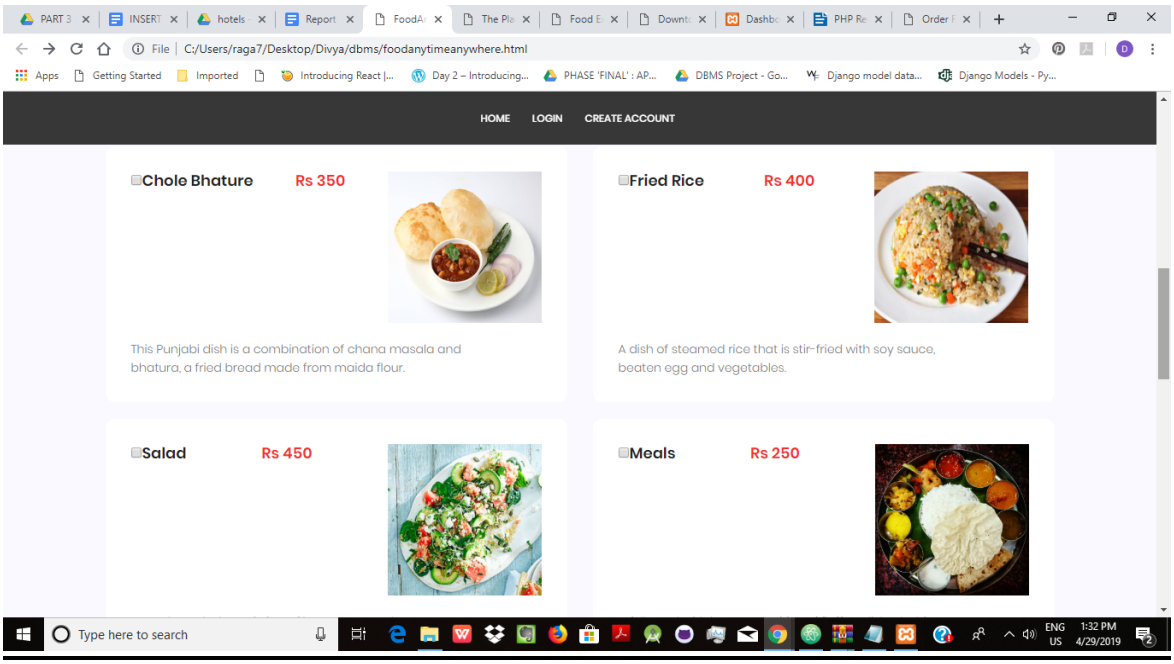
```
(
  Restaurant_FSSAI_No INTEGER NOT NULL,
  Item_Id INTEGER NOT NULL,
  PRIMARY KEY (Restaurant_FSSAI_No, Item_Id),
  FOREIGN KEY (Restaurant_FSSAI_No) REFERENCES Restaurant(Restaurant_FSSAI_No),
  FOREIGN KEY (Item_Id) REFERENCES Item(Item_Id)
);

CREATE TABLE Prefers
(
  User_Email_Id VARCHAR(100) NOT NULL,
  Item_Id INTEGER NOT NULL,
  PRIMARY KEY (User_Email_Id, Item_Id),
  FOREIGN KEY (User_Email_Id) REFERENCES User(User_Email_Id),
  FOREIGN KEY (Item_Id) REFERENCES Item(Item_Id)
);

CREATE TABLE Order
(
  Order_No INTEGER NOT NULL,
  User_Email_Id VARCHAR(100) NOT NULL,
  Restaurant_FSSAI_No INTEGER NOT NULL,
  PRIMARY KEY (Order_No),
  FOREIGN KEY (User_Email_Id) REFERENCES User(User_Email_Id),
  FOREIGN KEY (Restaurant_FSSAI_No) REFERENCES Restaurant(Restaurant_FSSAI_No)
);

CREATE TABLE User_Address
(
  House_No INTEGER NOT NULL,
  Street_No VARCHAR(20) NOT NULL,
  Area_Name VARCHAR(50) NOT NULL,
  User_Email_Id VARCHAR(100) NOT NULL,
  PRIMARY KEY (House_No, Street_No, Area_Name, User_Email_Id),
  FOREIGN KEY (User_Email_Id) REFERENCES User(User_Email_Id)
);

CREATE TABLE Ordered_Items
(
  Order_No INTEGER NOT NULL,
  Item_Id INTEGER NOT NULL,
  PRIMARY KEY (Order_No, Item_Id),
  FOREIGN KEY (Order_No) REFERENCES Order(Order_No),
  FOREIGN KEY (Item_Id) REFERENCES Item(Item_Id)
```

