PetAtl

Online Retail Store

Project Document

|  |  |
| --- | --- |
| **Prepared by:** | Group 6 Team |
| **Date:** | 10/05/2017 |
|  |  |
| **Course**  **Professor:** | DBMS  Dr Yu Kai Lin |

**GROUP 6**

Apurva Ranade

Archana Suresh Date

Vijay Paul Singh Naik Zarapala

Vinamratha Kuttugul Veeresh

Contents

[1. Conceptual Modeling 3](#_Toc494985273)

[1.1. Conceptual ER diagram 3](#_Toc494985274)

[1.2. Explanation of Conceptual ER diagram: 4](#_Toc494985275)

[1.3. Business Rules: 6](#_Toc494985276)

[2. ER Model Mapping 6](#_Toc494985277)

[2.1. Detailed model mapping steps (Cardinality): 6](#_Toc494985278)

[2.2. Detailed model mapping steps 7](#_Toc494985279)

[2.3. Logical model diagram (Primary Key 🡪 Foreign Key Mapping) 8](#_Toc494985280)

[2.4. Final list of relations (tables) 8](#_Toc494985281)

[3. Relation NormalizatioN 9](#_Toc494985282)

[3.1. Full Functional dependency analysis 9](#_Toc494985283)

[3.2. Transitive Functional Dependency Analysis 10](#_Toc494985284)

[3.3. 2NF Normalization 10](#_Toc494985285)

[3.4. 3NF Normalization 10](#_Toc494985286)

[4. Database implementation 10](#_Toc494985287)

[4.1. Physical table design 10](#_Toc494985288)

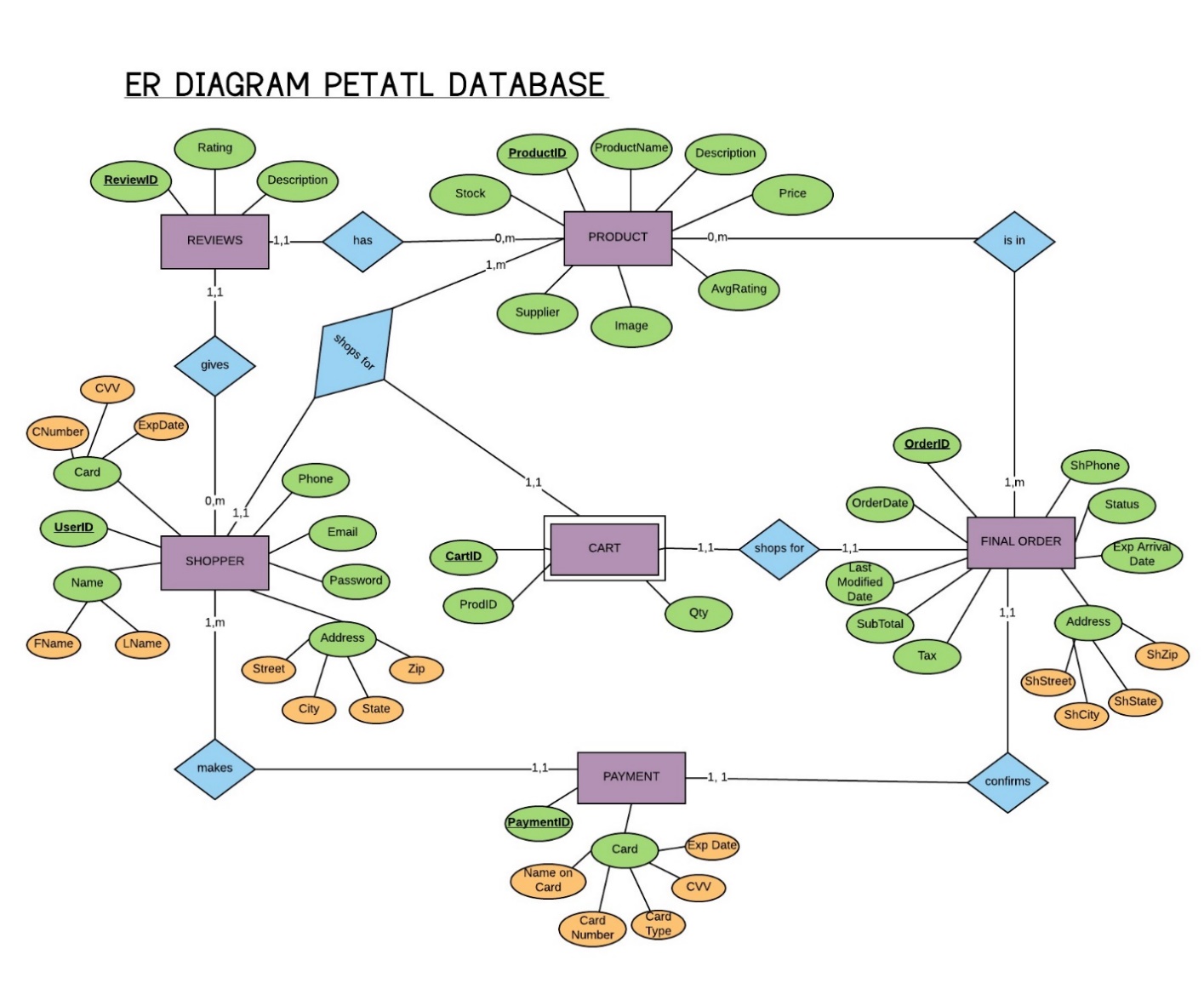
[5. Web application & SQL BASED APPLICATION desigN 29](#_Toc494985289)

[5.1. Graphical USer interface 29](#_Toc494985290)

[6. User Requirements Analysis 38](#_Toc494985291)

