

Project Documentation

Healtholic - Healthy food ordering app

Group No: 18

AU1841038 Mithilesh Thakkar

AU1841094 Yagnik Hingrajiya

AU1841124 Prince Dalsaniya

1.Requirement Specification Document

We want to build the Swiggy of health market, where we want to cover the health aspect of the food by providing him/her with the quantitative information of the fiber, vitamins, calories, fat, sugar, protein, carbohydrates.

❖ There will be many facilities like - food cart, order list, favourite food list, payment gateway, customized food.

1. Food cart: food cart will be useful for temporary save food which you want to order food, where you will be able to add, update or delete the food item in it, once you order the food cart will be discarded.
2. Order list: Here you will be able to track your precious order and you will be able to see your great previous orders.
3. Favourite Food list: this is the place where you can save your favourite food and favourite restaurant where you want to order most, in this list you can add, update and delete the food item or restaurant.
4. Customized food: - this is the place to show your creativity, here you can create your customized food on bases of the portion of the nutrition.
5. Payment gateway: here is the place for the payment part, three types of payment is available: cash, card, online.

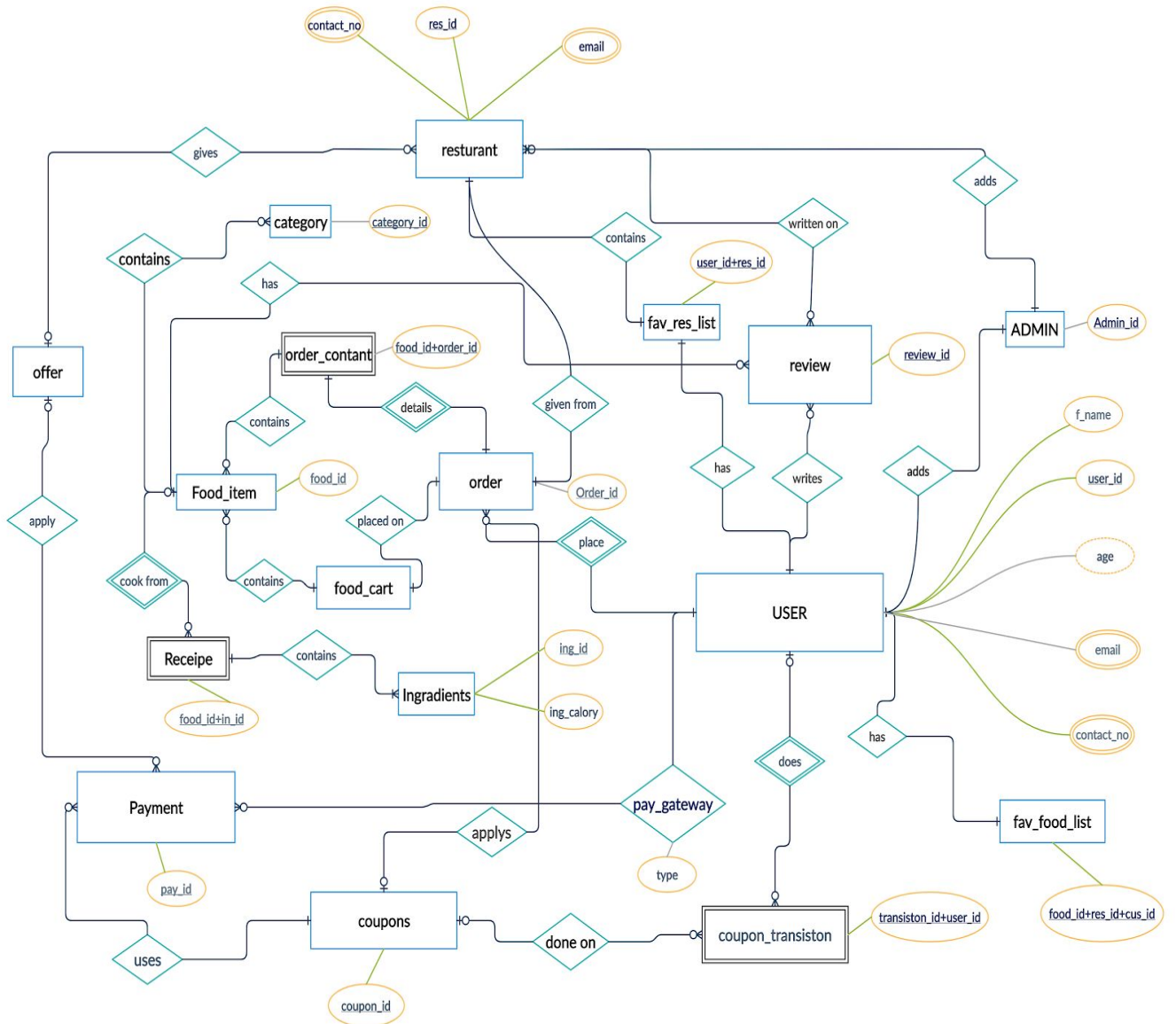
➤ There will be analysis be like:

1. How many times did you order your favourite food?
2. How many times did you order from your favourite restaurant?
3. Nutrition gain from every order and how much needed
4. Nutrition gain per rupee in a particular order
5. suggestion of the food based on your food ordering history.
6. Age-wise popular restaurant.
7. Rating-wise Popular restaurant.