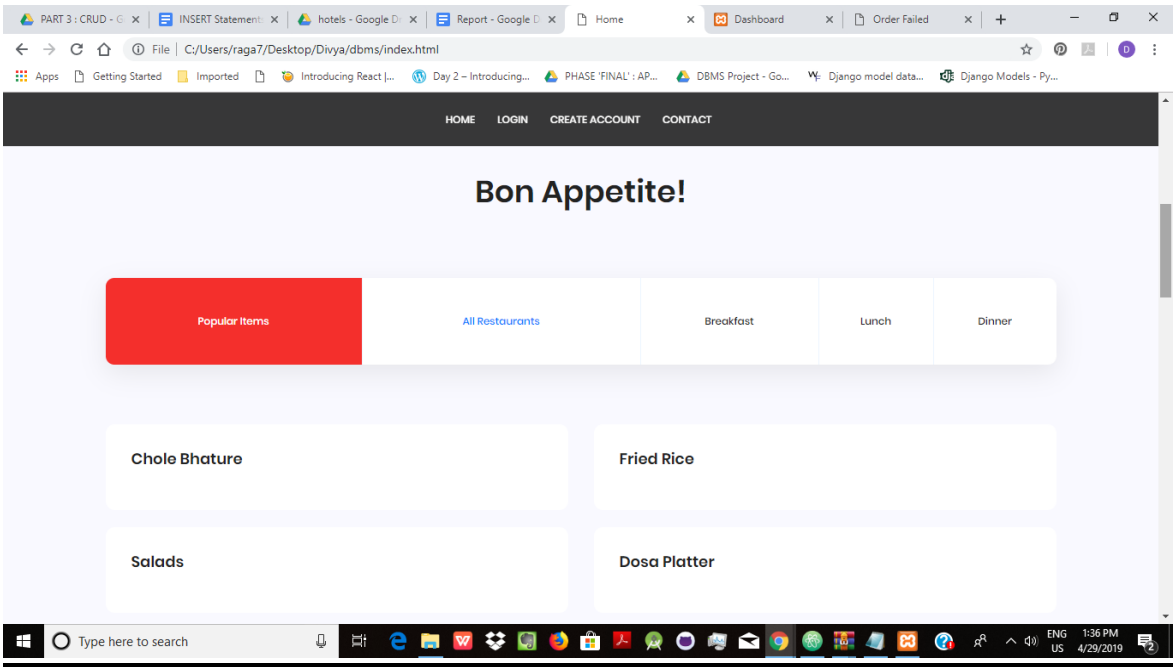
```
);

CREATE TABLE Delivery
(
 Restaurant_Area VARCHAR(100) NOT NULL,
 Delivery_Id INTEGER NOT NULL,
 User_Area VARCHAR(100) NOT NULL,
 Agent_Vehicle_No VARCHAR(11) NOT NULL,
 Order_No INTEGER NOT NULL,
 PRIMARY KEY (Delivery_Id),
 FOREIGN KEY (Agent_Vehicle_No) REFERENCES Agents(Agent_Vehicle_No),
 FOREIGN KEY (Order_No) REFERENCES Order(Order_No)
);

CREATE TABLE Payment
(
 Payment_Id INTEGER NOT NULL,
 Mode_of_Payment VARCHAR(10) NOT NULL,
 Total_Price DECIMAL(10,2) NOT NULL,
 Order_No INTEGER NOT NULL,
 PRIMARY KEY (Payment_Id),
 FOREIGN KEY (Order_No) REFERENCES Order(Order_No),
 CHECK (Total_Price > 0)
);
```

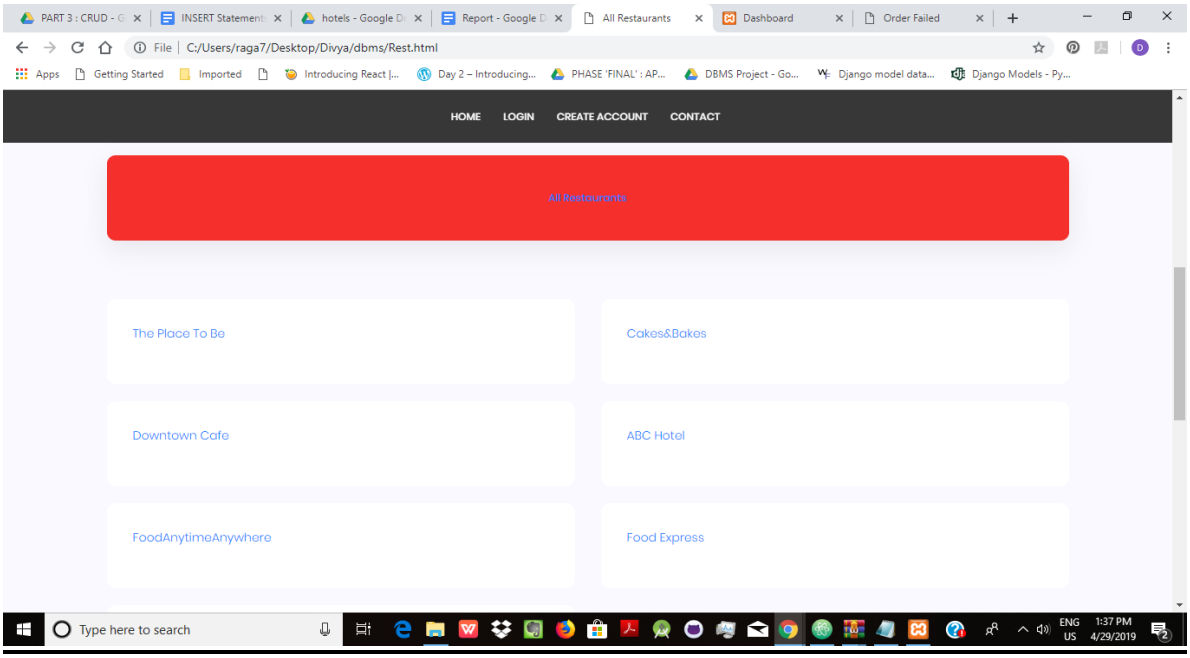
PHP Forms,UI Screens:

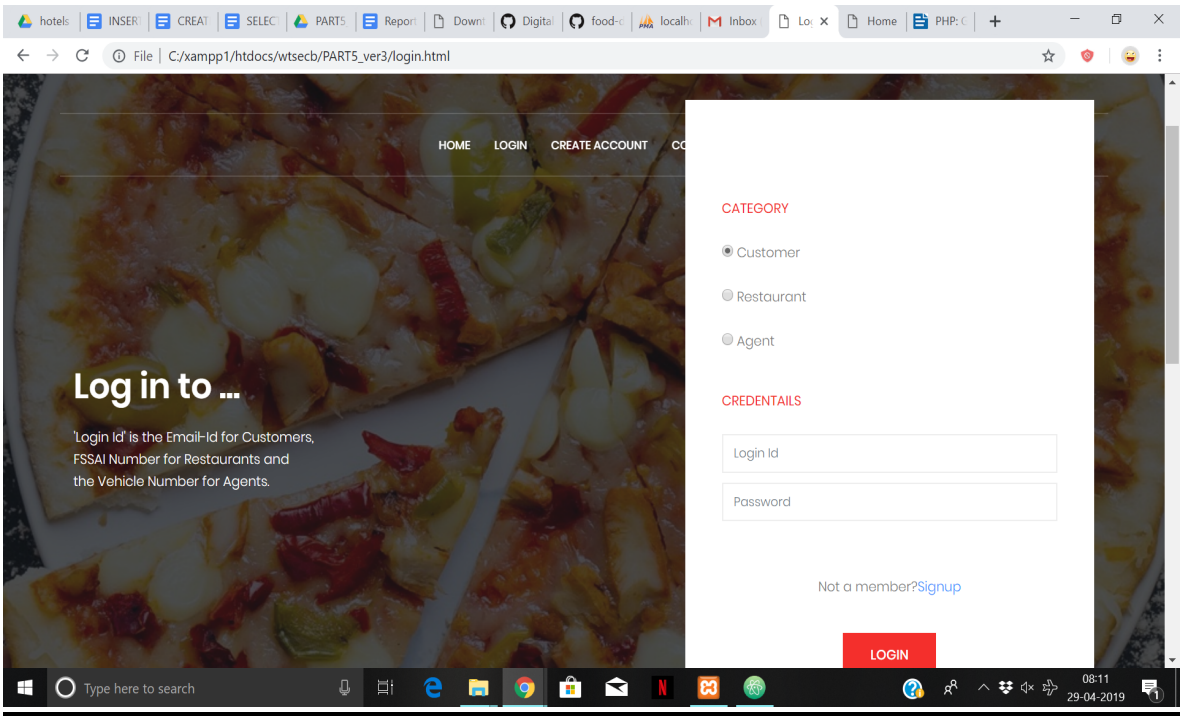


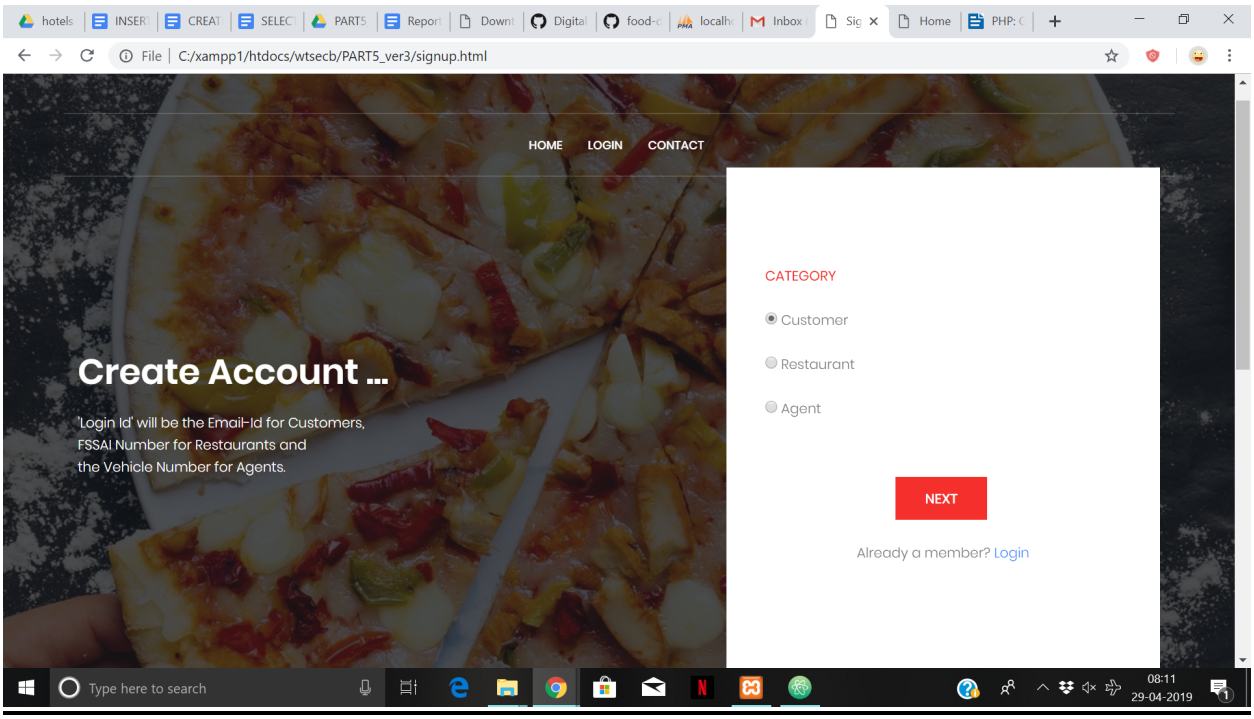

File C:/Users/raga7/Desktop/Divya/dbms/foodexpress.html
HOME
LOGIN
CREATE ACCOUNT
Food Express
STOP,EAT AND GO!
Type here to search
ENG US
1:33 PM
4/29/2019