## **Conceptual Modeling**

## **Conceptual ER diagram**



## **Explanation of Conceptual ER diagram:**

|  |
| --- |
| **SHOPPER ENTITY: Represents online shoppers at PetATL** |
| * **UserID** – Unique identifier of shopper * Name (Fname, Lname ) – the name of the shopper * Email – the email address of the shopper, always unique * Password– unique password to log into the shopper account * Phone number – phone numbers of shopper (Assumption: single-valued attribute) * Address (Street, City, State, Zip) – the address of the shopper * Card (CNumber, CVV, ExpDate) –credit card details given during account creation |

|  |
| --- |
| **PRODUCT ENTITY:** Represents products available at PetATL online store |
| * **ProductID** – Unique identifier of each product * ProductName – the name of the product * Description – gives description of the product * Price – unit price of the product * Supplier – supplier of the product * -Image – stores the link of the product image * Average Rating – derived attribute based on all the review scores |

|  |
| --- |
| **REVIEW ENTITY:** Stores the product reviews and ratings at PetATL online store |
| * **ReviewID** – Unique identifier of each product review * Rating – stores the rating of the Review * Description – gives description of the Review |

|  |
| --- |
| **CART ENTITY:** weak entity, depends on shopper entity, has details of products for order |
| * **CartID** – Unique identifier of each cart instance * ProductID – represents the product in the cart * Qty – represents the quantity of the corresponding product in the cart |

|  |
| --- |
| **FINALORDER ENTITY:** Represents the details of the placed order |
| * **OrderID** – Unique identifier of each order * Order Date – date of the order placed * Last Modified Date – gives the date when the product was last modified or cancelled * SubTotal – total sum of the product price X quantity of each product in the order * Tax – tax calculated on the sub total * Address (ShStreet, ShCity, ShState, ShZip) – the shipping address of the order * ShPhone – phone number of the person at shipping address * Expected Arrival Date – expected date of order delivery to shopper * Status – status of the order (yet to be shipped, in-transit, delivered, cancelled) |

|  |
| --- |
| **PAYMENT ENTITY:** Represents the payment details of the order |
| * **PaymentID** – Unique identifier of each payment * Card (NameOnCard, Card number, Card Type, CVV, ExpDate) – details of the card used to place order |

## **Business Rules:**

1. User will shop for products and adds product/products to cart (ternary relationship)
2. The first name of the shopper is considered as username for simplicity of operation
3. The order is confirmed only after the user makes payment through card for an order
4. While creating an account, user can enter details of any card and can use a different card while making a payment for an order
5. Every cart has a unique cart id, user must log in to add products into the cart
6. Once the order is successfully placed, the data from cart is deleted
7. The billing address is same as the address used during account creation

## **ER Model Mapping**

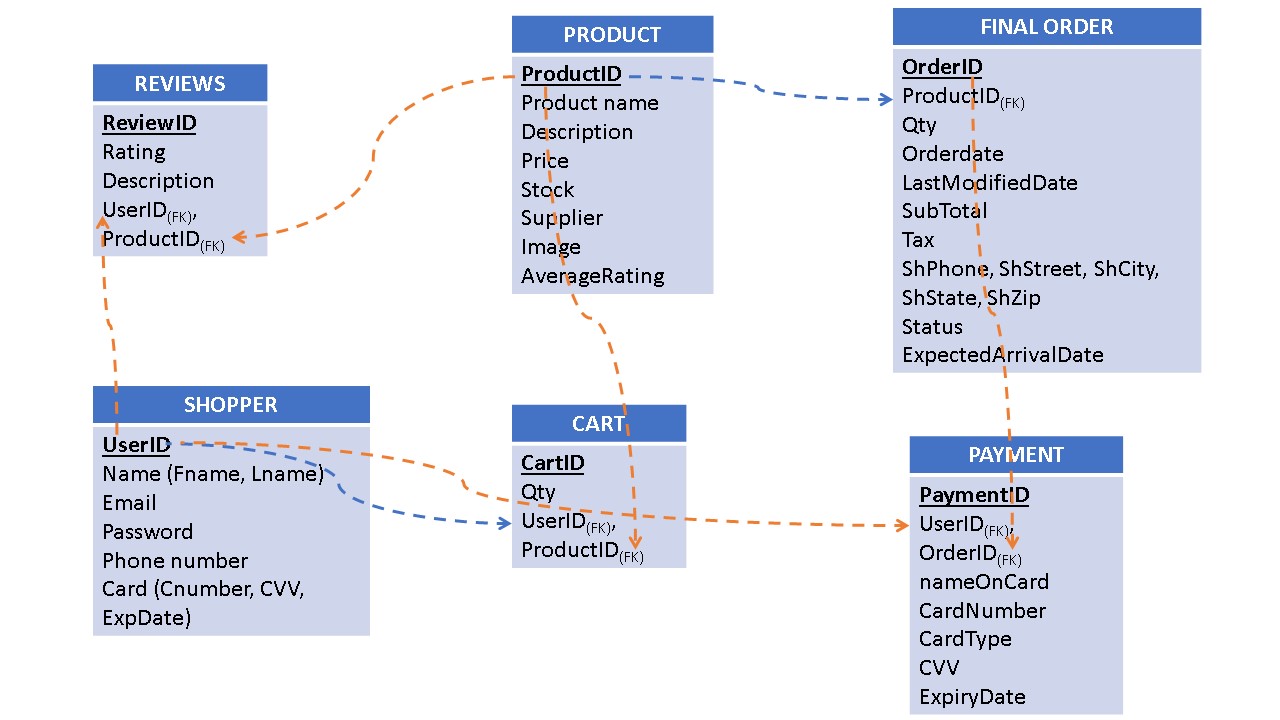
## **Detailed model mapping steps (Cardinality):**

* Shopper (0,m) 🡨🡪 (1,1) Review
  + Shopper may give many reviews (0, m)
  + A review can be given by only one shopper (1,1)
* Shopper (1,m) 🡨🡪 (1,1) Payment
  + Shopper can make 1 or many payments
  + A payment can be made by only one shopper (1,1)
* Payment (1,1) 🡨🡪 (1,1) Final Order
  + Only one payment can be made for one final order
  + One final order can be placed by one payment
* Shopper (1,1) 🡨🡪 (1,1) Cart
  + A shopper can have only one cart
  + A cart can be loaded by only one shopper
* Cart (1,1) 🡨🡪 (1,1) Final Order
  + A Cart can be converted to only one final order
  + A final order can be made from only one cart
* Product (1,m) 🡨🡪 (1,1) Cart
  + A product can be added into one or many carts of many shoppers
  + A cart can have one product (quantity is assumed to be optional)
* Product (0,m) 🡨🡪 (1,m) Final Order
  + A product may be in many final order
  + A final order must contain one or many products (1,m)
  + PRODUCT ORDER table is created to represent M-M relationship

## **Detailed model mapping steps**

* 1. UserID of Shopper is mapped to tables Reviews, Cart, Payment as Foreign key
     + 1. UserID 🡪 Reviews
       2. UserID 🡪 Cart
       3. UserID 🡪 Payment
  2. ProductID of Product is mapped to tables Reviews, Cart, Final Order as Foreign key
     + 1. ProductID 🡪 Reviews
       2. ProductID 🡪 Cart
       3. ProductID 🡪 FinalOrder
  3. OrderID of Final Order is mapped to table Payment as Foreign Key
     + 1. OrderID 🡪 Payment

