

PROJECT- SHOPPING APPLICATION

1.INTRODUCTION

1.1 Overview of project

Shopping Cart application that allows managing products, customers, and stores. The application will have two application roles:

- Administrators
 - To manage the products, customers and stores
 - To access a dashboard to identify the top products
- Customers
 - To review, add, delete, or edit the number of products in the shopping cart
 - To submit orders

1.2 Purpose

70% of clients prefer the shopping apps experience. Once a user has built a certain level of engagement with a brand through their site, they will switch to their mobile app for the actual shopping. Compared to the shopping experience on a mobile browser, shopping apps offer more convenience, speed and stored settings. Indeed, no need to remember a URL or to login every time. Speed is a crucial element too. Apps are generally 1.5 times faster than mobile sites as they don't need to fetch data from web servers.

2 THEORITICAL ANALYSIS

2.1 Block Diagram



Caption

2.2 Hardware/Software designing

Hardware

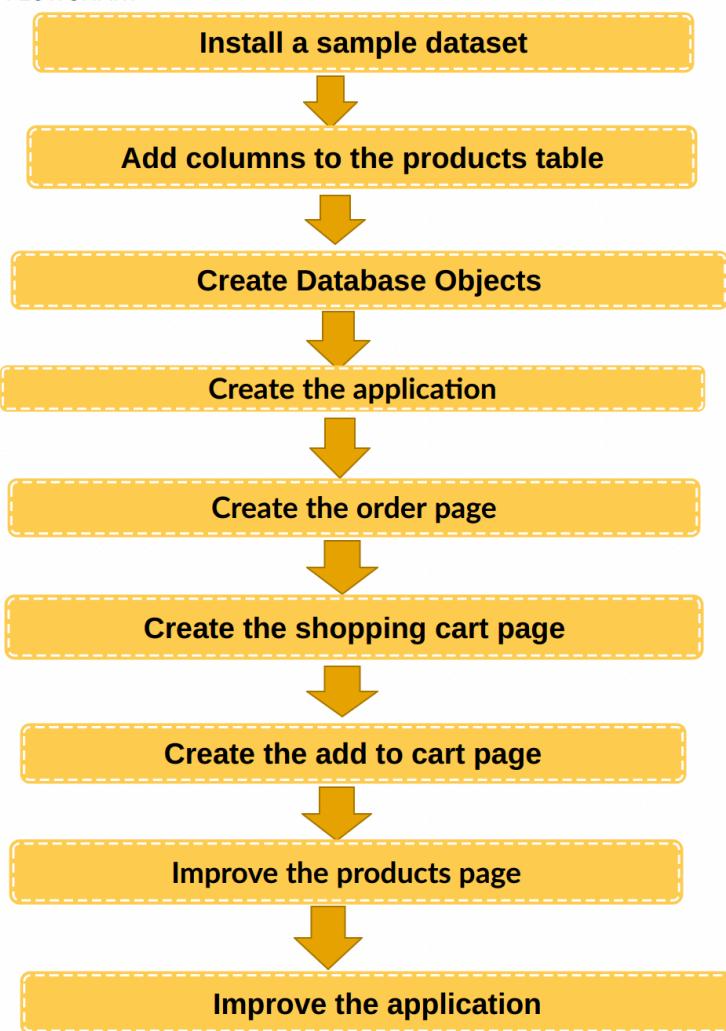
Oracle Database runs faster and more efficiently
Easy integration with Oracle's public cloud
Enhanced availability and data protection
Lower costs with high-performance consolidation