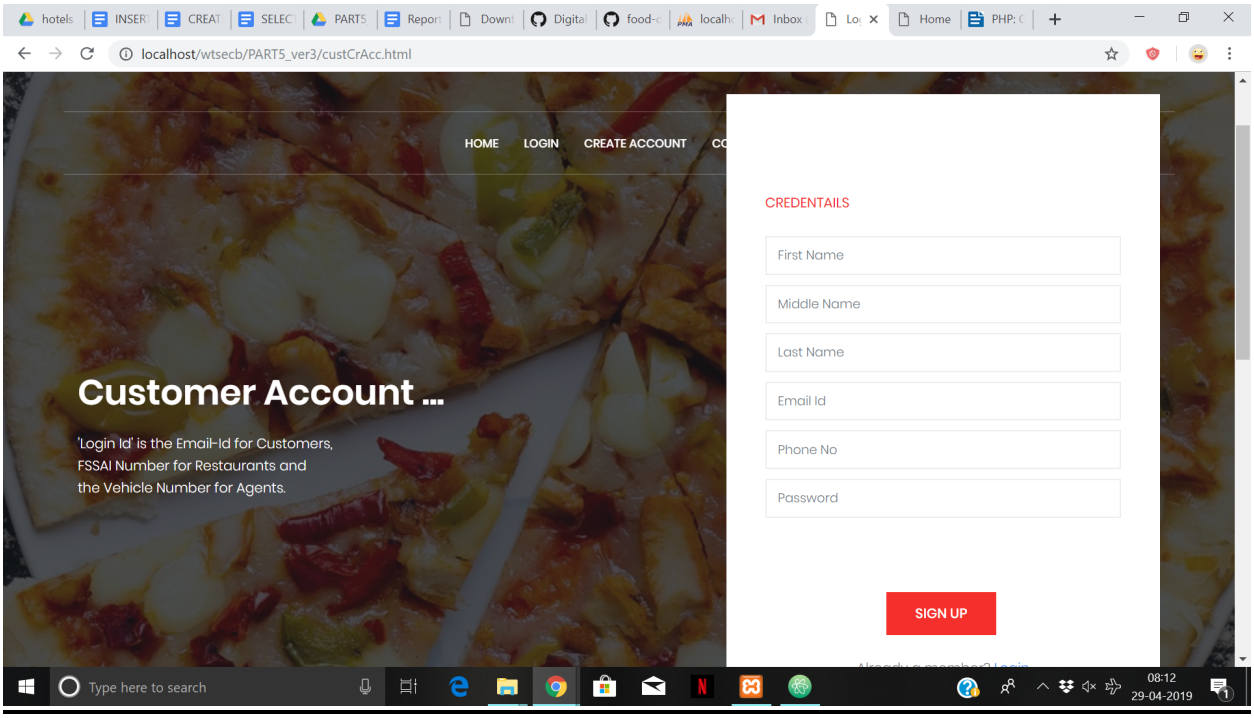

File C:/Users/raga7/Desktop/Divya/dbms/foodanytimeanywhere.html
HOME
LOGIN
CREATE ACCOUNT
Chole Bhature
Rs 350
This Punjabi dish is a combination of chana masala and bhatura, a fried bread made from maida flour.
Fried Rice
Rs 400
A dish of steamed rice that is stir-fried with soy sauce, beaten egg and vegetables.
Salad
Rs 450
Meals
Rs 250
Type here to search
ENG US
1:32 PM
4/29/2019