➤ There will be a trigger like this:


1. Validation on adding into your favourite food list.
2. Validation on adding into the favourite restaurant list.
3. Validation on the cart.
4. Insert the data into the cart.
5. Deleting the cart when going to check out.

2. ER diagram






3. Table Design (Data Dictionary):


1> Admin

	#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	username 	varchar(20)	latin1_swedish_ci		No	<i>None</i>
<input type="checkbox"/>	2	password	varchar(8)	latin1_swedish_ci		Yes	<i>NULL</i>


2> Cart

#	Name	Type	Collation	Attributes	Null	Default
1	customer_id 	varchar(5)	latin1_swedish_ci		Yes	<i>NULL</i>
2	r_id 	varchar(5)	latin1_swedish_ci		Yes	<i>NULL</i>
3	food_id 	varchar(5)	latin1_swedish_ci		Yes	<i>NULL</i>
4	quantity	int(11)			Yes	<i>NULL</i>

3> Category

#	Name	Type	Collation	Attributes	Null	Default
1	cat_id 	varchar(5)	latin1_swedish_ci		No	<i>None</i>
2	name	varchar(20)	latin1_swedish_ci		Yes	<i>NULL</i>

4> coupon

#	Name	Type	Collation	Attributes	Null	Default
1	coupon_id 	varchar(5)	latin1_swedish_ci		No	<i>None</i>
2	description	varchar(50)	latin1_swedish_ci		Yes	<i>NULL</i>

5> coupon_trasation

#	Name	Type	Collation	Attributes	Null	Default
1	coupon_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	customer_id 🔑	varchar(5)	latin1_swedish_ci		No	None
3	t_date	datetime			Yes	NULL
4	discount	int(11)			No	None

6> customer

#	Name	Type	Collation	Attributes	Null	Default
1	customer_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	fname	varchar(20)	latin1_swedish_ci		Yes	NULL
3	mname	varchar(20)	latin1_swedish_ci		Yes	NULL
4	lname	varchar(20)	latin1_swedish_ci		Yes	NULL
5	city	varchar(20)	latin1_swedish_ci		Yes	NULL
6	state	varchar(20)	latin1_swedish_ci		Yes	NULL
7	country	varchar(20)	latin1_swedish_ci		Yes	NULL
8	bdate	date			Yes	NULL
9	age	int(11)			Yes	NULL
10	contact_no	varchar(10)	latin1_swedish_ci		Yes	NULL
11	email	varchar(50)	latin1_swedish_ci		Yes	NULL
12	coupon_count	int(11)			Yes	NULL
13	username	varchar(20)	latin1_swedish_ci		Yes	NULL
14	password	varchar(8)	latin1_swedish_ci		Yes	NULL

7> favourite_food

#	Name	Type	Collation	Attributes	Null	Default
1	food_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	customer_id 🔑 🔒	varchar(5)	latin1_swedish_ci		No	None
3	r_id 🔑 🔒	varchar(5)	latin1_swedish_ci		No	None

8> favourite_restaurant

#	Name	Type	Collation	Attributes	Null	Default
1	customer_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	r_id 🔑 🔑	varchar(5)	latin1_swedish_ci		No	None

9> Food_item

#	Name	Type	Collation	Attributes	Null	Default
1	food_id 🔑	varchar(10)	latin1_swedish_ci		No	None
2	food_name	varchar(50)	latin1_swedish_ci		Yes	NULL
3	category_id	varchar(50)	latin1_swedish_ci		Yes	NULL

10> ingredients

#	Name	Type	Collation	Attributes	Null	Default
1	i_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	name	varchar(20)	latin1_swedish_ci		Yes	NULL
3	carbs	float			Yes	NULL
4	fat	float			Yes	NULL
5	fiber	float			Yes	NULL
6	protine	float			Yes	NULL
7	calory	float			Yes	NULL

11> menu

#	Name	Type	Collation	Attributes	Null	Default
1	food_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	r_id 🔑 🔑	varchar(10)	latin1_swedish_ci		No	None
3	price	int(11)			Yes	NULL
4	availability	varchar(1)	latin1_swedish_ci		Yes	NULL

12> offers

#	Name	Type	Collation	Attributes	Null	Default
1	offer_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	description	varchar(50)	latin1_swedish_ci		Yes	NULL
3	r_id	varchar(5)	latin1_swedish_ci		Yes	NULL
4	given_discount	int(11)			No	None

13> order_content:

#	Name	Type	Collation	Attributes	Null	Default
1	o_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	r_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
3	food_id 🔑 🔑	varchar(5)	latin1_swedish_ci		No	None
4	quantity	int(11)			Yes	NULL

14> Orders:

