E-Commerce Management System

18CSC303J - Database Management Systems

Submitted by

Sai Rohit Paturi (RA2011026010200)
Koushiki Ray (RA2011026010199)
Cathy Anand (RA2011026010216)

Submitted to

Dr. S Joseph James

Assistant Professor, Department of Computational Intelligence

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING WITH SPECIALIZATION IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTING
DEPARTMENT OF COMPUTATIONAL INTELLIGENCE
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203

May 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report "E-Commerce Management System" is the bonafide work of Sai Rohit Paturi (RA2011026010200), Koushiki Ray (RA2011026010199) and Cathy Anand (RA2011026010216), who carried out the project work under my supervision.

SIGNATURE

Dr.S Joseph James

Assistant Professor

Department of Computational Intelligence

SRM Institute of Science and Technology

SIGNATURE

Dr. R.Annie Uthra

HEAD OF THE DEPARTMENT

Professor & Head

Department of Computational Intelligence

SRM Institute of Science and Technology

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honourable Vice Chancellor Dr. C. MUTHAMIZHCHELVAN, for being the beacon in all our endeavours.

We would like to express my warmth of gratitude to our Registrar Dr. S. Ponnusamy, for his encouragement

We express our profound gratitude to our Dean (College of Engineering and Technology) Dr. T. V. Gopal, for bringing out novelty in all executions.

We would like to express our heartfelt thanks to the Chairperson, School of Computing Dr. Revathi Venkataraman, for imparting confidence to complete my course project

We are highly thankful to our Course project Faculty **Dr. S Joseph James, Assistant Professor, Department of Computational Intelligence, for his/herassistance, timely suggestions and guidance throughout the duration of this course project.**

We extend my gratitude to our HoD, Dr. R. Annie Uthra, Professor & Head, Department of Computational Intelligence and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course presentation.

Table of Contents

Abstract

In this modern era of online shopping, no seller wants to be left behind. Moreover, due to its simplicity, the shift from offline selling model to an online selling model is witnessing a rampant growth. Therefore, as an engineer, our job is to ease the path of this transition from the seller. An online site requires a suitable database for it to be able to store the items and their values in order to be presented to the user.

Problem Definition

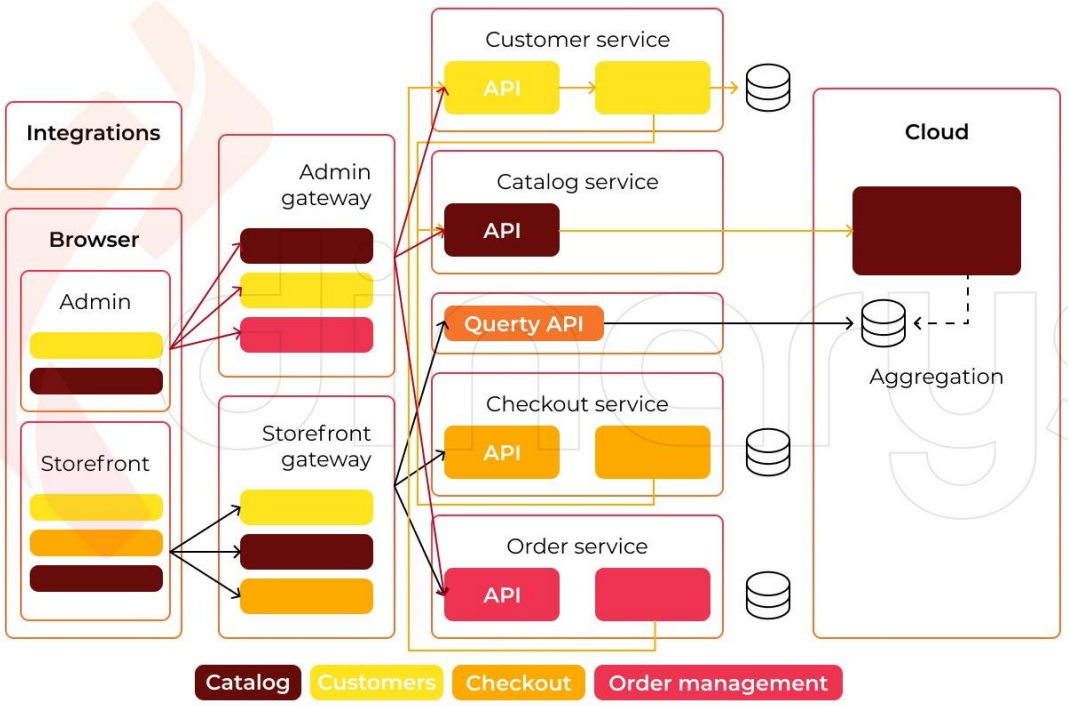
The problem that this project aims to address is the need for a user-friendly, efficient, and effective app that enables small business owners to sell products online as well as enable customers to buy them.

