ONLINE RETAIL SHOPPING

A PROJECT REPORT

Submitted by

Raghav Kapoor [Reg No: RA2211056030014]
Harshit Veerwal [Reg No: RA2211056030026]
Harrish Srinivasan [Reg No: RA2211056030029]

Under the guidance of

Dr. Franklin Vinod

In partial fulfilment for the course

of

(21CSC205P) DATABASE MANAGEMENT SYSTEM

SRM INSTITUTE OF SCIENCE & TECHNOLOGY, NCR CAMPUS
APRIL 2024

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified to be the bonafide record of work done by ***Raghav Kapoor RA2211056030014,*** ***Harshit Veerwal RA2211056030026*** *and* ***Harrish Srinivasan RA2211056030029*** *of* 4^{th} *semester* 2^{nd} *year* ***B.TECH*** *degree course in* ***SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, NCR Campus*** *of Department of Computer Science & Engineering in Database Management Systems Lab, during the academic year 2023-2024.*

SIGNATURE

Dr. Franklin Vinod
Assistant Professor
Computer Science & Engineering

SIGNATURE

Dr. Avneesh Vashistha
Head of the Department
Computer Science & Engineering

ABSTRACT

An online shopping cart system, built with PHP and SQL, allows Customer to browse products, add them to their cart, and complete purchases. In PHP, user interactions trigger SQL queries to manage the cart's state and inventory. When a user adds an item to the cart, PHP sends an SQL INSERT query to store the product details in a database table associated with the user's session ID. The cart's total cost is dynamically calculated using SQL queries to retrieve prices and quantities. During checkout, PHP processes the payment and updates the order status in the database using SQL UPDATE queries. Additionally, PHP ensures session persistence, maintaining user carts across page reloads, while SQL handles data retrieval and storage securely, guarding against unauthorized access or data loss. Overall, this integration of PHP and SQL enables a robust and secure online shopping experience, seamlessly managing user interactions.

INDEX

INTRODUCTION

A Shopping Management System is a comprehensive software solution designed to streamline the operations and processes involved in running an online store. It serves as the backbone for managing various aspects of shopping businesses, including inventory management, order processing, customer relationship management, marketing, and analytics.

Here's a breakdown of the key components and functionalities typically found in a Shopping Management System:

1. **Product Management**: This feature allows store owners to add, edit, categorize, and organize products effectively. It includes features such as product descriptions, images, pricing, and inventory levels.
2. **Order Processing**: The system automates the order fulfillment process from the point of purchase to delivery, including order confirmation, payment processing, shipping, and tracking.
3. Provide personalized shopping experiences through features like customer accounts, wish lists, targeted promotions, and seamless access to product listings for convenient purchasing.

Scalability and Customization: They are designed to scale with the growth of the business and offer customization options to tailor the platform according to specific business requirements.

PROBLEM IDENTIFICATION

Despite the widespread popularity of online shopping, many existing shopping platforms face various challenges that hinder user satisfaction and hinder business efficiency. Some of the common problems include:

1. Poor User Experience: Many shopping platforms have complex user interfaces, making it difficult for Customer to navigate through product listings, manage their shopping carts, and complete transactions smoothly.

2. Inefficient Database Management: Traditional Shopping platforms often rely on bulky and slow database systems, leading to sluggish performance, data inconsistencies, and difficulties in managing large volumes of transactions and customer data.

3. Security Concerns: Security breaches, data leaks, and unauthorized access to sensitive information are significant concerns for both Customer and businesses. Weak authentication mechanisms and inadequate data encryption can expose Customer' personal and financial data to potential threats.