#	Name	Type	Collation	Attributes	Null	Default
1	o_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	customer_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
3	block_no	varchar(5)	latin1_swedish_ci		Yes	NULL
4	street	varchar(20)	latin1_swedish_ci		Yes	NULL
5	landmark	varchar(20)	latin1_swedish_ci		Yes	NULL
6	city	varchar(20)	latin1_swedish_ci		Yes	NULL
7	state	varchar(20)	latin1_swedish_ci		Yes	NULL
8	pincode	varchar(6)	latin1_swedish_ci		Yes	NULL
9	p_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
10	o_date	datetime			Yes	NULL

15> payments

#	Name	Type	Collation	Attributes	Null	Default
1	p_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	customer_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
3	o_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
4	total	int(11)			Yes	NULL
5	payable	float			Yes	NULL
6	saved	float			Yes	NULL
7	p_mode	varchar(20)	latin1_swedish_ci		Yes	NULL
8	offer_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
9	coupon_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL

16> recipe:

#	Name	Type	Collation	Attributes	Null	Default
1	food_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	i_id 🔑 🔑	varchar(5)	latin1_swedish_ci		No	None
3	quantity	float			Yes	NULL

17> restaurant:

#	Name	Type	Collation	Attributes	Null	Default
1	r_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	r_name	varchar(20)	latin1_swedish_ci		Yes	NULL
3	rating	float			Yes	NULL
4	contact_no	varchar(10)	latin1_swedish_ci		Yes	NULL
5	website	varchar(100)	latin1_swedish_ci		Yes	NULL
6	email	varchar(50)	latin1_swedish_ci		Yes	NULL

18> reviews :

#	Name	Type	Collation	Attributes	Null	Default
1	review_id 🔑	varchar(5)	latin1_swedish_ci		No	None
2	r_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
3	customer_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
4	food_id 🔑	varchar(5)	latin1_swedish_ci		Yes	NULL
5	details	varchar(500)	latin1_swedish_ci		Yes	NULL

4. Procedures and triggers:

1> this is for the login part.

create or replace procedure Login() as

 cursor c_customer is select * from customer;

 r_customer c_customer%rowtype;

 Username varchar(20):=Username;

 password varchar(8):=password;

begin

 open c_customer;

 loop

 fetch c_customer into r_customer;

 if c_customer%notfound then exit;

 end if;

 if(username=r_customer.username)then

 if(password=r_customer.password)then

 dbms_output.put_line('successfully logged in');

 else

 dbms_output.put_line('password does not match');

 end if;

 else

 dbms_output.put_line('no username password');

 end if;

end loop;

close c_customer;

end;

Username

Password

Submit

Username and Password not matching.

Username

Password

Submit

Please fill all the fields.

2> this will fetch all records of the restaurants on the main screen.

```
create or replace procedure fetch_all_restaurants as
  cursor c_restaurant is select * from restaurant order by rating desc;
  r_restaurant c_restaurant%rowtype;
begin
  open c_restaurant;
  loop
    fetch c_restaurant into r_restaurant;
    if c_restaurant%notfound then exit;
    end if;
    dbms_output.put_line(r_restaurant.name||' '||r_restaurant.rating);
  end loop;
  close c_restaurant;
end;
```

ALL RESTAURANTS

According to their Ratings!!

<div>Mocha</div> <div>Ratings : 4.9</div> <div>Eat Here!!</div>	<div>Farki</div> <div>Ratings : 4.7</div> <div>Eat Here!!</div>	<div>Old School</div> <div>Ratings : 4.5</div> <div>Eat Here!!</div>
<div>Nini's Kitchen</div> <div>Ratings : 4.5</div> <div>Eat Here!!</div>	<div>Buttercupp</div> <div>Ratings : 4.3</div> <div>Eat Here!!</div>	<div>Sasuji</div> <div>Ratings : 4.1</div> <div>Eat Here!!</div>
<div>Domino's Pizza</div> <div>Ratings : 3.8</div> <div>Eat Here!!</div>		

3> it will display the menu category wise.

create or replace procedure disp_res_inf(id varchar) as

```
    cursor c_restaurant is select * from restaurant where restaurant_id=id;
    r_restaurant c_restaurant%rowtype;
    cursor c_menu is select * from menu where restaurant_id=id;
    r_menu c_menu%rowtype;
    cursor c_category is select * from category;
    r_category c_category%rowtype;
    food_name varchar(25);
begin
    open c_restaurant;
    loop
        fetch c_restaurant into r_restaurant;
        if c_restaurant%notfound then exit;
        end if;
        if(r_restaurant.restaurant_id = id)then
            dbms_output.put_line('Restaurant Name: '||r_restaurant.name);
            dbms_output.put_line('Rating: '|| r_restaurant.rating);
            dbms_output.put_line('Phone Number: '||r_restaurant.contact_no);
            dbms_output.put_line('website: '||r_restaurant.website);
            dbms_output.put_line('Email no: '||r_restaurant.email);
            dbms_output.put_line(' ');
            dbms_output.put_line('Menu');
        end if;
        open c_menu;
        loop
            fetch c_menu into r_menu;
            if c_menu%notfound then exit;
            end if;
            dbms_output.put_line('-----');
            select food_item.name into food_name from food_item where
r_menu.food_id=food_item.food_id;
            dbms_output.put_line(food_name||':- '||r_menu.price);
        end loop;
        close c_menu;
    end loop;
    close c_restaurant;
End;
```