## **Logical model diagram (Primary Key 🡪 Foreign Key Mapping)**



## **Final list of relations (tables)**

* Product (ProductID, Product name, Description, Price, Stock, Supplier, Image, AverageRating)
* Shopper (**UserID**, Fname, Lname, email, password, phone, street, city, state, zip, cnumber, cvv, expdate)
* Reviews (**ReviewID**, UserID(FK), ProductID(FK), rating, description)
* Cart (**CartID**, ProductID(FK), UserID(FK), Qty)
* ProductOrder (**OrderID**, ProductID(FK), Qty)
* FinalOrder (**OrderID**, Orderdate, LastModifiedDate, SubTotal, tax, ShPhone, ShStreet, ShCity, ShState, ShZip, status, ExpectedArrivalDate)
* Payment (**PaymentID**, UserID(FK), OrderID(FK), nameOnCard, CardNumber, CardType, CVV, ExpiryDate)

## **Relation NormalizatioN**

## **Full Functional dependency analysis**

|  |  |
| --- | --- |
| * 1. **PRODUCT** | |
| * 1. ProductID | * 1. Product name, Description, Price, Stock, Supplier, Image, AverageRating |

|  |  |
| --- | --- |
| * 1. Shopper | |
| * 1. **UserID** | * 1. Fname, Lname, email, password, phone, street, city, state, zip, cnumber, cvv, expdate |

|  |  |
| --- | --- |
| Reviews | |
| **ReviewID** | rating, description |

|  |  |
| --- | --- |
| Cart | |
| **CartID** | Qty |

|  |  |
| --- | --- |
| ProductOrder | |
| **OrderID, ProductID**(FK) | * 1. Qty |

|  |  |
| --- | --- |
| Payment | |
| **PaymentID** | nameOnCard, CardNumber, CardType, CVV, ExpiryDate |

|  |  |
| --- | --- |
| FINALORDER | |
| **OrderID** | Orderdate, LastModifiedDate, SubTotal, tax, ShPhone, ShStreet, ShCity, ShState, ShZip, status, ExpectedArrivalDate |

## **Transitive Functional Dependency Analysis**

There are no transitive dependencies in all the above relations. As all the attributes are directly related to primary key.

## **2NF Normalization**

There is no need for 2NF normalization as the tables are directly in 3NF.

## **3NF Normalization**

All the above relations are already in 3NF. We conclude this since:

* Redundancy of data is not present anywhere in the relations.
* There are no issues for data integrity.
* There are no create, update or delete anomalies.
* All the attributes are directly related to primary key

## **Database implementation**

## **Physical table design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SHOPPER |  |  |  |  |
| Attribute Name | **Type** | **Key** | **Nullable** | **Referential Constraint** |
| UserID | Int | PK | No |  |
| FName | Varchar2 (50) |  | No |  |
| LName | Varchar2 (50) |  | No |  |
| Email | Varchar2 (50) |  | No |  |
| Password | Varchar2 (50) |  | No |  |
| Phone | Varchar2 (50) |  | Yes |  |
| Street | Varchar2 (50) |  | Yes |  |
| City | Varchar2 (50) |  | Yes |  |
| State | Varchar2 (50) |  | Yes |  |
| Zip | Int |  | No |  |
| CardNumber | Varchar2 (50) |  | No |  |
| CVV | Int |  | No |  |
| ExpiryDate | Date |  | No |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRODUCT |  |  |  |  |
| Attribute Name | Type | Key | Nullable | Referential Constraint |
| ProductID | Int | PK | No |  |
| Pname | Varchar2 (50) |  | No |  |
| Description | Varchar2 (50) |  | No |  |
| Supplier | Varchar2 (50) |  | No |  |
| Price | Float |  | No |  |
| Stock | Int |  | No |  |
| Image | Varchar2 (50) |  | Yes |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CART |  |  |  |  | |
| Attribute Name | Type | Key | Nullable | Referential Constraint | |
| CartID | Int | PK | No |  |
| ProductID | Int | FOREIGN KEY(ProductID) REFERENCES Product(ProductID) | No | ON delete cascade |
| UserID | Int | FOREIGN KEY (UserID)  REFERENCES Shopper(UserID) | No | ON delete cascade |
| Qty | Int |  | No |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVIEW |  |  |  |  | |
| Attribute Name | Type | Key | Nullable | Referential Constraint | |
| ReviewID | Int | PK | No |  |
| ProductID | Int | FOREIGN KEY(ProductID) REFERENCES Product(ProductID) | No | ON delete cascade |
| UserID | Int | FOREIGN KEY (UserID)  REFERENCES Shopper(UserID) | No | ON delete cascade |
| Rating | Int |  | No |  |
| ReviewDescription | Varchar2(1000) |  | Yes |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PRODUCTORDER |  |  |  |  | |
| Attribute Name | Type | Key | Nullable | Referential Constraint | |
| OrderID | Int | PK, FOREIGN KEY (ORDERID)  REFERENCES FINALORDER(ORDERID) | No | ON delete cascade |
| ProductID | Int | PK, FOREIGN KEY(ProductID) REFERENCES Product(ProductID) | No | ON delete cascade |
| Qty | Int |  | No |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FINALORDER |  |  |  |  |
| Attribute Name | Type | Key | Nullable | Referential Constraint |
| OrderID | Int | PK | No |  |
| UserID | Int | FOREIGN KEY (UserID)  REFERENCES Shopper(UserID) | No | ON delete cascade |
| ShPhone | Varchar2 (50) |  | No |  |
| ShStreet | Varchar2 (50) |  | No |  |
| ShCity | Varchar2 (50) |  | No |  |
| ShState | Varchar2 (50) |  | No |  |
| ShZip | Int |  | No |  |
| Tax | Float |  | No |  |
| SubTotal | Float |  | No |  |
| OrderDate | Date |  | No |  |
| Status | Varchar2 (50) |  | Yes |  |
| LastModified | Date |  | No |  |
| ExpectedArrivalDate | Date |  | Yes |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAYMENT |  |  |  |  |
| Attribute Name | Type | Key | Nullable | Referential Constraint |
| PaymentID | Int | PK | No |  |
| OrderID | Int | FOREIGN KEY (ORDERID)REFERENCES FINALORDER(ORDERID) | No | ON delete cascade |
| P\_CardNumber | Varchar2 (50) |  | No |  |
| P\_Name\_on\_Card | Varchar2 (50) |  | No |  |
| P\_CardType | Varchar2 (50) |  | No |  |
| P\_ExpiryDate | Date |  | No |  |
| P\_CVV | Int |  | Yes |  |

## **Creation of sequences**

-----------------------------sequence creation------------------------

CREATE SEQUENCE userid\_seq

START WITH 1001

INCREMENT BY 1;

CREATE SEQUENCE finalorderid\_seq

START WITH 7001

INCREMENT BY 1;