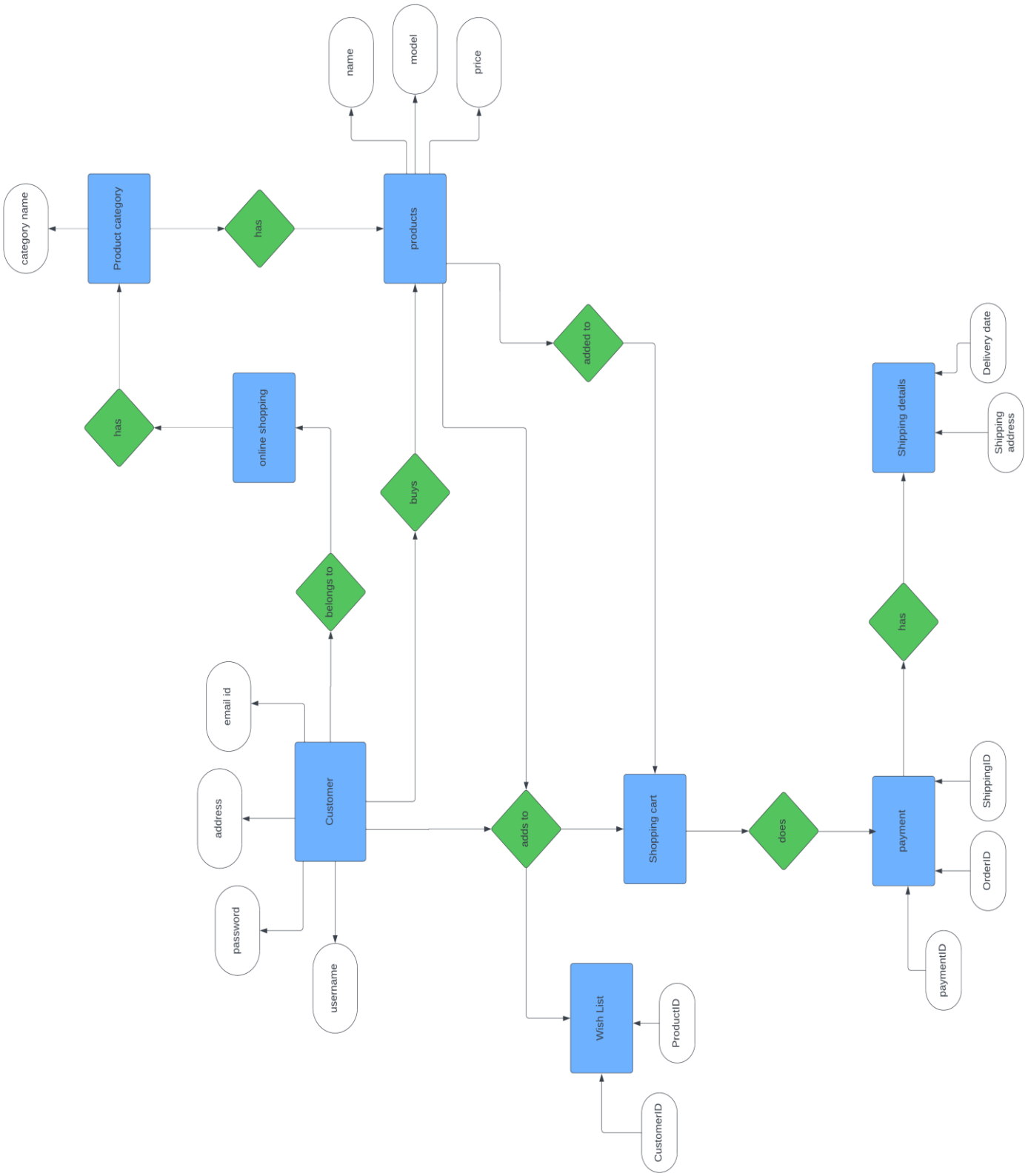
4. Limited Scalability: Some Shopping platforms struggle to handle increasing trafficand transaction volumes, resulting in system crashes, slow response times, and downtime during peak periods such as sales events or holidays.

5. Lack of Customization: Many Shopping platforms offer limited customizationoptions for Customer and administrators, making it challenging to tailor the platform to specific business needs, branding requirements, or user preferences.

6. High Development Costs: Developing and maintaining a custom Shopping platformcan be costly and time-consuming, especially for small businesses or startups with limited resources. High development costs may deter businesses from investing in a robust online storefront.