Nini's Kitchen

Our Rating: 4.5/5

[Add To Favorites](#)
[Our Food](#) [Our Menu](#) [Our Malware](#)

SouthIndian



Punjabi



Gujarati



4> this will fetch the favourite food list.

```
create or replace procedure fetch_favourite_food(c_id varchar(6)) as
  cursor c_favourite_food is select * from favourite_favourite_food;
  cursor c_Order_content is select * from Order_content where order_id=(select order_id from
orders where customer_id=c_id);
  r_favourite_food c_favourite_food%rowtype;
  r_Order_content c_Order_content%rowtype;
  food_name varchar(30):='';
  cnt_odr int:=0;
  ctgry varchar(15):=' ';
begin
  open c_favourite_food;
  loop
    fetch c_favourite_food into r_favourite_food;
    if c_favourite_food%notfound then exit;
    end if;
    open c_Order_content;
    loop
      fetch c_Order_content into r_Order_content;
      if c_Order_content%notfound then exit;
      end if;
      if(r_favourite_food.food_id=r_Order_content.food_id)then
        select count(r_Order_content.order_id) into cnt_odr from r_Order_content where
r_Order_content.food_id=r_favourite_food.food_id;
        select food_item.name,food_item.category into food_name,ctgry from food_name
where r_Order_content.food_id=food_name.food_id;
        dbms_output.put_line((r_favourite_food.food_id||' '||food_name||' '||cnt_odr||'
'||ctgry);
      end if;
    end loop;
    close c_Order_content;
  end loop;
  close c_favourite_food;
End;
```

Favourite Food

Select From your all time favourites !!

Roman style pizza

Having rating : 4.6

From : Mocha

Having Category : Italian

ADD to CART

Know More

Remove

Paneer Kofta Naan

Having rating : 4.5

From : Nini's Kitchen

Having Category : Punjabi

ADD to CART

Know More

Remove

Dal rice

Having rating : 4.4

From : Nini's Kitchen

Having Category : Gujarati

ADD to CART

Know More

Remove

5> this will show your favourite restaurant list.

```
create or replace procedure fetch_favourite_restaurant(c_id varchar(6)) as
    cursor c_favourite_restaurant is select * from favourite_restaurant;
    cursor c_Order_content is select * from Order_content where order_id=(select order_id from
orders where customer_id=c_id);
    r_favourite_restaurant c_favourite_restaurant%rowtype;
    r_Order_content c_Order_content%rowtype;
    rest_name varchar(30);
    cnt_odr int:=0;
    rtng float:=0;
begin
    open c_favourite_restaurant;
    loop
        fetch c_favourite_restaurant into r_favourite_restaurant;
        if c_favourite_restaurant%notfound then exit;
        end if;
        open c_Order_content;
        loop
            fetch c_Order_content into r_Order_content;
            if c_Order_content%notfound then exit;
            end if;
            if(r_favourite_restaurant.restaurant_id=r_Order_content.restaurant_id)then
                select count(r_Order_content.order_id) into cnt_odr from Order_content where
r_Order_content.restaurant_id=r_favourite_restaurant.restaurant_id;
                select restaurant.name,restaurant.rating into rest_name,rtng from restaurant where
r_Order_content.restaurant_id=restaurant.restaurant_id;
                dbms_output.put_line((r_favourite_restaurant.restaurant_id||' '||rest_name||'
'||rtng||' '||cnt_odr);
            end if;
        end loop;
        close c_Order_content;
    end loop;
    close c_favourite_restaurant;
End;
```

Favourite Restaurants

Select From your all time favourites !!

Farki

Having rating : 4.7

Eat Here!!

Remove

Nini's Kitchen

Having rating : 4.5

Eat Here!!

Remove

6>this will show the item from your cart.

```
create or replace procedure fetch_all_cart_items(customer_id varchar) as
  cursor c_Cart is select * from Cart;
  r_Cart c_Cart%rowtype;
  food_name varchar(30);
  Rest_name varchar(25);
  price int:=0;
  Tprice int:=0;
  bill_amount int:=0;
  Tax float:=0;
begin
  open c_Cart;
  loop
    fetch c_Cart into r_Cart;
    if c_Cart%notfound then exit;
    end if;
    if r_Cart.customer_id=customer_id then
      select food_item.name into food_name from food_item where
food_item.food_id=r_Cart.food_id;
      select restaurant.name into Rest_name from restaurant where
restaurant.restaurant_id=r_Cart.restaurant_id;
      select Menu.price into price from Menu where Menu.food_id=r_Cart.food_id and
Menu.restaurant_id=r_Cart.restaurant_id;
      dbms_output.put_line(food_name||' - '||price);
      dbms_output.put_line('from: '||food_name);
      Tprice:=price*r_Cart.quantity;
      dbms_output.put_line('Quantity: '||r_Cart.quantity||' Total price: '||Tprice);
      bill_amount:= bill_amount + Tprice;

      dbms_output.put_line('-----');
    end if;
  end loop;
  close c_Cart;
  dbms_output.put_line('Bill amount: '||bill_amount);
  Tax:=bill_amount*0.18;
  dbms_output.put_line('Tax: '||Tax);
  dbms_output.put_line('Bill amount with tax: '||bill_amount+Tax);
End;
```


Your Cart

Have a look at it, Manage it, Order it!!