CREATE SEQUENCE cartid\_seq

START WITH 3001

INCREMENT BY 1;

CREATE SEQUENCE reviewid\_seq

START WITH 9001

INCREMENT BY 1;

CREATE SEQUENCE payment\_seq

START WITH 11001

INCREMENT BY 1;

## **Creation of tables**

-----------------------------create table statements------------------

CREATE TABLE Shopper (

UserID INT NOT NULL,

Fname VARCHAR2(50) NOT NULL,

Lname VARCHAR2(50) NOT NULL,

email VARCHAR2(50) NOT NULL,

password VARCHAR2(50) NOT NULL,

phone VARCHAR2(50),

street VARCHAR2(300),

city VARCHAR2(50),

state VARCHAR2(50),

zip INT NOT NULL,

cardnumber VARCHAR2(50) NOT NULL,

CVV INT NOT NULL,

expirydate DATE NOT NULL,

CONSTRAINT Userid\_pk

PRIMARY KEY ( UserID )

);

CREATE TABLE Product (

ProductID INT NOT NULL,

Pname VARCHAR2(1000) NOT NULL,

description VARCHAR2(4000) NOT NULL,

supplier VARCHAR2(100) NOT NULL,

price FLOAT NOT NULL,

stock INT NOT NULL,

image VARCHAR2(2000),

CONSTRAINT Productid\_pk PRIMARY KEY ( ProductID )

);

CREATE TABLE Cart (

CartID INT NOT NULL,

ProductID INT NOT NULL,

UserID INT NOT NULL,

Qty INT NOT NULL,

CONSTRAINT CartID\_pk PRIMARY KEY ( CartID, ProductID ),

CONSTRAINT fk\_Productid

FOREIGN KEY (ProductID)

REFERENCES Product(ProductID)

ON delete cascade ,

CONSTRAINT fk\_Userid

FOREIGN KEY (UserID)

REFERENCES Shopper(UserID)

ON delete cascade

);

CREATE TABLE Review (

ReviewID INT NOT NULL,

ProductID INT NOT NULL,

UserID INT NOT NULL,

Rating INT NOT NULL,

ReviewDescription Varchar2(1000),

CONSTRAINT ReviewID\_pk PRIMARY KEY ( ReviewID ),

CONSTRAINT fk\_ReviewProductid

FOREIGN KEY (ProductID)

REFERENCES Product(ProductID)

ON delete cascade ,

CONSTRAINT fk\_ReviewUserid

FOREIGN KEY (UserID)

REFERENCES Shopper(UserID)

ON delete cascade

);

CREATE TABLE ProductORDER(

ORDERID INT NOT NULL,

PRODUCTID INT NOT NULL,

QTY INT NOT NULL,

CONSTRAINT ORDERID\_PK PRIMARY KEY(ORDERID, PRODUCTID),

CONSTRAINT FK\_PRODUCTID\_FINALORDER

FOREIGN KEY(PRODUCTID)

REFERENCES PRODUCT(PRODUCTID)

ON delete cascade

);

CREATE TABLE FinalORDER(

ORDERID INT NOT NULL,

USERID INT NOT NULL,

SH\_STREET VARCHAR2(50) NOT NULL,

SH\_CITY VARCHAR2(50) NOT NULL,

SH\_STATE VARCHAR2(50) NOT NULL,

SH\_ZIP INT NOT NULL,

SH\_PHONE VARCHAR2(50) NOT NULL,

Tax FLOAT NOT NULL,

Subtotal FLOAT NOT NULL,

Orderstatus VARCHAR(100),

Orderdate DATE NOT NULL,

LastModifiedDate DATE,

Expected\_arrival DATE,

CONSTRAINT FinalORDERID\_PK

PRIMARY KEY(ORDERID),

CONSTRAINT FK\_USERID\_FINALORDER

FOREIGN KEY(USERID)

REFERENCES SHOPPER(USERID)

ON delete cascade

);

CREATE TABLE PAYMENT(

PAYMENTID INT NOT NULL,

ORDERID INT NOT NULL,

P\_CARDNUMBER VARCHAR2(50) NOT NULL,

P\_NAME\_ON\_CARD VARCHAR2(50) NOT NULL,

P\_CARDTYPE VARCHAR2(50) NOT NULL,

P\_EXPIRYDATE DATE NOT NULL,

P\_CVV INT NOT NULL,

CONSTRAINT PAYMENTID\_PK

PRIMARY KEY(PAYMENTID),

CONSTRAINT FK\_ORDERID\_PAYMENT

FOREIGN KEY (ORDERID)

REFERENCES FINALORDER(ORDERID)

ON delete cascade

);

## **INSERTION into tables**

---------------------------------INSERT statements--------------------

----------------------------------Shopper Table-----------------------

INSERT INTO Shopper

VALUES (

USERID\_SEQ.NEXTVAL,

'Rachel',

'Green',

'rachelgreen@gmail.com',

'rachel',

'404123456',

'1654 Terell Mill Road',

'Atlanta',

'GA',

30039,

'4567435212348765',

'234',

TO\_DATE('01/10/2025','mm/dd/yyyy'));

INSERT INTO Shopper

VALUES (

USERID\_SEQ.NEXTVAL,

'Monica',

'Geller',

'monicageller@gmail.com',

'monica',

'4041234678',

'1654 Akers Mill Road',

'Atlanta',

'GA',

30037,

'987678904321',

'654',

TO\_DATE('02/10/2025','mm/dd/yyyy'));

INSERT INTO Shopper

VALUES (

USERID\_SEQ.NEXTVAL,

'Phoebe',

'Buffay',

'phoebebuffay@gmail.com',

'phoebe',

'4041237896',

'1634 Bridge Mill Dr',

'Atlanta',

'GA',

30039,

'1234765498762345',

'879',

TO\_DATE('03/10/2025','mm/dd/yyyy'));

INSERT INTO Shopper

VALUES (

USERID\_SEQ.NEXTVAL,

'Ross',

'Geller',

'rossgeller@gmail.com',

'ross',

'4047868897',

'1634 Lenox Rd',

'Atlanta',

'GA',

30023,

'3454765789563345',

'385',

TO\_DATE('04/10/2025','mm/dd/yyyy'));

INSERT INTO Shopper

VALUES (

USERID\_SEQ.NEXTVAL,

'Chandler',

'Bing',

'chandlerbing@gmail.com',

'chandler',

'4047869876',

'1634 Lenox Rd',

'Atlanta',

'GA',

30023,

'6576435489097687',

'564',

TO\_DATE('05/10/2025','mm/dd/yyyy'));