This project seeks to behave like typical online sale applications. However, this is project aims to enable small business sellers to be able to sell their products without having to care too much about how much the application requires of them. The idea is to surpass apps by being user-friendly, efficient, and effective in meeting the needs of consumers as well as making sure that sellers are able to advertise and sell their products.

Objectives:

1. To create a well accessible login page catering both to users and sellers with different types of authentications. It will also be able to smoothly transition to the respective pages as required to be viewed by the user and sellers.
2. Develop an easily manageable database to save product, seller, customer and cart details for each use.
3. Use the insights gained from the research to design and develop a user-friendly, efficient, and effective online e-commerce application that enables simple and effective online sales page.

System Architecture



UI: The front-end of the application is developed using a Python GUI. The back-end of the application is developed using Oracle SQL Server. The connectivity between the two is done through Oracle.

Database: The database used for this project is based in Oracle.

User Authentication: The project includes a user authentication system that allows users to register, login, and access the home page only when authenticated.

Customer functions: The application allows authenticated users to add new items to cart, view their details, filter products by product details and save their cart for future purchases.

Seller/Admin functions: The application allows authenticated users to add new products to their catalog, update product details, them, set up price changes and discounts.

User Feedback: The project includes a mechanism for collecting user feedback and conducting user acceptance testing to evaluate the effectiveness and user-friendliness of the application.

Features:

User/Admin Authentication: The application includes a user/admin authentication system that allows users to register, login, and access the home page only when authenticated. Similarly, it allows for admins/sellers to login and access their product pages.

Sale and Product pages: The application enables users to view product details from specific sellers, filter products by detail specification and update their cart. On the other end, it enables sellers to update their product catalog with new products or remove old ones. It also allows them to them to set up product discounts.