Check Out

Clear Cart

Masala dosa

From : Nini's Kitchen

Total Price : 80

-

Quantity : 1

+

Remove

Maysore masala dosa

From : Nini's Kitchen

Total Price : 170

-

Quantity : 2

+

Remove

Dal rice

From : Nini's Kitchen

Total Price : 140

-

Quantity : 2

+

Remove

7> this will show all offers.

create or replace procedure fetch_all_offers as

 cursor c_offers is select * from offers;

 r_offers c_offers%rowtype;

begin

 open c_offers;

 loop

 fetch c_offers into r_offers;

 if c_offers%notfound then exit;

 end if;

 dbms_output.put_line('-----');;

 dbms_output.put_line('Offer code:'||r_offers.offer_id);

 dbms_output.put_line('Description'||r_offers.Description);

 end loop;

 close c_offers;

End;

HELTHOHOLIC

Categories • OFFER • FAVORITE FOODS • FAVORITE RESTAURANTS • CMB • ORDER HISTORY • PROFILE

Make Your Payment

Get your order Delivered

Now you can't change your mind. There is no way going BACK.

Choose Payment Mode

Select Payment Type ▾

You must have to choose Payment Mode.

Have a quick promooode guide

Guide

ADD PROMOCODE

Coupons :

BDAY - Get 80% discount on your Birthday

FIRST - Get 15% discount on your first order

Offers :

OF_1 - Get 40% off on orders with minimum order amount 200

Total Amount of Order : 305 Rs.

8> this will display the check out form.

```
create or replace procedure price_cart(cus_id varchar2)as
cursor cur_quan is select quantity,food_name from cart,food_item where cart.customer_id =
cus_id and food_item.food_id = cart.food_id;
cursor cur_price is select price from menu ,cart where cus_id = cart.customer_id and
cart.food_id = menu.food_id;
r_quan cur_quan%rowtype;
r_price cur_price%rowtype;
temp_total int:=0;
begin
for r_quan in cur_quan loop
  for r_price in cur_price loop
    dbms_output.put_line('name of the food is: '||r_quan.food_name||'price
is:'||r_price.price||'quantity is: '||r_quan.quantity);
    temp_total := temp_total + r_quan.quantity*r_price.price;
  end loop;
end loop;
  dbms_output.put_line('total price is:'||temp_total);
end;

declare
cus_id varchar2(10):=:cus_id;
begin
price_cart(cus_id);
end;
```

Checking Out From
Nini's Kitchen
Here is your final cart!
You can go back to cart, Make changes, checkout again.
[Go Back](#)

1. vada sambhar
Quantity : 1
Total Price : 60

2. Paneer Bhurji Naan
Quantity : 1
Total Price : 105

3. Dal rice
Quantity : 2
Total Price : 140

Final Cart Amount **305 Rs.**
If everything is fine than you can move to payments.
[PAY!!](#)

9> this will show the payment pathway.

```
create or replace procedure payment(cus_id varchar2,coup_id varchar2,off_id varchar2) as
cursor cur_coupon is select coupon_id,description,coupon_discount from coupons where
coupons.coupon_id = coup_id;
cursor cur_offer is select * from offers where offers.offer_id = off_id;
cursor cur_quan is select quantity,food_name from cart,food_item where cart.customer_id =
cus_id and food_item.food_id = cart.food_id;
cursor cur_price is select price from menu ,cart where cus_id = cart.customer_id and
cart.food_id = menu.food_id;
r_offer cur_offer%rowtype;
r_coupon cur_coupon%rowtype;
r_quan cur_quan%rowtype;
r_price cur_price%rowtype;

temp_total int:=0;
begin
for r_coupon in cur_coupon loop
    dbms_output.put_line('available coupons are: ');
    dbms_output.put_line('coupon id is: '||r_coupon.coupon_id||' coupon description is:
'||r_coupon.description||' discount is: '||r_coupon.coupon_discount);
    for r_offer in cur_offer loop
        dbms_output.put_line('available offers are: ');
        dbms_output.put_line('offer id is: '||r_offer.offer_id||' offer description is:
'||r_offer.description||' discount is: '||r_offer.offer_discount);

        for r_quan in cur_quan loop
            for r_price in cur_price loop
                dbms_output.put_line('name of the food is: '||r_quan.food_name||'price
is:'||r_price.price||'quantity is: '||r_quan.quantity);
                temp_total := temp_total + ((r_quan.quantity*r_price.price)) -
(r_coupon.coupon_discount*(r_quan.quantity*r_price.price)/100 ) -
(r_offer.offer_discount*(r_quan.quantity*r_price.price)/100);
            end loop;
        end loop;
    end loop;
end loop;

dbms_output.put_line('total price is:'||temp_total);

end;
```

```
declare
cus_id varchar2(10):=:cus_id;
coup_id varchar2(10):=:coup_id;
off_id varchar2(10):=:off_id;
```

```
begin
payment(cus_id,coup_id,off_id);
end;
```

HELTHOHOLIC

Categories ▾ OFFERS FAVOURITE FOODS FAVOURITE RESTAURANTS CART ORDER HISTORY PROFILE

Make Your Payment

Get your order Delivered

Now you can't change your mind. There is no way going BACK.