INSERT INTO Shopper

VALUES (

USERID\_SEQ.NEXTVAL,

'Joey',

'Tribbiani',

'joeytribbiani@gmail.com',

'joey',

'4046775456',

'1645 Lenox Rd',

'Atlanta',

'GA',

30023,

'4565675687796754',

'274',

TO\_DATE('06/10/2025','mm/dd/yyyy'));

--------------------------------Product table-------------------------

INSERT INTO PRODUCT

VALUES(5001,

'Purina Beyond Natural Dry Dog Food',

'Every dog deserves great nutrition. That’s why Beyond offers grain free solutions you’re looking for. Our Grain Free Beef and Egg Recipe is made with simple, natural ingredients plus vitamins and minerals. Real beef is the first ingredient, a source of protein and essential amino acids. So your dog gets all the nourishment he needs and nothing he doesn’t.',

'Purina Beyond',

32.55,

201,

'https://studentgsu-my.sharepoint.com/personal/vkuttagulveeresh1\_student\_gsu\_edu/\_layouts/15/onedrive.aspx?slrid=02531f9e-f021-4000-7a48-3b8700554158&FolderCTID=0x0120005A4BFE7484588F46A635C211DAE5BE7E&id=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images%2FPurina%2Ejpg&parent=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images'

);

INSERT INTO PRODUCT

VALUES (5002,

'PEDIGREE Complete Nutrition Adult Dry Dog Food',

'PEDIGREE Adult Complete Nutrition Roasted Chicken, Rice & Vegetable Flavor Dry Dog Food is formulated to give dogs all of the energy and nourishment they need to continue living life to the fullest. This chicken-flavor dog food is packed with B vitamins, zinc and omega-6 fatty acid to help keep adult dogs looking their best. Now with no high fructose corn syrup, no artificial flavors, and no added sugar. PEDIGREE Dry Dog Food also contains leading levels of the antioxidant, vitamin E, to help keep adult dogs immune systems strong. Plus, this balanced dog food features our special fiber blend, which helps promote healthy digestion, to keep yard patrol in control. Feed your dog this oral care kibble to help keep teeth clean and give them a great taste.',

'Pedigree',

27.85,

167,

'https://studentgsu-my.sharepoint.com/personal/vkuttagulveeresh1\_student\_gsu\_edu/\_layouts/15/onedrive.aspx?slrid=02531f9e-f021-4000-7a48-3b8700554158&FolderCTID=0x0120005A4BFE7484588F46A635C211DAE5BE7E&id=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images%2FPedigree%2Ejpg&parent=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images'

);

INSERT INTO PRODUCT

VALUES(5003,

'Rachael Ray Nutrish Natural Dry Dog Food',

'Rachael Ray Nutrish Real Chicken & Veggies Recipe is made with simple, natural ingredients, like real U.S. farm-raised chicken, which is always the number one ingredient, combined with wholesome vegetables and added vitamins & minerals. Every ingredient has a purpose. Thats why we carefully select high-quality proteins like chicken, beef and seafood, and then combine them with wholesome veggies and essential vitamins and minerals. A portion of proceeds from each sale of Nutrish is donated to The Rachael Ray Foundation, which helps animals in need through Rachaels Rescue. Through September 2017, Rachaels Rescue has donated more than $21 million to pet charities and other organizations that do good for pets. The funds are used for food, medical supplies, treatments and more for animals in need. When you buy a bag of Nutrish you can feel good knowing that Rachaels personal proceeds from sales of Nutrish go to Rachaels Rescue, providing food, medical supplies and treatment for animals in need.',

'Rachel Ray Nutrish',

41.25,

312,

'https://studentgsu-my.sharepoint.com/personal/vkuttagulveeresh1\_student\_gsu\_edu/\_layouts/15/onedrive.aspx?slrid=02531f9e-f021-4000-7a48-3b8700554158&FolderCTID=0x0120005A4BFE7484588F46A635C211DAE5BE7E&id=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images%2FNutrish%2Ejpg&parent=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images'

);

INSERT INTO PRODUCT

VALUES(5004,

'Natural Ultramix Adult Dry Dog Food',

'At Castor & Pollux Natural PetWorks, we are obsessed with dog and cat nutritional health. For over a decade, we have made innovative and high-quality food and treats for pets that pet parents love feeding. We use only the highest quality ingredients and we test and re-test every food product to ensure its safety. We believe natural is better and no one does natural and organic like we do. Natural Ultramix brand is a natural way to provide complete and essential nutrition for pets. Dry formulas provide natural sources of vitamins, minerals, fiber and antioxidants with real pieces of minimally processed fruits and vegetables for your dog. Only the highest quality proteins, like natural salmon, chicken, duck or beef are used as the #1 ingredient in our formulas. And added omega fatty acides help promote healthy skin and a luxurious coat. There is no wheat, corn, soy or fillers, and never any ingredients from China in any of our recipes.',

'Natural Ultramix',

35.40,

421,

'https://studentgsu-my.sharepoint.com/personal/vkuttagulveeresh1\_student\_gsu\_edu/\_layouts/15/onedrive.aspx?slrid=02531f9e-f021-4000-7a48-3b8700554158&FolderCTID=0x0120005A4BFE7484588F46A635C211DAE5BE7E&id=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images%2FUltramix%2Ejpg&parent=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images'

);

INSERT INTO PRODUCT

VALUES(5005,

'Merrick Grain Free Dry Dog Food',

'The Best Dog Ever deserves the Best Food Ever. That’s why Merrick’s grain-free recipes provide only the best to your best friend in quality, nutrition and taste. Merrick Grain-Free recipes start with real deboned meat, fish or poultry as the #1 ingredient for a high protein, nutritious grain-free meal, along with farm-fresh fruits and vegetables like sweet potatoes, blueberries, peas and more. Dogs thrive on quality protein and healthy fats. That is why Merrick Grain Free Real Duck & Sweet Potato Dry Dog Food provides balanced nutrition with real whole foods sourced from local farmers. This natural, grain-free formula dog food is ideal for all life stages and all breeds of dogs with a focus on naturally nutrient rich ingredients to support overall health. Deboned buffalo is the first ingredient and helps comprise the 70% of meat and fish sourced ingredients that help build and maintain lean, healthy muscle tissue with high quality protein, support healthy skin and coat with omega fatty acids and support joint health with glucosamine.',

'Merrick',

21.35,

305,

'https://studentgsu-my.sharepoint.com/personal/vkuttagulveeresh1\_student\_gsu\_edu/\_layouts/15/onedrive.aspx?slrid=02531f9e-f021-4000-7a48-3b8700554158&FolderCTID=0x0120005A4BFE7484588F46A635C211DAE5BE7E&id=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images%2FMerrick%2Ejpg&parent=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images'