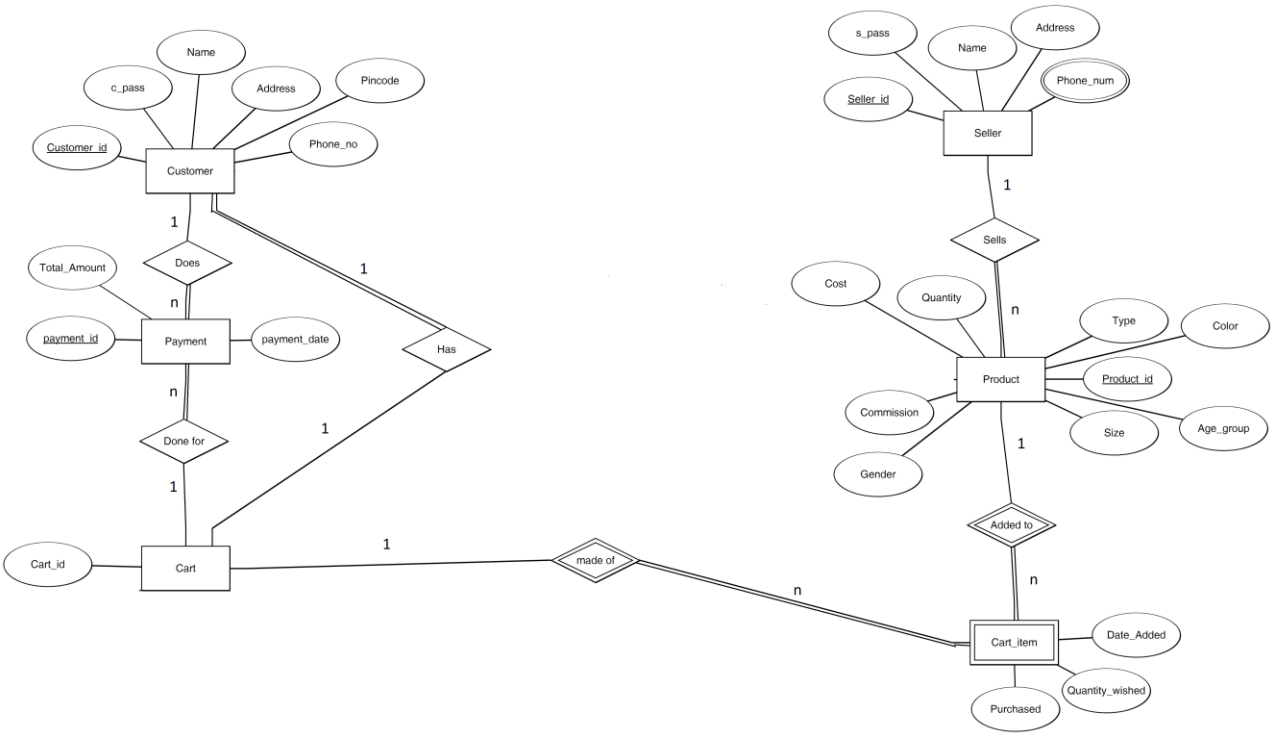
Responsive Design: The application is designed to be responsive, which means that it can be accessed and used on multiple devices, including desktop computers, laptops, tablets, and smartphones.

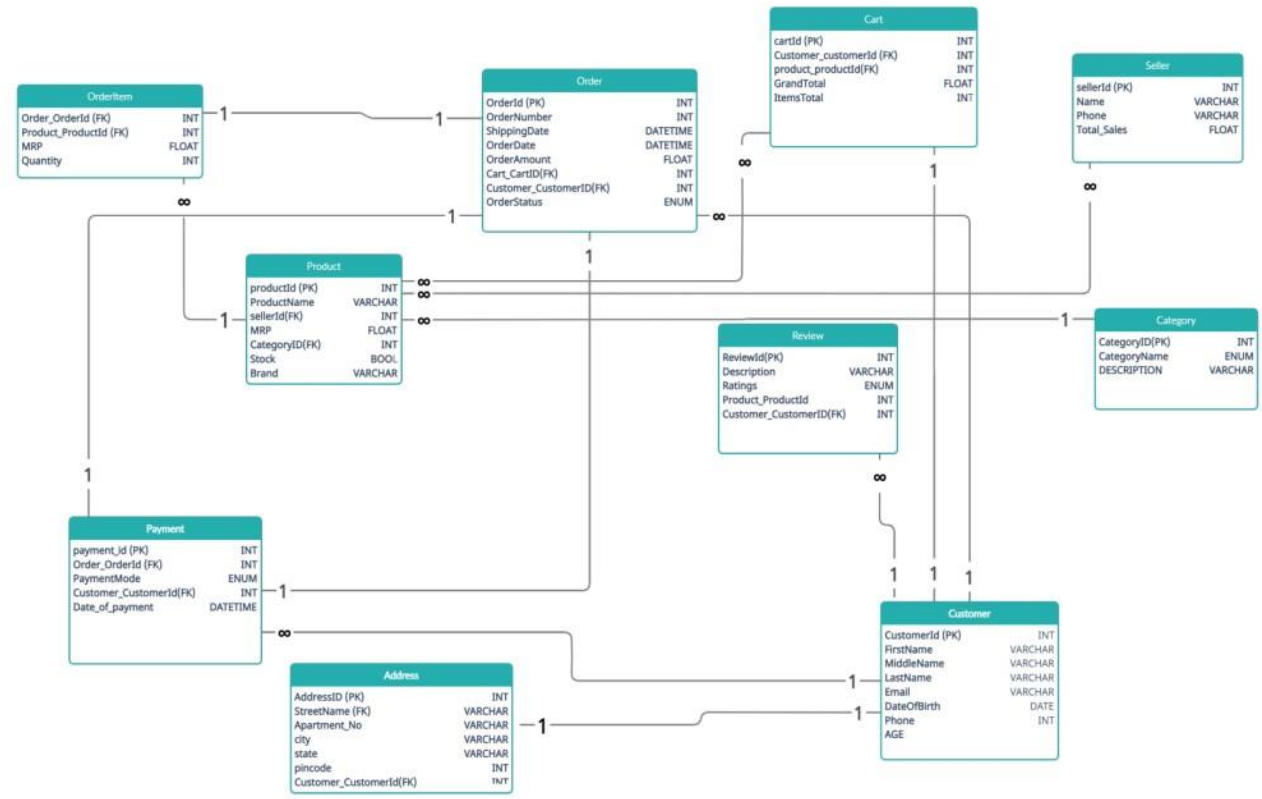
User Feedback: The project includes a mechanism for collecting user feedback and conducting user acceptance testing to evaluate the effectiveness and user-friendliness of the application.