Addressing these challenges requires the development of a modern and efficient Shopping platform that prioritizes user experience, data security, scalability, customization, and affordability. By leveraging Python, SQLite, and Tkinter, this project aims to tackle these problems and deliver a comprehensive solution that meets the needs of both Customer and businesses in the digital marketplace.

ER DIAGRAM


name
model
price
category name
Product category
has
products
added to
has
online shopping
buys
Shipping details
Delivery date
Shipping address
belongs to
has
email id
address
Customer
adds to
Shopping cart
does
payment
ShippingID
OrderID
password
username
Wish List
ProductID
CustomerID
paymentID

ENTITIES

1. **Shopping cart:**
    - Description: The Shopping cart entity represents a collection of items that a user desires to acquire or keep track of within an shopping website. Customer often create Shopping carts to bookmark products they are interested in purchasing at a latertime or to share with others as gift suggestions.
    - Attributes:
        - product_id: A unique identifier for each Shopping cart, serving as the primarykey for the Shopping cart entity. It distinguishes one Shopping cart from another inthe database.

2. **Customer:**
    - Description: The Customer entity represents individuals who have registered on theshopping website to avail its services, such as purchasing products, creating Shopping carts, or leaving reviews. Each user is identified uniquely within the system and possesses various attributes that help manage their account and facilitate communication.
    - Attributes:
        - Email: The email address associated with the user's account, serving as a unique identifier for login purposes.
        - User_Id: A unique identifier assigned to each user, functioning as the primary key for the Customer entity. It distinguishes one user from anotherin the database.
        - Password: The encrypted string used for user authentication and access control.
        - Phone: A multi-valued attribute representing the contact phone number(s) associated with the user. Customer may have more than one phone number stored in their profile.
        - Name: Name of the user.
        - Address: A composite attribute comprising several sub-attributes that collectively represent the user's physical location. It includes:

- H.no (House Number): The building or house number of the user's address.
- Street: The street name where the user resides.

City: The city or locality where the user's address is located.

- State: The state or province where the user resides.
- Pincode: The postal code or ZIP code associated with the user's area.
- Landmark: A notable point of reference near the user's address, aiding in locating their residence.

3. Products:

- Description: The Products entity represents the various items available for purchase on the shopping website. It includes details such as the product's price, description, and name.
- Attributes:
  - Product_id: A unique identifier assigned to each product, serving as the primary key for the Products entity. It distinguishes one product from another in the database.
  - Price: The cost of the product, indicating the amount Customer need to pay to purchase it.
  - Description: A brief description or overview of the product, providing Customer with information about its features, specifications, and uses.
  - Name: The name or title of the product, serving as a concise identifier for Customer to recognize and refer to the item.

RELATIONSHIPS

1. Customer has Shopping cart:

    - One-to-Many relationship from Customer to Shopping cart (One user can have multipleShopping carts)
    - Total Participation of Shopping cart (Every Shopping cart must be associated with a user)
    - Partial Participation of Customer(A user may or maynot have a Shopping cart)

2. Products added to Shopping cart:

    - Many-to-Many relationship between Products and Shopping cart (A product can beadded to multiple Shopping carts, and a Shopping cart can contain multiple products)
    - Partial Participation of Products(Every product may not be present in a Shopping cart)
    - Partial Participation of Shopping cart (A Shopping cart can be empty)

3. Customer place Products:

    - One-to-Many relationship from Customer to Products (One user can place multipleProducts)
    - Total Participation of Products(Every order must be associated with an user)
    - Partial Participation of Customer(A user may or maynot place an order)

ER DIAGRAM TO RELATIONAL TABLE

CUSTOMER

USER_id (PK)	Username	Password	Mobile_no	Email	Address
101	User1	Abcd	985481290	123@example.com	Noida
102	User2	Asdf	1639820560	gsfd@example.com	Delhi
103	User3	Sefr241	2136547892	safgr@example.com	Ghaziabad
104	User4	23fer	2365479324	rgffg@example.com	Meerut
105	User5	12334e	5486123657	jaso@example.com	Delhi