);

INSERT INTO PRODUCT

VALUES(5006,

'Kibbles n Bits Beef Steak Flavor Dry Dog Food',

'As part of our Chefs Choice line of culinary-inspired foods, every bite of Kibbles ‘n Bits American Grill dry dog food tastes like a backyard barbeque! Thoughtfully prepared with grilled beef steak flavor and potatoes, then touched with the taste of real cheddar cheese, this meal is made with high-quality ingredients that offer your dog complete, balanced nutrition. Its a deliciously satisfying meal you can feel good about serving your dog every day',

'Kibbles n Bits',

12.45,

267,

'https://studentgsu-my.sharepoint.com/personal/vkuttagulveeresh1\_student\_gsu\_edu/\_layouts/15/onedrive.aspx?slrid=d65c1f9e-b0f9-4000-7a48-3cad15de94b2&FolderCTID=0x0120005A4BFE7484588F46A635C211DAE5BE7E&id=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images%2FKibbles%20n%20Bits%2Ejpg&parent=%2Fpersonal%2Fvkuttagulveeresh1\_student\_gsu\_edu%2FDocuments%2FDBMS%20Project%20Images'

);

select \* from Product

---------------------------------Review Table-------------------------

INSERT INTO Review

VALUES (

REVIEWID\_SEQ.NEXTVAL,

5001,

1001,

5,

'Outstanding Product ! '

);

INSERT INTO Review

VALUES (

REVIEWID\_SEQ.NEXTVAL,

5001,

1003,

5,

'A must have for all Dogs ! '

);

INSERT INTO Review

VALUES (

REVIEWID\_SEQ.NEXTVAL,

5003,

1005,

3,

'Palatable, but not nutritious. '

);

INSERT INTO Review

VALUES (

REVIEWID\_SEQ.NEXTVAL,

5004,

1002,

5,

'My dog loves it ! '

);

------------------------------SELECT STATEMENTS-----------------------

SELECT \* from shopper;

SELECT \* from product;

SELECT \* from review;

## **DELETION of sequences, tables**

-----------------------------drop table statements--------------------

drop table PAYMENT;

drop table FinalORDER;

drop table ProductORDER;

drop table Review;

drop table Cart;

drop table Product;

drop table Shopper;

--------------------------drop sequences statements-------------------

drop sequence userid\_seq;

drop sequence finalorderid\_seq;

drop sequence cartid\_seq;

drop sequence reviewid\_seq;

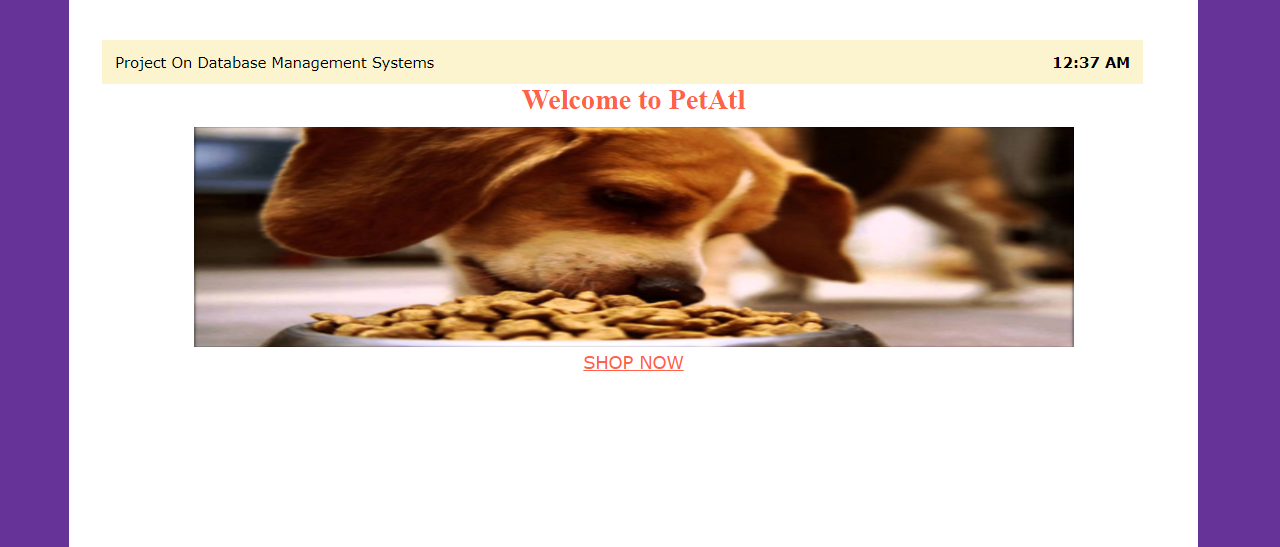
drop sequence payment\_seq

## **Web application & SQL BASED APPLICATION desigN**

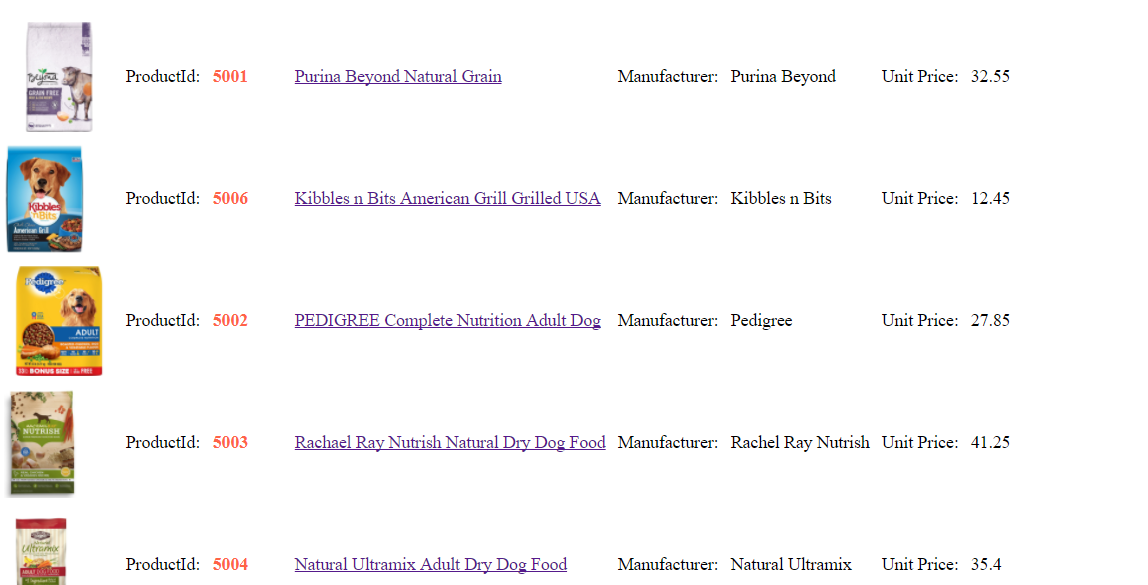
## **Graphical USer interface**

Welcome page

* File name - main\_new.cfm



1. When u click n **SHOP NOW** link it will show all the products
   1. File name - product.cfm
   2. **SQL Query** on page - select productid , pname , description, supplier , price ,image from product
   3. Screen shot of page –