Database Management: The application uses Oracle SQL Server as the database management system, which is easy to use and maintain. It is integrated to the frontend by utilizing Oracle services.

Code Documentation: The project is well-documented with comments and docstrings, making it easy for other developers to understand and modify the code.

Database Schema and ER Diagram





User Interface

Screen Layout:

The Customer Database Management System features a clean and simple screen layout, with a navigation bar at the top of the screen and a search bar in the header. The main section of the screen includes a table displaying all customer details, with options to add, edit, and delete customer records.

Input Forms:

The add and edit customer forms include fields for Name, Email, Address, Phone Number, and Purchase History. Each field is validated to ensure that the input is in the correct format, and error messages are displayed if required fields are left blank or invalid data is entered.

The page for sellers will also include fields for Name, contact details and Catalog. Additionally, they can enter a page to change catalog features and product details within.

Functional requirements

- A new user can register on the website.
- A customer can see details of the product present in the cart
- A customer can view his order history.
- Admin can start a sale with certain discount on every product.
- Customer can filter the product based on the product details.
- A customer can add or delete a product from the cart.
- A seller can unregister/ stop selling his product.
- A seller/ customer can update his details.
- Admin can view the products purchased on particular date.
- Admin can view number of products sold on a particular date.
- A customer can view the total price of product present in the cart unpurchased.
- Admin can view details of customer who have not purchased anything.
- Admin can view total profit earned from the website.

Code:

1. Creating Tables

```
CREATE TABLE Cart
(
    Cart_id VARCHAR(7) NOT NULL,
    PRIMARY KEY(Cart_id)
);

CREATE TABLE Customer
(
    Customer_id VARCHAR(6) NOT NULL,
    c_pass VARCHAR(10) NOT NULL,
    Name VARCHAR(20) NOT NULL,
    Address VARCHAR(20) NOT NULL,
    Pincode NUMBER(6) NOT NULL,
    Phone_number_s number(10) NOT NULL,
    PRIMARY KEY (Customer_id),
    Cart_id VARCHAR(7) NOT NULL,
    FOREIGN KEY(Cart_id) REFERENCES cart(Cart_id)
);

CREATE TABLE Seller
(
    Seller_id VARCHAR(6) NOT NULL,
    s_pass VARCHAR(10) NOT NULL,
    Name VARCHAR(20) NOT NULL,
    Address VARCHAR(10) NOT NULL,
    PRIMARY KEY (Seller_id)
);

CREATE TABLE Seller_Phone_num
(
    Phone_num NUMBER(10) NOT NULL,
    Seller_id VARCHAR(6) NOT NULL,
    PRIMARY KEY (Phone_num, Seller_id),
    FOREIGN KEY (Seller_id) REFERENCES Seller(Seller_id)
    ON DELETE CASCADE
);

CREATE TABLE Payment
(
    payment_id VARCHAR(7) NOT NULL,
    payment_date DATE NOT NULL,
    Payment_type VARCHAR(10) NOT NULL,
    Customer_id VARCHAR(6) NOT NULL,
    Cart_id VARCHAR(7) NOT NULL,
    PRIMARY KEY (payment_id),
    FOREIGN KEY (Customer_id) REFERENCES Customer(Customer_id),
    FOREIGN KEY (Cart_id) REFERENCES Cart(Cart_id),
    total_amount numeric(6)
);
```

```
CREATE TABLE Product
(
    Product_id VARCHAR(7) NOT NULL,
    Type VARCHAR(7) NOT NULL,
    Color VARCHAR(15) NOT NULL,
    P_Size VARCHAR(2) NOT NULL,
    Gender CHAR(1) NOT NULL,
    Commission NUMBER(2) NOT NULL,
    Cost NUMBER(5) NOT NULL,
    Quantity NUMBER(2) NOT NULL,
    Seller_id VARCHAR(6),
    PRIMARY KEY (Product_id),
    FOREIGN KEY (Seller_id) REFERENCES Seller(Seller_id)
    ON DELETE SET NULL
);

CREATE TABLE Cart_item
(
    Quantity_wished NUMBER(1) NOT NULL,
    Date_Added DATE NOT NULL,
    Cart_id VARCHAR(7) NOT NULL,
    Product_id VARCHAR(7) NOT NULL,
    FOREIGN KEY (Cart_id) REFERENCES Cart(Cart_id),
    FOREIGN KEY (Product_id) REFERENCES Product(Product_id),
    Primary key(Cart_id,Product_id)
);

alter table Cart_item add purchased varchar(3) default 'NO';
```

2. Inserting Values (Demo Values)

```
insert into Cart values('crt1011');

    insert into Customer values('cid100','ABCM1235','rajat','G-
453','632014',9893135876, 'crt1011');

    insert into Seller values('sid100','12345','aman','delhi cmc');

    insert into Product
values('pid1001','jeans','red','32','M',10,10005,20,'sid100');

    insert into Seller_Phone_num values('9943336206','sid100');

    insert into Cart_item values(3,to_date('10-OCT-1999','dd-mon-
yyyy'),'crt1011','pid1001','Y');

    insert into Payment values('pmt1001',to_date('10-OCT-1999','dd-mon-
yyyy'),'online','cid100','crt1011',NULL);
```

3. Basic Queries

If a customer wants to see details of the product present in the cart

```
select * from product where product_id in(
     select product_id from Cart_item where (Cart_id in (
         select Cart_id from Customer where Customer_id='cid100'
     ))
 and purchased='NO');
```

If the customer wants to see order history

```
   select product_id,Quantity_wished from Cart_item where (purchased='Y' and Cart_id
in (select Cart_id from customer where Customer_id='cid101'));
```

Customer wants to see filtered products on the basis of size, gender, type

```
   select product_id, color, cost, seller_id from product where (type='jeans' and
_size='32' and gender='F' and quantity>0)
```

Admin wanys to see what product was purchased on a particular date

```
   select product_id from cart_item where (purchased='Y' and date_added='12-dec-
2018');
```

How many products were sold on a particular date

```
   select count(product_id) count_pid,date_added from Cart_item where purchased='Y'
group by(date_added);
```

If customer wants to know the total price of the cart

```
   select sum(quantity_wished * cost) total_payable from product p join cart_item c on
.product_id=c.product_id where c.product_id in (select product_id from cart_item where
art_id in(select Cart_id from customer where customer_id='cid101') and purchased='Y');
```

Show details of customer who has not purchased anything

```
   Select * from customer where customer_id not in (select customer_id from Payment);
```

Total profit from sales

```
   select sum(quantity_wished * cost * commission/100) total_profit from product p
join cart_item c on p.product_id=c.product_id where purchased='Y'
```