PRODUCT CATEGORY

category_id (PK)	Category_name
M101	Mobiles
C101	Computers
A101	Appliances
M201	Men’s fashion
M202	Women’s fashion
K301	Kitchen
B401	Beauty
H501	Health
S601	Sports
M701	Movies
V801	Videogames

PRODUCT

Product_id (PK)	Product_name	Category ID (FK)	price
S101	Smartwatch	M101	5,000
T102	Tshirt	M201	1,200
L103	Light bulb	A101	500
E104	Earphones	M101	2,500
L201	Laptop	C101	75,000

WISH LIST

Cart_item (PK)	Customer_ID	Product_ID (FK)
iPhone	101	I101
Cupboard	102	C102
Facewash	101	F103
Projector	104	P104

SHOPPING CART

Cart_item (PK)	Customer_ID	Product_ID (FK)
Smartwatch	101	S101
Earphones	103	E104
Light bulb	101	L103

PAYMENT

Payment_ID	Shopping_ID	Order_ID (FK)	Payment_method
P101	1001	1234567890123456	Credit card
P102	1002	9876543210987654	Netbanking
P103	1003	357924680135792	Credit card
P104	1004	2468013579246801	Cash on delivery
P105	1005	1235468434846188	Cash on delivery
P106	1006	23569197397629731	Credit card
P107	1007	94319976197329732	Netbanking
P108	1008	69845896858626848	Credit card

SHOPPING DETAILS

Shopping_ID	Product_ID	Order_ID	Shipping_address	Delivery_date
1002	S101	9876543210987654	123 Main St	2024-04-30
1003	E104	357924680135792	453 Main St	2024-05-31