Choose Payment Mode

Select Payment Type ▾

You must have to choose Payment Mode.

Have a quick promooocde guide

Guide ADD PROMOCODE

Total Amount of Order : 305 Rs.

10> this will show your full order.

```
create or replace procedure fetch_order(day int,c_id varchar) as
  cursor c_Orders is select * from Orders;
  cursor c_Payments is select * from Payments;
  r_Orders c_Orders%rowtype;
  r_Payments c_Payments%rowtype;
  t_date date;
  o_time date;
begin
  open c_Orders;
  loop
    fetch c_Orders into r_Orders;
    if c_Orders%notfound then exit;
    end if;
    open c_Payments;
    loop
      fetch c_Payments into r_Payments;
      if c_Payments%notfound then exit;
      end if;
      select sysdate into t_date from dual;
      if(r_Orders.order_id=r_Payments.order_id and (t_date - day)<
to_date(to_char(r_Orders.date_time,'MM-DD-YYYY'),'MM-DD-YYYY') and
r_Order.customer_id=c_id)then
        dbms_output.put_line(r_Orders.order_id||' '|| r_Payments.payment_id||'
'||r_Payments.payable_amount||' '||r_Payments.payment_mode||' '||r_Payments.saved_amount);
      end if;
    end loop;
    close c_Payments;
  end loop;
  close c_Orders;
end;
```


Checking Out From

Nini's Kitchen

Here is your final cart!

You can go back to cart, Make changes, checkout again.

[Go Back](#)

1. vada sambhar
Quantity : 1
Total Price : 60

2. Paneer Bhurji Naan
Quantity : 1
Total Price : 105

3. Dal rice
Quantity : 2
Total Price : 140

Final Cart Amount 305 Rs.

If everything is fine than you can move to payments.

[PAY!!](#)

11> this will display nutrition gain per order.

```
create or replace procedure disp_nutri(id varchar) as
  cursor c_food_item is select * from food_item where food_id=id;
  r_food_item c_food_item%rowtype;
  cursor c_recipe is select * from recipe where food_id=id;
  r_recipe c_recipe%rowtype;
  cursor c_ingredients is select * from ingredients;
  r_ingredients c_ingredients%rowtype;
  fat float:=0;
  Carb float:=0;
  fiber float:=0;
  protein float:=0;
  calory float:=0;
begin
  open c_food_item;
  loop
    fetch c_food_item into r_food_item;
    if c_food_item%notfound then exit;
    end if;
    open c_recipe;
    loop
      fetch c_recipe into r_recipe;
      if c_recipe%notfound then exit;
      end if;
      open c_ingredients;
      loop
        fetch c_ingredients into r_ingredients;
        if c_ingredients%notfound then exit;
        end if;
        if(r_food_item.food_id=r_recipe.food_id and
r_recipe.ingredient_id=r_ingredients.ingredient_id)then
          fat := fat + (r_ingredients.fat*r_recipe.quantity);
          Carb := Carb + (r_ingredients.carbs*r_recipe.quantity);
          fiber := fiber + (r_ingredients.fiber*r_recipe.quantity);
          protein := protein + (r_ingredients.proteine*r_recipe.quantity);
          calory := calory + (r_ingredients.calory*r_recipe.quantity);
        end if;
      end loop;
    end loop;
    close c_ingredients;
  end loop;
  close c_recipe;
```

```
end loop;  
close c_food_item;  
dbms_output.put_line('Total Nutrition:');  
dbms_output.put_line('fat:'||fat||' Carb:'||Carb||' fiber:'||fiber||' protein:'||protein||'  
calories:'||calory);  
End;
```

```
Total Nutrition:  
fat:202.2109 Carb:721.92 fiber:40.438 protein:47.281 calories:5882.22  
Statement processed.
```

12> this will show an age-wise popular restaurant list.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `age_wise_resturant`(IN  
`temp_age` INT) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY  
DEFINER  
BEGIN  
  
SELECT COUNT(order_content.o_id), restaurant.r_name FROM  
orders,restaurant,customer,order_content where (customer.age>temp_age AND  
customer.age<temp_age+20) AND (orders.customer_id = customer.customer_id)  
AND (order_content.r_id = restaurant.r_id);  
  