4. PL/SQL Functions

Procedure returns the types of product with the cost less than the given cost

```
  create or replace procedure cost_filter(c in number,t in varchar)
    is
    cs product.cost%type;
    ty product.type%type;
    id product.product_id%type;
    cursor cf is
    select product_id,cost,type from product where cost<c and type=t;
    begin
    open cf;
    loop
    fetch cf into id,cs,ty;
    exit when cf%notfound;
    dbms_output.put_line('Product' || id || 'has cost ' || cs || ' and the type is' ||
ty);
    end loop;
    close cf;
    exception
    when no_data_found then
    dbms_output.put_line('Sorry no such products exist');
    end;
```

Function returns the total number of products which a particular seller sells

```
create or replace function totalProducts(sId in varchar)
    return number
    is
    total number(2):=0;
    begin
    select count(*) into total
    from product
    where seller_id=sId;
    return total;
    end;
    /
```

Function execution

```
create or replace procedure prod_details(p_id in varchar)
    is
    quan number(2);
    begin
    select quantity into quan from product where product_id=p_id;
    exception
    when no_data_found then
    dbms_output.put_line('Sorry no such product exist !!');
    end;
    /
```

5. Triggers

To count number of cart items

```
create or replace function numCartId(cd in varchar)
return number
is
total number(2):=0;
begin
select count(*) into total
from cart_item
where cart_id=cd;
return total;
end;
Trigger
Create or replace trigger before_customer
before insert
on
customer
for each row
declare
c varchar(10);
n number(2);
begin
c:= :new.cart_id;
n:=numCartId(c);
if n>0 then
dbms_output.put_line('Sorry');
end if;
insert into cart values(c);
end;
```

Trigger to update the total amount of user every time he adds something to payment table

```
create or replace function total_cost(cId in varchar)
    return number
    is
    total number(2) :=0;
    begin
    select sum(cost) into total from product,cart_item where
product.product_id=cart_item.product_id and cart_id=cId;
    return total;
    end;

    create or replace trigger before_pay_up
    before insert
    on
    payment
    for each row
    declare
    total number(3);
    begin
    total :=total_cost(:new.cart_id);
    insert into payment
values(:new.payment_id,:new.payment_date,:new.payment_type,:new.customer_id,:new.cart_i
d,total);
    end;
```

Future Prospects:

Machine Learning Integration: The application can be enhanced by incorporating machine learning algorithms that analyze user data to provide personalized recommendations and insights.

Social Media Integration: The application can be integrated with social media platforms, allowing users to get personalized ads for their requirements according to their search preferences. This will also be an opportunity for sellers to increase their chances to publicize their products.

Cloud-based Storage: The application can be enhanced by implementing cloud-based storage, which allows users to access their data from anywhere, anytime, and provides greater security and reliability.

Business Opportunities: The application can be expanded into a business opportunity, where sellers will eventually be able to get better product selling prospects.

Conclusion:

In conclusion, the E-Commerce Management system provides an easy and accessible platform for both small scale sellers and customers. Our project aims to provide an efficientand user-friendly platform for sellers to advertise and sell their products as well as a hub for users to purchase new products.

Our project offers a range of features, including user authentication, viewing purchase details, cart details, updating of product details, applying discounts as well as updating of user/seller details. The application is built using Python GUI for a pleasant access experience for users and the Oracle SQL Server to serve as the database service.

Overall, we believe that our project provides a valuable tool for small scale business owners who want an accessible online platform to advertise and sell their products. By combining technical features, data accuracy, and value for money, we have designed an application that meets the needs and preferences of users, and provides an effective platform for empowering small business and to help them grow.

References:

- https://www.google.com/search?q=er+diagram&rlz=1C1CHBF_enIN914IN914&oq=er+diagram&aqs=chrome..69i57j0i512j0i433i512l3j0i512l2j69i60.2605j0j7&sourceid=chrome&ie=UTF-8
- https://www.dragon1.com/terms/architecture-diagram-definition
- https://www.princeton.edu/~rcurtis/ultradev/ecommdatabase.html