MY SQL QUERIES

```
CREATE DATABASE onlineretail;

USE onlineretail;

CREATE TABLE customer(
customer_id INT PRIMARY KEY AUTO_INCREMENT,
customer_name VARCHAR(20),
emailID VARCHAR(20)
);

CREATE TABLE product_category(
category_id VARCHAR(50) PRIMARY KEY,
category_name VARCHAR(20)
);

CREATE TABLE products(
product_id VARCHAR(10) PRIMARY KEY,
category_id VARCHAR(50),
product_name VARCHAR(20),
price DECIMAL (10,2) NOT NULL,
FOREIGN KEY (category_id) REFERENCES product_category(category_id)
);

CREATE TABLE wishlist(
customer_id INT,
product_id VARCHAR(10),
PRIMARY KEY (customer_id, product_id),
FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
FOREIGN KEY (product_id) REFERENCES products(product_id)
);

CREATE TABLE shoppingcart(
customer_id INT,
product_id VARCHAR(10),
quantity INT NOT NULL,
PRIMARY KEY (customer_id, product_id),
FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
FOREIGN KEY (product_id) REFERENCES products(product_id)
);

CREATE TABLE payment(
payment_id INT PRIMARY KEY AUTO_INCREMENT,
customer_id INT,
```

```
payment_method VARCHAR(50),
card_number FLOAT NOT NULL,
FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);

CREATE TABLE shippingdetails(
shipping_id INT PRIMARY KEY AUTO_INCREMENT,
customer_id INT,
address VARCHAR(50) UNIQUE,
city VARCHAR(10),
FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);


INSERT INTO customer
VALUES
(101,"Olivia","oliv12@gmail.com"),
(102,"Eren","eren23@outlook.com"),
(103,"Rahul","rbansal@gmail.com"),
(104,"Cain","cc1234@gmail.com"),
(105,"Lucy","lucy45@gmail.com");

INSERT INTO product_category
VALUES
("M101", "Mobiles"),
("C101", "Computers"),
("A101", "Appliances"),
("M201", "Men's Fashion"),
("M202", "Women's Fashion"),
("K301", "Kitchen"),
("B401", "Beauty"),
("H501", "Health"),
("S601", "Sports"),
("M701", "Movies"),
("V801", "Videogames");

INSERT INTO products
VALUES
('P001', 'M101', 'Smartphone', 499.99),
('P002', 'C101', 'Laptop', 899.99),
('P003', 'M701', 'Television', 699.99),
('P004', 'A101', 'Washing Machine', 399.99),
('P005', 'M202', 'Shirt', 29.99),
('P006', 'M202', 'Dress', 49.99),
('P007', 'M202', 'T-shirt', 19.99),
('P008', 'M201', 'Jeans', 39.99),
('P009', 'A101', 'Microwave Oven', 149.99),
('P010', 'K301', 'Blender', 59.99);

-- Inserting data into the wishlist table
```

```
INSERT INTO wishlist (customer_id, product_id)
VALUES
(101, 'P001'),
(101, 'P003'),
(102, 'P002'),
(103, 'P005'),
(104, 'P004'),
(105, 'P007');

-- Inserting data into the shoppingcart table
INSERT INTO shoppingcart (customer_id, product_id, quantity)
VALUES
(101, 'P001', 2),
(101, 'P003', 3),
(102, 'P002', 1),
(103, 'P005', 1),
(104, 'P004', 2),
(105, 'P007', 1);

-- Inserting data into the payment table
INSERT INTO payment (customer_id, payment_method, card_number)
VALUES
(101, 'Netbanking', 1234623464),
(102, 'Creditcard', 8573901846),
(103, 'Creditcard', 6481034867),
(104, 'Netbanking', 6482048574),
(105, 'Creditcard', 2385028572),
(101, 'Debitcard', 23849158624),
(102, 'Debitcard', 16584981354),
(103, 'Netbanking', 1687486235),
(104, 'Netbanking', 4567456125),
(105, 'Creditcard', 7914952438);

-- Inserting data into the shippingdetails table
INSERT INTO shippingdetails (customer_id, address, city)
VALUES
(101, "123 Main St", "Noida"),
(102, "456 Elm St", "Delhi"),
(103, "789 Oak St", "Ghaziabad"),
(104, "101 Pine St", "Ghaziabad"),
(105, "202 Maple St", "Noida");

SELECT * FROM customer;

SELECT * FROM product_category;

SELECT * FROM products;

SELECT * FROM shippingdetails;
```

```
SELECT * FROM wishlist;

SELECT * FROM shoppingcart;

SELECT * FROM payment;

-- Performing JOIN operations
-- INNER JOIN between customer and wishlist
SELECT c.customer_name, w.product_id
FROM customer c
INNER JOIN wishlist w ON c.customer_id = w.customer_id;

-- LEFT JOIN between customer and shoppingcart
SELECT c.customer_name, s.product_id
FROM customer c
LEFT JOIN shoppingcart s ON c.customer_id = s.customer_id;

-- RIGHT JOIN between payment and customer
SELECT p.payment_id, c.customer_name
FROM payment p
RIGHT JOIN customer c ON p.customer_id = c.customer_id;
```

RELATIONAL ALGEBRA

1.Selection (**σ**)

σUsername = (CUSTOMER)

σProduct_name = (PRODUCT)

CUSTOMER_DETAILS

USER_id (PK)	Username	Password	Mobile_no	Email	Address	USER_id (PK)

PRODUCT

Product_id (PK)	Product_name	Category ID (FK)	Price

2. Projection(π)

Π(Mobile_no,Email) CUSTOMER

Mobile_no	Email

3. Union(U)

π(Category_ID)PRODUCT U π(Category_ID)PRODUCT CATEGORY

Category_ID

4.Set Difference(-)

Π PRODUCT – π PRODUCT CATEGORY

Product_id	Category_ID	Product_name	Price	decription

5.Set Intersection(∩)

ΠPAYMENT ∩ ΠSHOPPING DETAILS

Shopping_ID

6.Rename(ρ)

ρ(CUSTOMER/CUSTOMER_DETAIL**)R** will rename the attribute 'CUSTOMER' of the relation by 'CUSTOMER_DETAILS'.