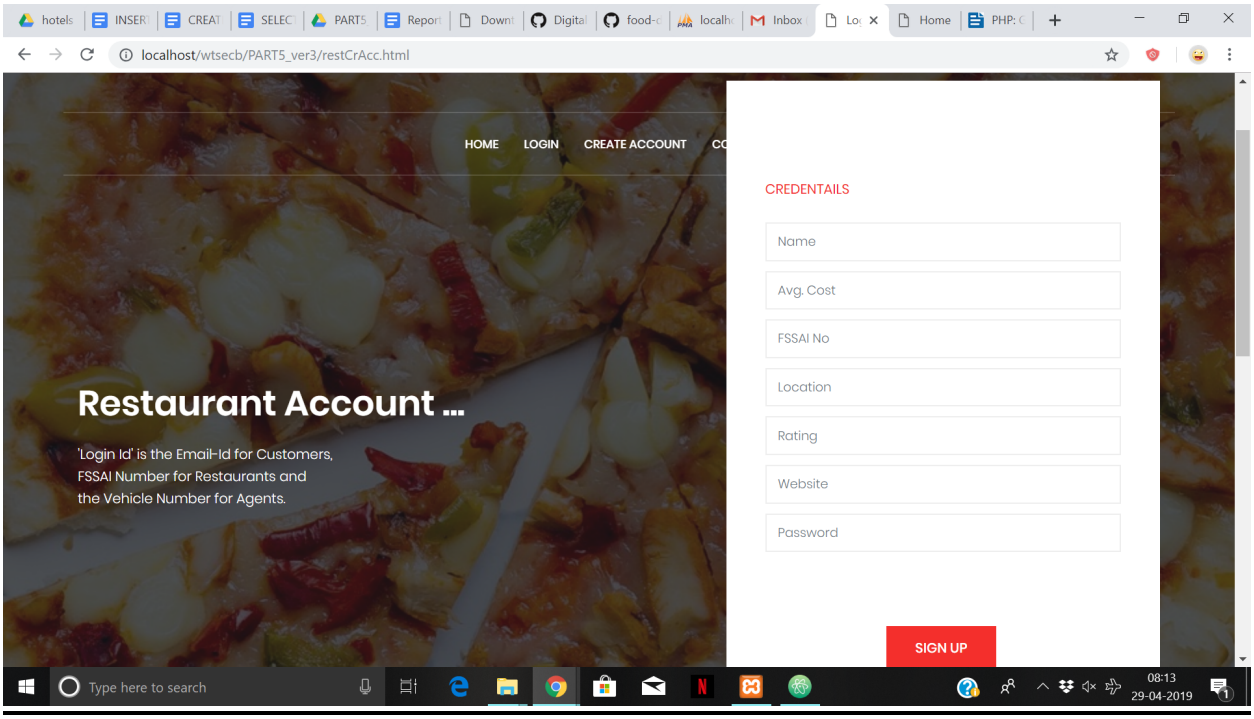

HOME
LOGIN
CREATE ACCOUNT
CONTACT
Bon Appetite!
Popular Items
All Restaurants
Breakfast
Lunch
Dinner
Chole Bhature
Fried Rice
Salads
Dosa Platter

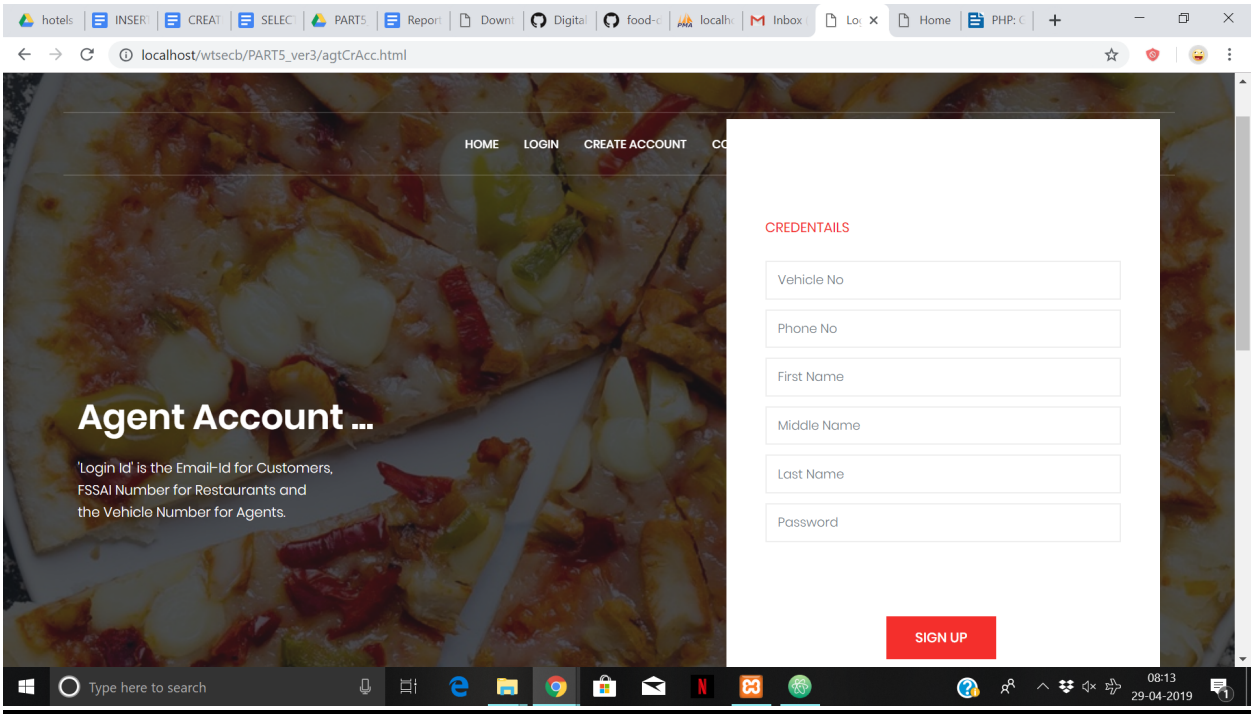

HOME
LOGIN
CREATE ACCOUNT
CONTACT
All Restaurants
The Place To Be
Cakes&Bakes
Downtown Cafe
ABC Hotel
FoodAnytimeAnywhere
Food Express