Software

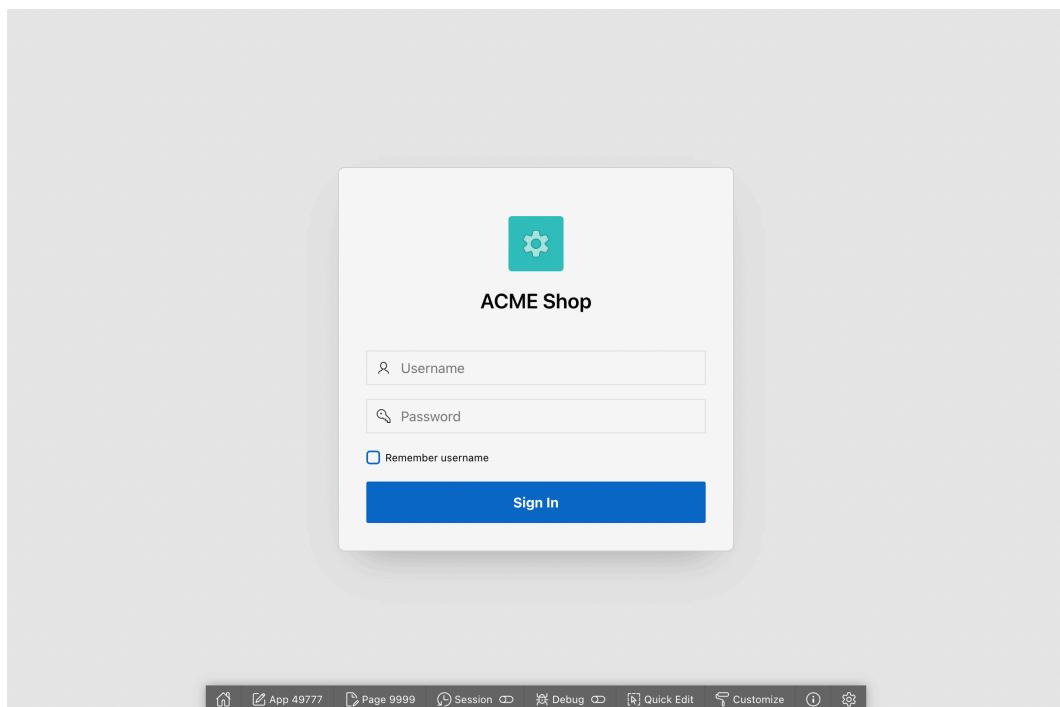
Oracle Database
Developer Tools
E-business suite
Engineered System

3 FLOWCHART

3 FLOWCHART



5 RESULT



Caption

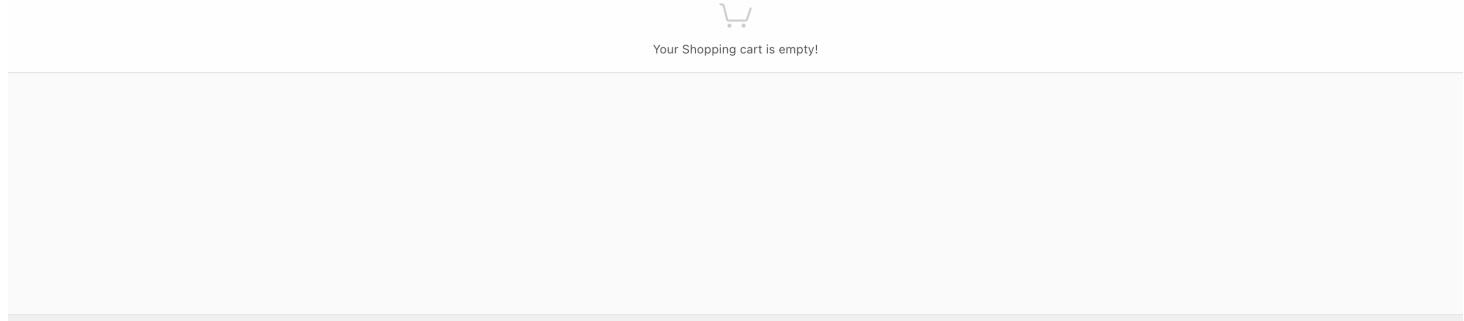
A screenshot of the ACME Shop application showing a list of products on the left and a "Manage to cart" modal on the right. The products listed include men's socks, a sweater, trousers, a coat, a hoodie, pyjamas, shorts, and a shirt. The "Manage to cart" modal displays a large image of a blue t-shirt, a quantity selector (set to 1), and "Update Quantity" and "Add to Cart" buttons. A sidebar on the right lists various items such as Boy's Coat(Brown), Boy's Jeans (Blue), etc.

Caption

Order Information

Email Address
Full Name
Store -Select a Store-

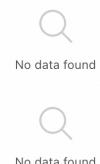
Shopping Cart



Caption

♥ Thank you for your order!