CUSTOMER_DETAILS

USER_id (PK)	Username	Password	Mobile_no	Email	Address

7. Cross Product(X)

PAYMENT X SHOPPING DETAILS

Payment_ID	Shopping_ID	Order_ID (FK)	Payment_ method	Shopping_ID	Product_ID	Order_ID	Shipping_address	Delivery_date

NORMALIZATION

Since all the tables in the above database does not violate any of the rules of 1NF, 2NF, 3NF, BCNF or 4NF, thus we produce the following modifications in some of the tables in order to apply normalization over them:

1. SHOPPING CART

Shopping cart_Id	Customer _Id	Product_Details
s1	2	P1, P2, P5, P6
s2	3	P4, P5
s3	2	P3
s4	4	P3, P8
s5	5	P6, P9

First Normal Form (1NF)

Shopping cart_Id	Customer _Id	Product_Details
s1	2	P1
s1	2	P2
s1	2	P5
s1	2	P6
s2	3	P4
s2	3	P5
s3	2	P3
s4	4	P3
s4	4	P8
s5	5	P6
s5	5	P9

Second Normal Form (2NF)

Shopping cart_Id	User_Id
s1	2
s2	3
s3	2
s4	4
s5	5

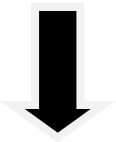
Table: SHOPPING CART

Shopping cart_Id	Product_Id
s1	P1
s1	P2
s1	P5
s1	P6
s2	P4
s2	P5
s3	P3
s4	P3
s4	P8
s5	P6
s5	P9

Table: SHOPPING CART _Products

2. PRODUCTS CATEGORY

Category_Id	Name	Has	Discount_Offer
C1	Fashion	4	50
C2	Mobiles	2	20
C3	Electronics	1	33
C4	Grocery	3	5
C5	Homeand Furniture	7	90



First Normal Form (1NF)

Since the table is already in 1NF we will move on to the next step that is converting the table into second normal form.

Second Normal Form (2NF)

Category_Id	Name	Has
C1	Fashion	D4
C2	Mobiles	D2
C3	Electronics	D1
C4	Grocery	D3
C5	Homeand Furniture	D5

Table:Categories

Discount_Id	Offer
D1	33
D2	20
D3	5
D4	50
D5	10

Table:Discount

RESULT

	product_id	category_id	product_name	price
▶	P001	M101	Smartphone	499.99
	P002	C101	Laptop	899.99
	P003	M701	Television	699.99
	P004	A101	Washing Machine	399.99
	P005	M202	Shirt	29.99
	P006	M202	Dress	49.99
	P007	M202	T-shirt	19.99
	P008	M201	Jeans	39.99
	P009	A101	Microwave Oven	149.99
	P010	K301	Blender	59.99
*	NULL	NULL	NULL	NULL

PRODUCT TABLE

	customer_id	customer_name	emailID
▶	101	Olivia	oliv12@gmail.com
	102	Eren	eren23@outlook.com
	103	Rahul	rbansal@gmail.com
	104	Cain	cc1234@gmail.com
	105	Lucy	lucy45@gmail.com
*	NULL	NULL	NULL

CUSTOMER TABLE

payment_id	customer_id	payment_method	card_number
1	101	Netbanking	1234620000
2	102	Creditcard	8573900000
3	103	Creditcard	6481030000
4	104	Netbanking	6482050000
5	105	Creditcard	2385030000
6	101	Debitcard	23849200000
7	102	Debitcard	16585000000
8	103	Netbanking	1687490000
9	104	Netbanking	4567460000
10	105	Creditcard	7914950000
NULL	NULL	NULL	NULL

PAYMENT TABLE

category_id	category_name
A101	Appliances
B401	Beauty
C101	Computers
H501	Health
K301	Kitchen
M101	Mobiles
M201	Men's Fashion
M202	Women's Fashion
M701	Movies
S601	Sports
V801	Videogames
NULL	NULL