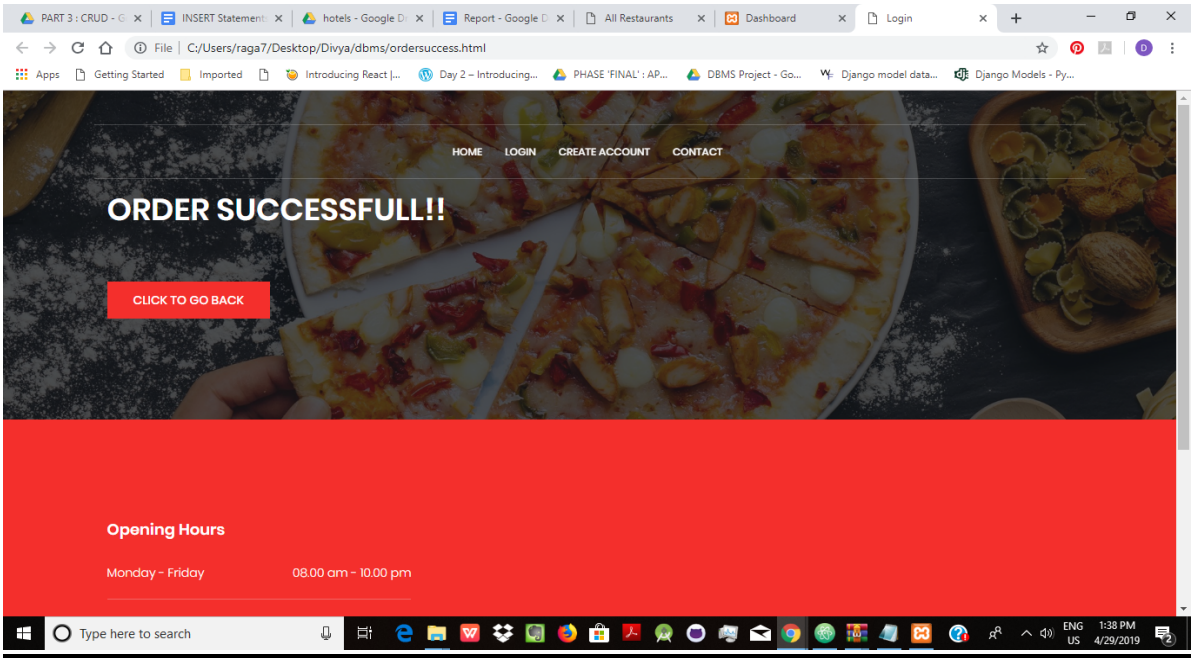

File | C:/xampp1/htdocs/wtsecb/PART5_ver3/login.html
HOME
LOGIN
CREATE ACCOUNT
Log in to ...
'Login Id' is the Email-Id for Customers,
FSSAI Number for Restaurants and
the Vehicle Number for Agents.
CATEGORY
Customer
Restaurant
Agent
CREDENTAILS
Login Id
Password
Not a member?Signup
LOGIN
Type here to search
08:11
29-04-2019

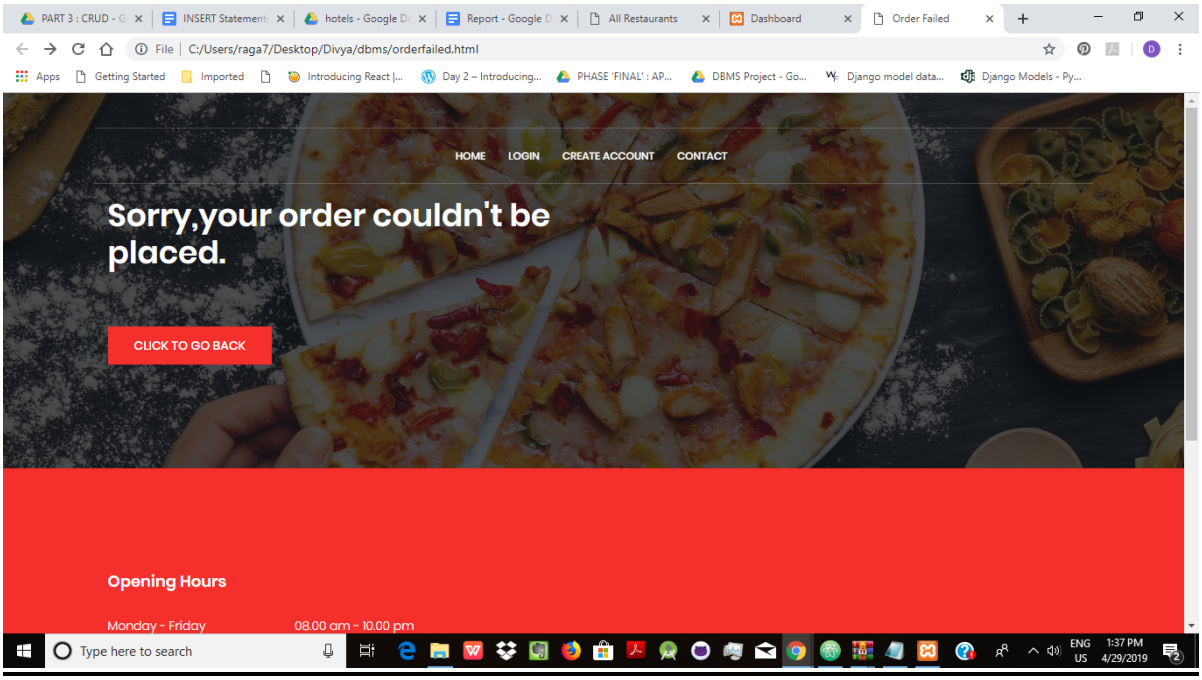

File | C:/xampp1/htdocs/wtsecb/PART5_ver3/signup.html
HOME
LOGIN
CONTACT
Create Account ...
'Login Id' will be the Email-Id for Customers,
FSSAI Number for Restaurants and
the Vehicle Number for Agents.
CATEGORY
Customer
Restaurant
Agent
NEXT
Already a member? Login
Type here to search
08:11
29-04-2019


localhost/wtsecb/PART5_ver3/custCrAcc.html
HOME
LOGIN
CREATE ACCOUNT
Customer Account ...
'Login Id' is the Email-Id for Customers,
FSSAI Number for Restaurants and
the Vehicle Number for Agents.
CREDENTAILS
First Name
Middle Name
Last Name
Email Id
Phone No
Password
SIGN UP
Type here to search
08:12
29-04-2019