1. If u click on name of a page , you can get the product details and description

**SQL Query on page**

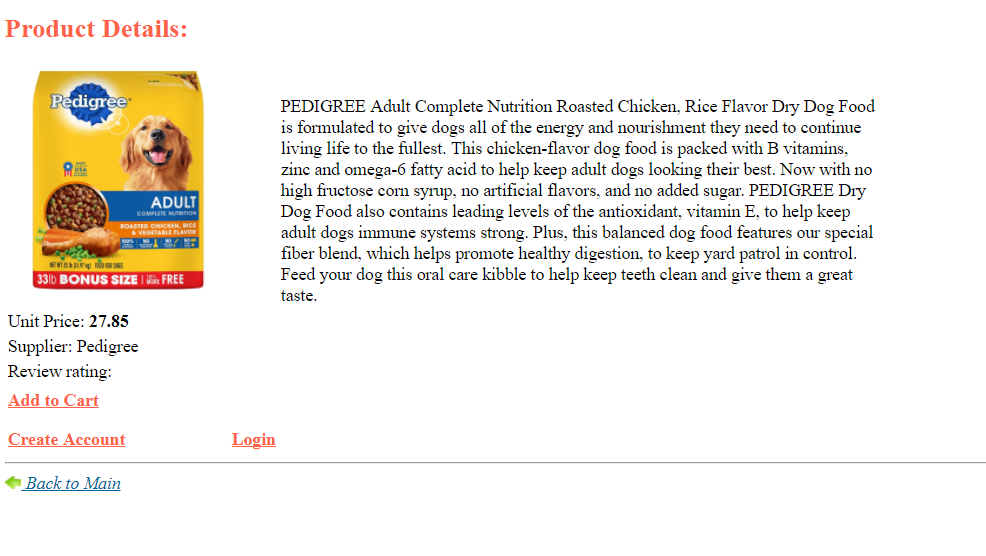
select productid , pname , description, supplier , price ,image from product

where productid = #url.id# and price = #url.price#

select avg(rating) as avgrating from review

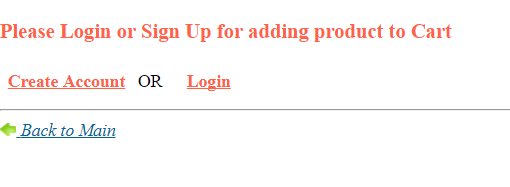
where productid = #url.id#

file name - Product\_details.cfm



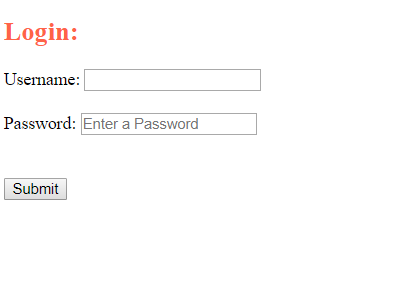
1. If you click on link **Add to Cart** , it will ask you to login or create account following screen will appear

File name - Cart\_insert.cfm



1. If login is successful, following screen will appear for login –

File name - login.cfm



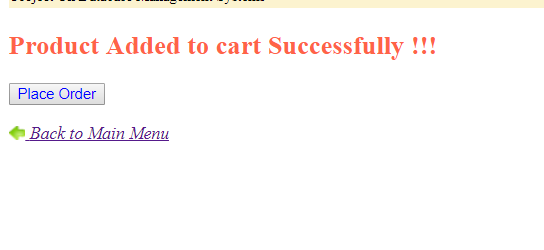
1. After clicking login, it will check in user table if username and password are correct if its valid credentials following screen will appear

File name – Product\_Cart\_insert.cfm

SQL Queries –

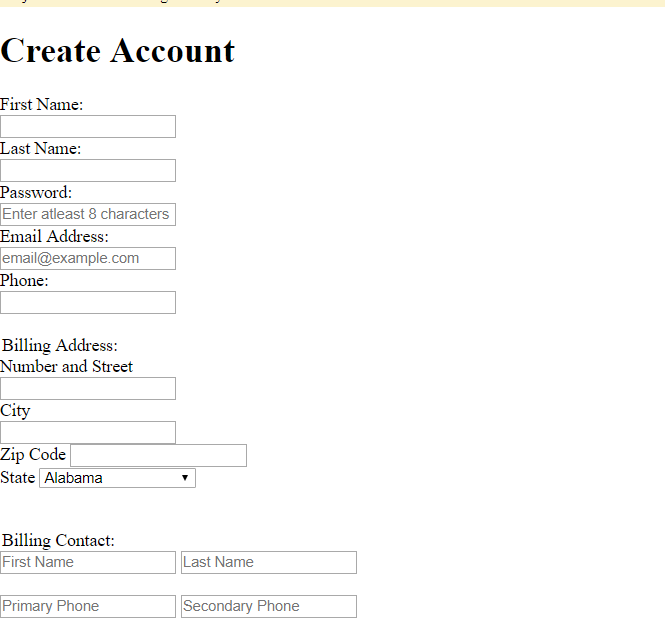
select userid from shopper where fname = '#Form.username#' and password = '#Form.password#'

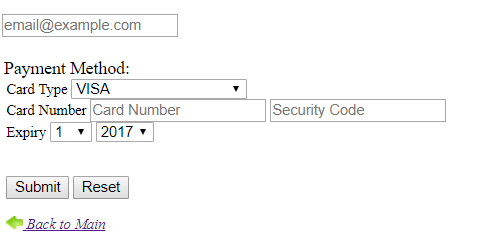
INSERT INTO cart VALUES (cartid\_seq.NEXTVAL,#url.id#,#userid#,1)