END
```

13> this will show which mode of payment are used how many times.

Create or replace procedure most_used_pay_mode(cus_id varchar2) AS

Cursor pay_mode is select p_mode from payment where payment.customer_id =
cus_id;

r_pay_mode pay_mode%rowtype;

Count_cash number;

Count_online number;

Count_credit_card number;

begin

For r_pay_mode in pay_mode loop

if(r_pay.p_mode = 'cash') Then

Count_cash := count_cash+1;

Elsif (r_pay.p_mode = 'online') then

Count_online := count_online +1;

Elsif (r_pay.p_mode = 'credit_card') then

Count_credit_card := count_credit_Card + 1;

End if;

dbms.output_put_line('cash is used: '||count_cash||'online is
used'||count_online||'credit card is used: '||credit_card);

End loop;

End;

14> Nutrition gain per rupee in a particular order.

create or replace procedure disp_order_nutri(id int) as

```
    cursor c_order_content is select * from order_content where order_id=id;
```

```
    r_order_content c_order_content%rowtype;
```

```
    cursor c_recipe is select * from recipe;
```

```
    r_recipe c_recipe%rowtype;
```

```
    cursor c_ingredients is select * from ingredients;
```

```
    r_ingredients c_ingredients%rowtype;
```

```
    fat float:=0;
```

```
    Carb float:=0;
```

```
    fiber float:=0;
```

```
    protein float:=0;
```

```
    calory float:=0;
```

```
    Tprice int:=0;
```

```
begin
```

```
    open c_order_content;
```

```
    loop
```

```
        fetch c_order_content into r_order_content;
```

```
        if c_order_content%notfound then exit;
```

```
    end if;
```

```
    open c_recipe;
```

```
    loop
```



```

    fetch c_recipe into r_recipe;

    if c_recipe%notfound then exit;

    end if;

    open c_ingredients;

    loop

        fetch c_ingredients into r_ingredients;

        if c_ingredients%notfound then exit;

        end if;

        if(r_order_content.food_id=r_recipe.food_id and
r_recipe.ingredient_id=r_ingredients.ingredient_id)then

            fat := fat + (r_ingredients.fat*r_recipe.quantity *
r_order_content.quantity);

            Carb := Carb + (r_ingredients.carbs*r_recipe.quantity *
r_order_content.quantity);

            fiber := fiber + (r_ingredients.fiber*r_recipe.quantity *
r_order_content.quantity);

            protein := protein + (r_ingredients.proteine*r_recipe.quantity *
r_order_content.quantity);

            calory := calory + (r_ingredients.calory*r_recipe.quantity *
r_order_content.quantity);

            end if;

        end loop;

        close c_ingredients;

    end loop;

    close c_recipe;

```

```

end loop;

close c_order_content;

dbms_output.put_line('Total Nutrition from order:');

dbms_output.put_line('fat:'||fat||' Carb:'||Carb||' fiber:'||fiber||' protein:'||protein||'
calories:'||calory);

select Payments.payable_amount into Tprice from Payments where
Payments.order_id=id;

dbms_output.put_line('Total Nutrition from order per 1 Rupee:');

dbms_output.put_line('fat:-'||fat/Tprice||' Carb:-'||Carb/Tprice||' fiber:-'||fiber/Tprice||'
protein:-'||protein/Tprice||' calories:-'||calory/Tprice);

end;

declare

begin

disp_order_nutri(166);

end;

```

```

Total Nutrition from order:
fat:507.2488 Carb:1582.843 fiber:83.58 protein:139.028 calories:13347.64

```

```

-----
Total Nutrition from order per 1 Rupee:
fat: .2498762561576354679802955665024630541872 Carb: .7797256157635467980295566502463054187192 fiber: .0411724137931034482758620689655172413793
protein: .068486699507389162561576354679802955665 calories: 6.57519211822660098522167487684729064039

```

```

Statement processed.

```

15> this procedure is used for fetching all orders.

create or replace procedure fetch_order(day int,c_id varchar) as

cursor c_Orders is select * from Orders;

cursor c_Payments is select * from Payments;

r_Orders c_Orders%rowtype;

r_Payments c_Payments%rowtype;

t_date date;

o_time date;

begin

open c_Orders;

loop

fetch c_Orders into r_Orders;

if c_Orders%notfound then exit;

end if;

open c_Payments;

loop

fetch c_Payments into r_Payments;

if c_Payments%notfound then exit;

end if;

select sysdate into t_date from dual;

if(r_Orders.order_id=r_Payments.order_id and (t_date - day)<
to_date(to_char(r_Orders.date_time,'MM-DD-YYYY'),'MM-DD-YYYY') and
r_Order.customer_id=c_id)then