localhost/wtsecb/PART5_ver3/restCrAcc.html
HOME
LOGIN
CREATE ACCOUNT
Restaurant Account ...
'Login Id' is the Email-Id for Customers,
FSSAI Number for Restaurants and
the Vehicle Number for Agents.
CREDENTAILS
Name
Avg. Cost
FSSAI No
Location
Rating
Website
Password
SIGN UP
Type here to search
08:13
29-04-2019


localhost/wtsecb/PART5_ver3/agtCrAcc.html
HOME
LOGIN
CREATE ACCOUNT
CREDENTAILS
Vehicle No
Phone No
First Name
Middle Name
Last Name
Password
SIGN UP
Agent Account ...
'Login Id' is the Email-Id for Customers,
FSSAI Number for Restaurants and
the Vehicle Number for Agents.
Type here to search
08:13
29-04-2019


File | C:/Users/raga7/Desktop/Divya/dbms/ordersuccess.html
HOME
LOGIN
CREATE ACCOUNT
CONTACT
ORDER SUCCESSFULL!!
CLICK TO GO BACK
Opening Hours
Monday - Friday
08.00 am - 10.00 pm
Type here to search
ENG
US
1:38 PM
4/29/2019



SQL Queries: (SELECT Statements)

```
select * from Customer;
select * from Restaurant;
select * from Item;
select * from Agents;
select * from Agents_Areas;
select * from Caters;
select * from Prefers;
select * from foodorder;
select * from Customer_Address;
select * from Ordered_Items;
select * from Delivery;
select * from Payment;
```

1 : List all the items delivered by 'Surya'

```
SELECT Item_Name FROM Item WHERE Item_Id in
(
        SELECT Item_Id FROM Ordered_Items WHERE Order_No in
        (
                SELECT Order_No FROM Delivery WHERE Agent_Vehicle_No =
                (
                        SELECT Agent_Vehicle_No FROM Agents WHERE Agent_First_Name= 'Surya'
                )
        )
);
```

2 : Find the number of items delivered by each agent

```
SELECT Agent_First_Name, COUNT(Item_Id)
FROM Agents, Delivery, Ordered_Items
WHERE Delivery.Agent_Vehicle_No = Agents.Agent_Vehicle_No and Delivery.Order_No = Ordered_Items.Order_No
GROUP BY Agent_First_Name;
```

3 : Find the number of items bought by each customer

```
SELECT Customer_First_Name, COUNT(Item_Id)
FROM Customer, foodorder, Ordered_Items
WHERE foodorder.Customer_Email_Id = Customer.Customer_Email_Id and foodorder.Order_No = Ordered_Items.Order_No
GROUP BY Customer_First_Name;
```

4 : The total amount the customer has paid

```
SELECT Customer_First_Name, SUM(Item_Price)
FROM Item, Customer, foodorder, Ordered_Items
WHERE foodorder.Customer_Email_Id = Customer.Customer_Email_Id and foodorder.Order_No = Ordered_Items.Order_No and Item.Item_Id=Ordered_Items.Item_Id GROUP BY Customer_First_Name;
```

5 : Who are the customers that have purchased all the items purchased by ‘Asha’

```
SELECT Customer_First_Name FROM customer WHERE Customer_Email_Id IN
(
```

```
SELECT Customer_Email_Id FROM foodorder WHERE order_no in
(
        SELECT order_no FROM Ordered_Items Ord_Itm WHERE NOT EXISTS
        (
                (SELECT Item_Id FROM Customer, foodorder, Ordered_Items WHERE
                foodorder.Customer_Email_Id = Customer.Customer_Email_Id and
                foodorder.Order_No = Ord_Itm.Order_No and customer. Customer_First_Name
                = 'Asha')

                EXCEPT

                (SELECT Item_Id FROM Customer, foodorder WHERE
                foodorder.Customer_Email_Id = Customer.Customer_Email_Id and
                foodorder.Order_No = Ord_Itm.Order_No)
        )
)
);
```

6 : Name the agent who has had most number of delivery orders… is it the same as the agent who delivered items of the highest cost(total)... is it the same as the agent who had the highest number of items to be delivered

```
SELECT Agent_First_Name, no_del no_of_deliveries, tot_cost total_cost_of_delivery, no_items number_of_items_delivered
FROM Agents,
(
        SELECT AD, no_del, tot_cost, no_items
        FROM
                (SELECT Agent_vehicle_no AD, COUNT(order_no) no_del
                FROM delivery
                GROUP BY Agent_vehicle_no) AS D,

                (SELECT Agent_vehicle_no AC, SUM(item_price) tot_cost
                FROM ordered_items,delivery,item
                WHERE ordered_items.order_no=delivery.order_no and
                item.item_id=ordered_items.item_id
                GROUP BY Agent_vehicle_no) AS C,
```

```
                (SELECT Agent_vehicle_no AI, COUNT(item_id) no_items
                FROM ordered_items,delivery
                WHERE ordered_items.order_no=delivery.order_no GROUP by Agent_vehicle_no) AS I

        WHERE
                D.AD=I.AI and D.AD=C.AC and I.AI=C.AC
)
AS req_info
WHERE Agents.Agent_vehicle_no = req_info.AD;
```