1. If you are first time user, Click on create account then following screen will appear

File name - adduser.cfm





1. After creating account successfully following screen will appear it will add product to cart

File name – adduser.cfm

**SQL Query**

* INSERT INTO Shopper

VALUES (

USERID\_SEQ.NEXTVAL,

'#Form.fname#',

'#Form.lname#',

'#Form.email#',

'#Form.Password#',

'#Form.phone#',

'#Form.street#',

'#Form.city#',

'#Form.state#',

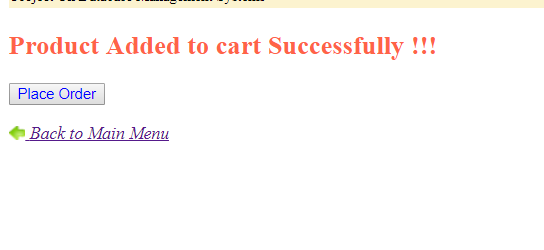
#Form.ZipCode#,

'#Form.CardNumber#',

'#Form.SecurityCode#',

TO\_DATE('#Form.month#/01/#Form.year#','mm/dd/yyyy'))

* select userid from shopper where fname = '#Form.fname#' and password = '#Form.Password#'
* INSERT INTO cart VALUES (cartid\_seq.NEXTVAL,#url.id#,#userid#,1)



1. By clicking on Place order, it will ask order related info to confirm order

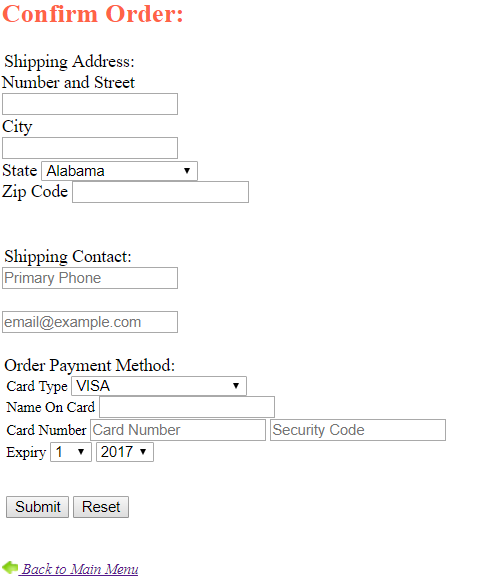
Filename – order.cfm

**Sql Query**

select price, (price\*0.05)\*price as tax from product

where ProductId = #url.id#

Screenshot –



1. After submitting, it will place an order and following entries will be made to data

File name – ordersuccess.cfm

**SQL Query**

1. SELECT price, (price\*0.05)\*price as tax from product

where ProductId = #url.id#

1. INSERT INTO finalorder

VALUES (

finalorderid\_seq.NEXTVAL,

#url.userid#,

'#Form.street#',

'#Form.city#',

'#Form.state#',

#Form.ZipCode#,

'#Form.pphone#',

#url.tax#,

#url.price#,

'InProgress',

sysdate,

sysdate,

sysdate+10)

3. insert into productorder values(finalorderid\_seq.CURRVAL,#url.id#,1)

4. insert into payment values (payment\_seq.NEXTVAL,finalorderid\_seq.CURRVAL,

'#Form.CardNumber#',

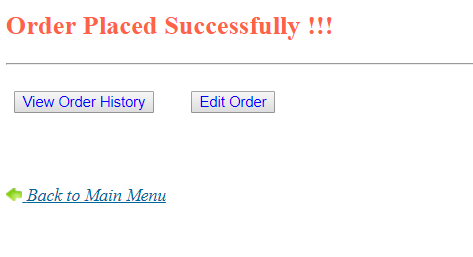
'#Form.CardName#',

'#Form.CardType#',

TO\_DATE('#Form.month#/01/#Form.year#','mm/dd/yyyy'),

#Form.SecurityCode#)

5. delete from cart where userid = #url.userid#



1. You can click on View Order History To View Order

File name – Order\_history.cfm

SQL –

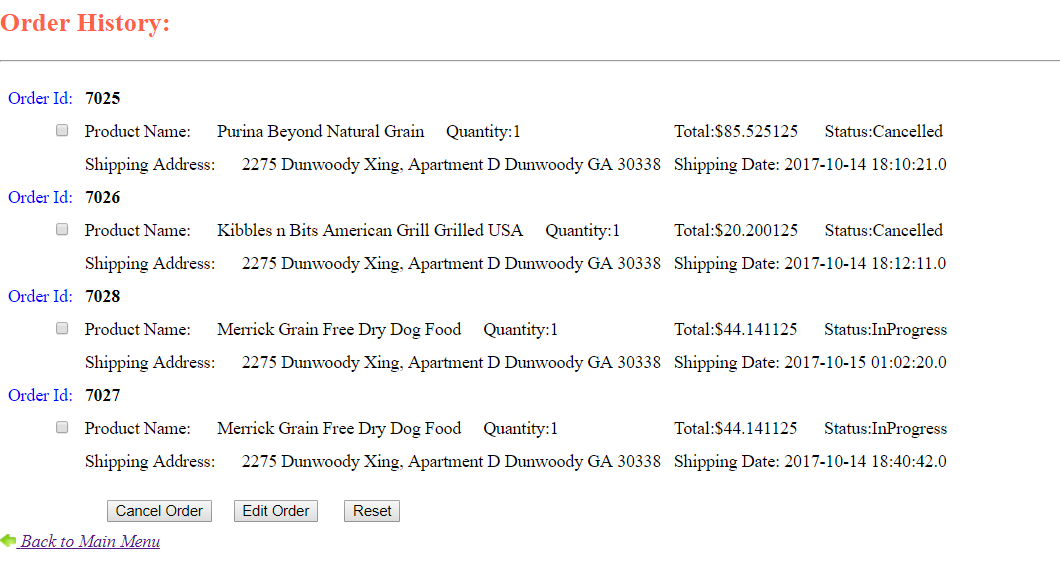
select fo.orderid as orderid, fo.sh\_street as street, fo.sh\_city as city, fo.sh\_state as state,

fo.sh\_zip as zip , p.pname as pname,po.qty as qty , (fo.tax+fo.subtotal) as total , fo.orderstatus as status ,

fo.expected\_arrival as ShippingDate from

finalorder fo, product p , productorder po

where fo.orderid = po.orderid and po.productid = p.productid and userid = #url.userid#

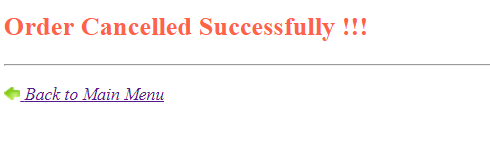


1. You can select order by clicking on checkbox to cancel it and order will get cancelled.

File name – order\_cancel.cfm

Sql –

UPDATE finalorder set orderstatus = 'Cancelled' where orderid = #Form.orderid#



1. **You can go back to main menu and keep shopping.**

## **USer Requirement analysis**

URA is submitted as a different document. Please see the attachments.