PRODUCT CATEGORY

shipping_id	customer_id	address	city
1	101	123 Main St	Noida
2	102	456 Elm St	Delhi
3	103	789 Oak St	Ghaziabad
4	104	101 Pine St	Ghaziabad
5	105	202 Maple St	Noida
NULL	NULL	NULL	NULL

SHIPPING DETAILS TABLE

	customer_name	product_id
▶	Olivia	P001
	Olivia	P003
	Eren	P002
	Rahul	P005
	Cain	P004
	Lucy	P007

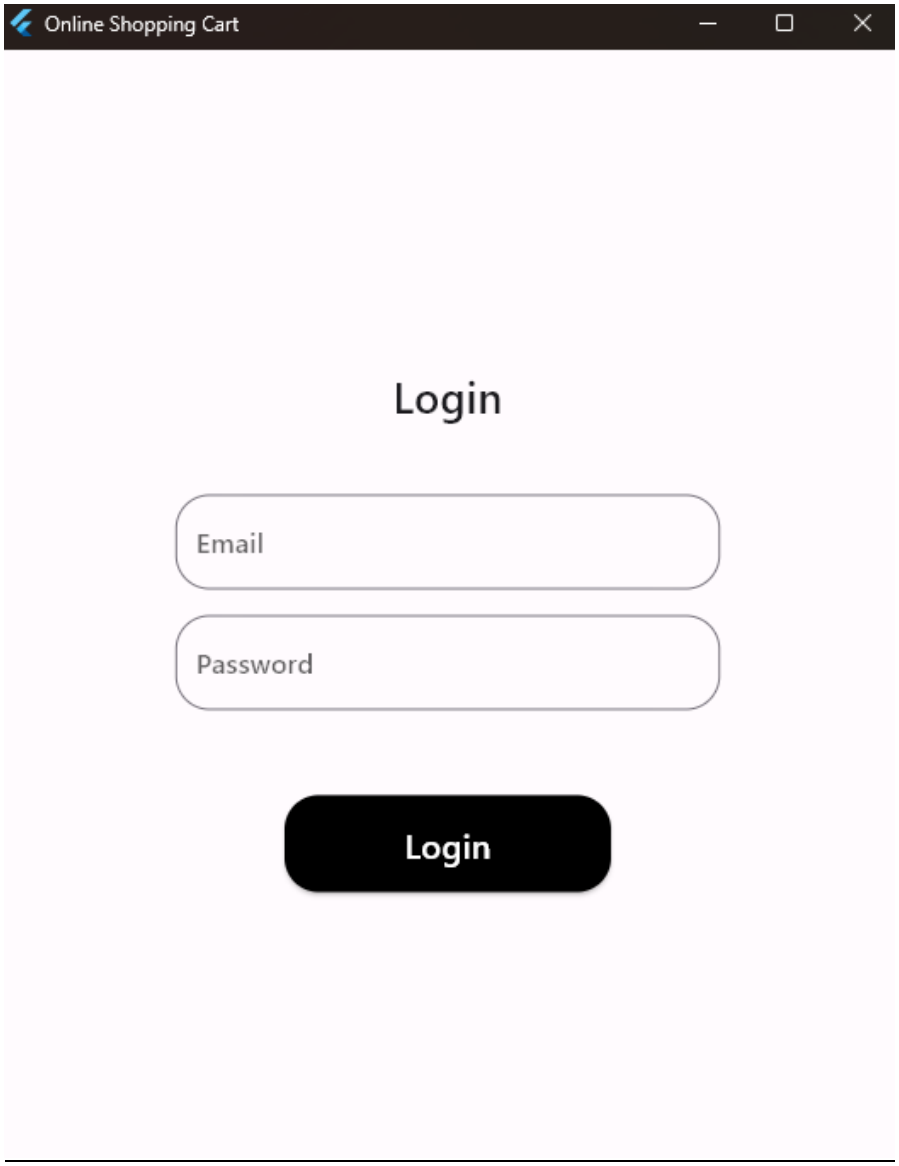
INNER JOIN

	payment_id	customer_name
▶	1	Olivia
	6	Olivia
	2	Eren
	7	Eren
	3	Rahul
	8	Rahul
	4	Cain
	9	Cain
	5	Lucy
	10	Lucy