7 : Which is the most popular dish among the customers

```
SELECT Item_Name Most_Popular_Items
FROM Item
WHERE item_id IN
(
        SELECT Item_id
        FROM
        (
                SELECT Item_id, COUNT(*) cnt
                FROM ordered_items
                GROUP BY item_id
        )
        AS itm_cnt
        WHERE cnt IN
        (
                SELECT MAX(cnt)
                FROM
                (
                        SELECT Item_id, COUNT(*) cnt
                        FROM ordered_items
                        GROUP BY item_id
                )
                AS itm_cnt
        )
);
```

8 : Retrieve the dish most frequently bought by each customer

No trend observed….

```
SELECT Customer_Email_Id, Item_id, COUNT(Item_id)
FROM foodorder,Ordered_Items
WHERE Ordered_Items.order_no=foodorder.order_no
GROUP BY Customer_Email_Id, Item_id;
```

9 : Which restaurant is the most popular according to number of purchases…. What is its rating

```
SELECT Restaurant_Name, Restaurant_Rating
FROM Restaurant
WHERE Restaurant_FSSAI_No IN
(
	SELECT Restaurant_FSSAI_No
	FROM
	(
		SELECT Restaurant_FSSAI_No, COUNT(*) cnt
		FROM foodorder
		GROUP BY Restaurant_FSSAI_No
	)
	AS r_cnt
	WHERE r_cnt.cnt IN
	(
		SELECT MAX(cnt)
		FROM
		(
			SELECT Restaurant_FSSAI_No, COUNT(*) cnt
			FROM foodorder
			GROUP BY Restaurant_FSSAI_No
		)
		AS rc
	)
);
```

10 : Which area has the popular restaurants

```
SELECT Restaurant_Location Areas_most_pop_rest
FROM
(
	SELECT Restaurant_Location, COUNT(Restaurant_Location) cnt
```

```
        FROM Restaurant
        WHERE Restaurant_FSSAI_No IN
        (
                SELECT Restaurant_FSSAI_No
                FROM
                (
                        SELECT Restaurant_FSSAI_No, COUNT(*) cnt
                        FROM foodorder
                        GROUP BY Restaurant_FSSAI_No
                )
                AS r_cnt
                WHERE r_cnt.cnt IN
                (
                        SELECT MAX(cnt)
                        FROM
                        (
                                SELECT Restaurant_FSSAI_No, COUNT(*) cnt
                                FROM foodorder
                                GROUP BY Restaurant_FSSAI_No
                        )
                        AS rc
                )
        )
        GROUP BY Restaurant_Location
) AS r_loc
WHERE r_loc.cnt IN
(
        SELECT MAX(cnt)
        FROM
        (
                SELECT Restaurant_Location, COUNT(Restaurant_Location) cnt
                FROM Restaurant
                WHERE Restaurant_FSSAI_No IN
                (
                        SELECT Restaurant_FSSAI_No
                        FROM
                        (
                                SELECT Restaurant_FSSAI_No, COUNT(*) cnt
                                FROM foodorder
                                GROUP BY Restaurant_FSSAI_No
                        )
                        AS r_cnt
```

```
                        WHERE r_cnt.cnt IN
                        (
                                SELECT MAX(cnt)
                                FROM
                                (
                                        SELECT Restaurant_FSSAI_No, COUNT(*) cnt
                                        FROM foodorder
                                        GROUP BY Restaurant_FSSAI_No
                                )
                                AS rc
                        )
                )
                GROUP BY Restaurant_Location
        )
        AS r_loc
);
```

11 : Which area is the source of most number of customers

```
SELECT Area_name Areas_most_cust
FROM
(
        SELECT Area_name, COUNT(Area_name) cnt
        FROM
                Customer_Address
                RIGHT JOIN
                (Select Customer_Email_Id, count(Customer_Email_Id) from foodorder group by
                Customer_Email_Id) AS cust_cnt
                ON Customer_Address.Customer_Email_Id = cust_cnt.Customer_Email_Id

        group by area_name) as c_loc
Where c_loc.cnt in
(Select max(cnt) from (Select Area_name, count(Area_name) cnt from Customer_Address RIGHT JOIN
(Select Customer_Email_Id, count(Customer_Email_Id) from foodorder group by Customer_Email_Id) AS
cust_cnt ON Customer_Address.Customer_Email_Id = cust_cnt.Customer_Email_Id group by area_name)
as c_loc);
```

14 : Draw a mapping between the number of restaurants, customers and agents locality-wise

Select area, cnt_r no_restaurants, cnt_c no_customers, cnt_a no_agents from (Select agent_area area, cnt_a, cnt_r from (select agent_area, count(agent_area) cnt_a from Agents_areas group by agent_area) AS ag LEFT JOIN (select Restaurant_Location, count(Restaurant_Location) cnt_r from Restaurant group by Restaurant_Location) AS rs ON ag.agent_area=rs.Restaurant_Location) AS ar LEFT JOIN (select Area_Name, count(Area_Name) cnt_c from Customer_Address group by Area_Name) AS cs ON ar.area=cs.Area_Name;

Test cases for testing exception/error conditions:

The user can only order from a single restaurant at a time.When a user goes to a particular restaurant's page and orders, it is stored as one order. When the same user goes to some other restaurant's page and tries to order, that order is saved as a new order which is completely separate from the previous order placed by that user.

Discussion & Conclusion

The project intends to help us understand the process behind developing a relational database system.

Starting off, we learnt the basic steps of creating the database, and modifying it. We further on went about, learning how to query out the information in a useful manner, using complex query systems.

Finally, we developed a user interface, using which the users/customers can access the required information from the database while placing the food order in this case.