```
        dbms_output.put_line(r_Orders.order_id||' '|| r_Payments.payment_id||  
'||r_Payments.payable_amount||' '||r_Payments.payment_mode||'  
'||r_Payments.saved_amount);  
  
        end if;  
  
    end loop;  
  
    close c_Payments;  
  
end loop;  
  
close c_Orders;  
  
end;
```

16> this will fetch all record of order content.

create or replace procedure disp_order(id int) as

cursor c_order_content is select * from order_content where order_id=id ;

r_Order_content c_Order_content%rowtype;

food_name varchar(6);

rest_name varchar(6);

price int;

Tprice int;

begin

open c_Order_content;

loop

fetch c_Order_content into r_Order_content;

if c_Order_content%notfound then exit;

end if;

if(r_Order_content.order_id=id)then

select food_item.name into food_name from food_item where
food_item.food_id= r_Order_content.food_id;

select restaurant.name into rest_name from restaurant where
restaurant.restaurant_id= r_Order_content.restaurant_id;

select menu.price into price from menu where menu.food_id=
r_Order_content.food_id and menu.restaurant_id= r_Order_content.restaurant_id;

dbms_output.put_line(food_name||' Quantity:'||r_Order_content.quantity||'
Price:'||price);

Tprice:= Tprice + (r_Order_content.quantity * price);

```
        end if;

    end loop;

    close c_Order_content;

    dbms_output.put_line('Order Id:'||r_Order_content.order_id||' Restaurant name: '||
rest_name||' Total amount:'||Tprice);

end;
```

17> delete the cart after order placing the order.

```
CREATE PROCEDURE DELETE_cart()
BEGIN
  DECLARE finished INTEGER DEFAULT 0;
  DECLARE temp_customer_id varchar(5) DEFAULT "";
  DECLARE cur_del CURSOR FOR SELECT customer_id FROM cart;
  DECLARE CONTINUE HANDLER
  FOR NOT FOUND SET finished = 1;
  OPEN cur_del;
getcart: LOOP
  FETCH cur_del INTO temp_customer_id;
  IF finished = 1 THEN
    DELETE FROM cart where old.customer_id = temp_customer_id;
    LEAVE getcart;
  END IF;
END LOOP getcart;
CLOSE cur_del;
END;
```

Your Cart

Have a look at it, Manage it, Order it!!



Your Cart is Empty... Go and buy something !!

→ triggers

1> validation for favourite food.

```
CREATE TRIGGER prevent_duplication BEFORE INSERT ON favourite_food
FOR EACH ROW
BEGIN
  IF (EXISTS(SELECT * FROM favourite_food WHERE food_id = NEW.food_id AND r_id
    = NEW.r_id AND customer_id = NEW.customer_id)) THEN
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'It is already in
    your favourite list!!';
  END IF;
END;
```

It is already in your favourite list!!

2> this will prevent wrong entry on favourite food.

```
CREATE TRIGGER prevent_wrong_entry BEFORE INSERT ON favourite_food
```

```
FOR EACH ROW
```

```
IF NOT EXISTS(SELECT * from menu m WHERE m.food_id = NEW.food_id AND  
m.r_id = NEW.r_id) THEN
```

```
    SIGNAL SQLSTATE VALUE '45001' SET MESSAGE_TEXT = 'This restaurant  
dont sell this food item. Sorry !!';
```

```
END IF;
```

```
END;
```

3> this trigger will prevent wrong entry on favourite restaurant.


```
CREATE TRIGGER prevent_wrong_entry BEFORE INSERT ON favourite_restaurant  
FOR EACH ROW
```

```
IF (EXISTS(SELECT * FROM favourite_food WHERE r_id = NEW.r_id AND  
customer_id = NEW.customer_id)) THEN
```

```
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'It is already in  
your favourite list!!';
```

```
END IF;
```

```
END;
```



It is already in your favourite list!!

4> this trigger will prevent the duplication of the data in the cart.

```
CREATE TRIGGER prevent_duplication_cart BEFORE INSERT ON cart
```

```
FOR EACH ROW
```

```
IF (EXISTS(SELECT * FROM cart c WHERE c.food_id = NEW.food_id AND c.r_id =  
NEW.r_id AND c.customer_id = NEW.customer_id)) THEN
```

```
    SIGNAL SQLSTATE VALUE '45000' SET MESSAGE_TEXT = 'It is already in  
your Cart!!';
```

```
ELSEIF EXISTS(SELECT * from cart where r_id = NEW.r_id and customer_id =  
NEW.customer_id) THEN
```

```
    SIGNAL SQLSTATE VALUE '45002' SET MESSAGE_TEXT = 'You cannot add  
this in cart it. At one time you can order from one restaurant only!!';
```

```
ELSE
```

```
    SIGNAL SQLSTATE VALUE '45001' SET MESSAGE_TEXT = 'sucessfully  
added';
```

```
END IF;
```

```
END;
```

It is already in your Cart!!

x

You can only add food items from one restaurant for one order. Items from multiple restaurants in cart is not allowed.

x

5> this trigger will ensure your success entry.

```
CREATE TRIGGER success_entry BEFORE INSERT ON cart
```

```
FOR EACH ROW
```

```
IF (NOT EXISTS(SELECT * from cart c where c.customer_id =  
NEW.customer_id))THEN
```

```
    INSERT INTO cart  
VALUES(new.customer_id,new.r_id,new.food_id,new.quantity);
```

```
SIGNAL SQLSTATE VALUE '45001' SET MESSAGE_TEXT = 'successfully added to  
your cart'
```

```
end IF;
```

```
END;
```

This food item is successfully added to your Cart!! :)

×