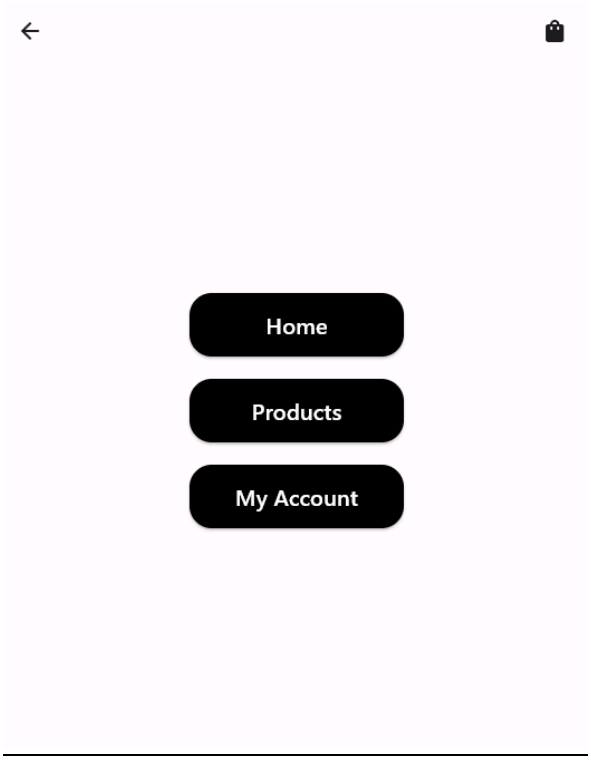
RIGHT JOIN

	customer_id	product_id	quantity
▶	101	P001	2
	101	P003	3
	102	P002	1
	103	P005	1
	104	P004	2
	105	P007	1
*	NULL	NULL	NULL

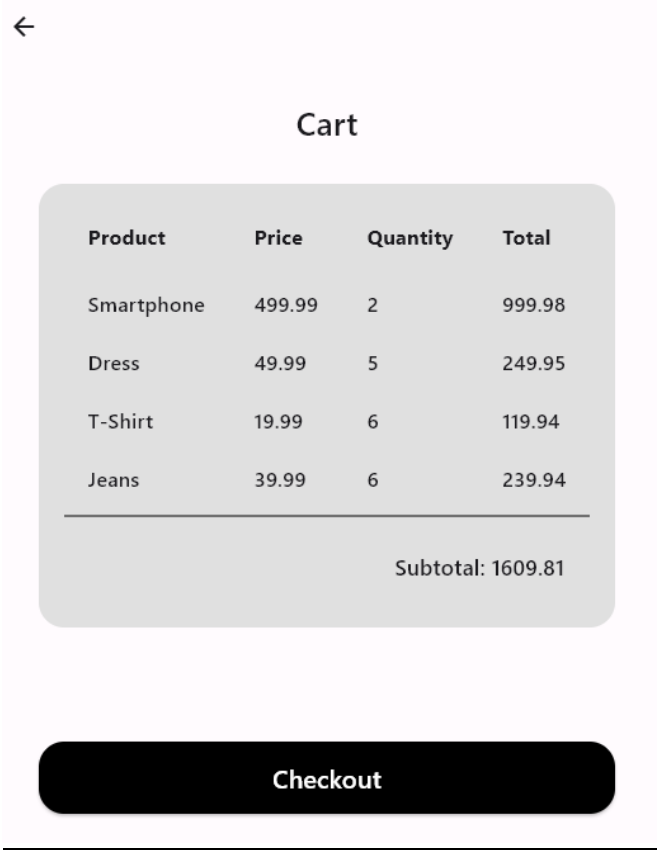
SHOPPING CART TABLE



LOGIN PAGE



HOME PAGE



CART