Order.&P17_ORDER



ACME Shop

Shopping cart administration ▾

Order Information

Caption

Administration

Application Administration

- Dashboard**
Manage component
- Clothing Lookup**
Manage component
- Color Lookup**
Manage component
- Customers**
Manage component
- Department Lookup**
Manage component
- Product Reviews**
Manage component
- Stores**
Manage component
- Manage Product**
Manage component

Access Control

Only users defined in the application access control list may access this application

Role	Count
Administrator	1
Contributor	0
Reader	0

Users
Set level of access for authenticated users of this application

Access Control
Change access control settings and disable access control

Caption

The screenshot shows a browser window for the Oracle APEX application at the URL apex.oracle.com/pls/apex/r/jin_a505_sql_s06/acme-shop123456789101112131415161718192012223242526272829303132333435363738394041424344454647484.... The page title is "Manage User Access". The user list table has one row:

Username	Roles
anushkabadi30@gmail.com	Administrator

The bottom navigation bar includes links for Home, App 49777, Page 10011, Session, Debug, Quick Edit, Customize, and Help.

Caption

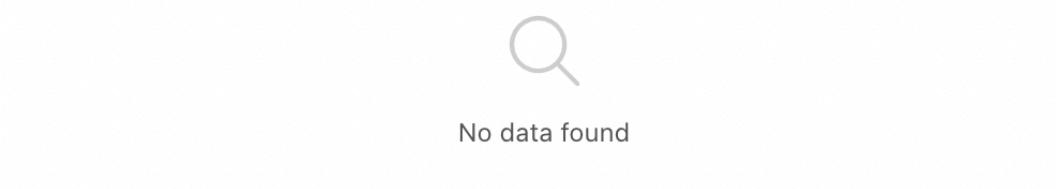
Administration

Application Administration		Access Control
 Dashboard Manage component	 Add User	
 Clothing Lookup Manage component	Only users defined in the application access control list may access this application	
 Color Lookup Manage component	Administrator	1
 Customers Manage component	Contributor	0
 Department Lookup Manage component	Reader	0
 Product Reviews Manage component	 Users Set level of access for authenticated users of this application	
 Stores Manage component	 Access Control Change access control settings and disable access control	
 Manage Product Manage component		

Caption

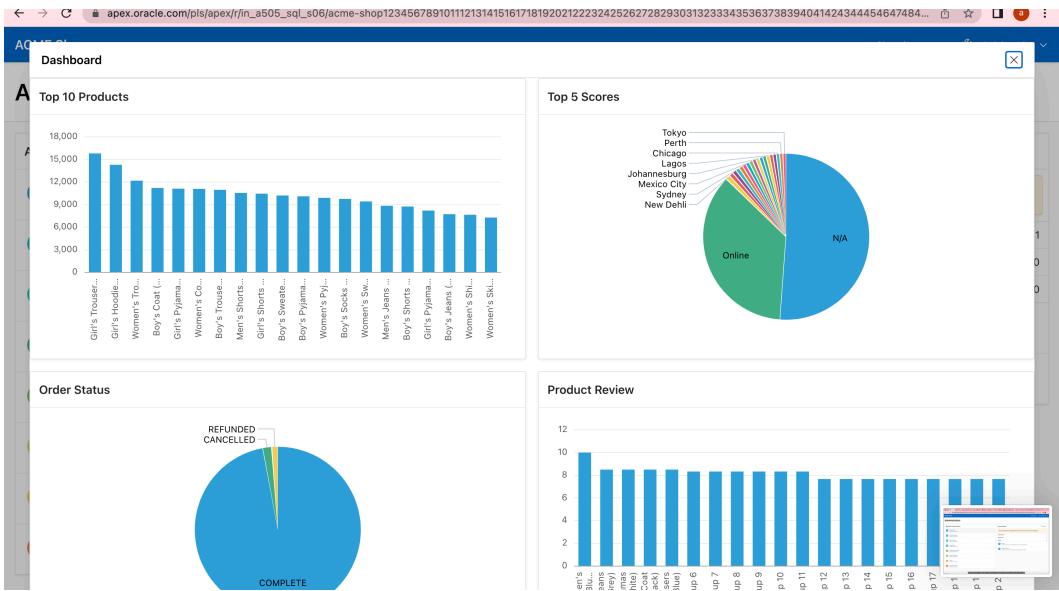


No data found

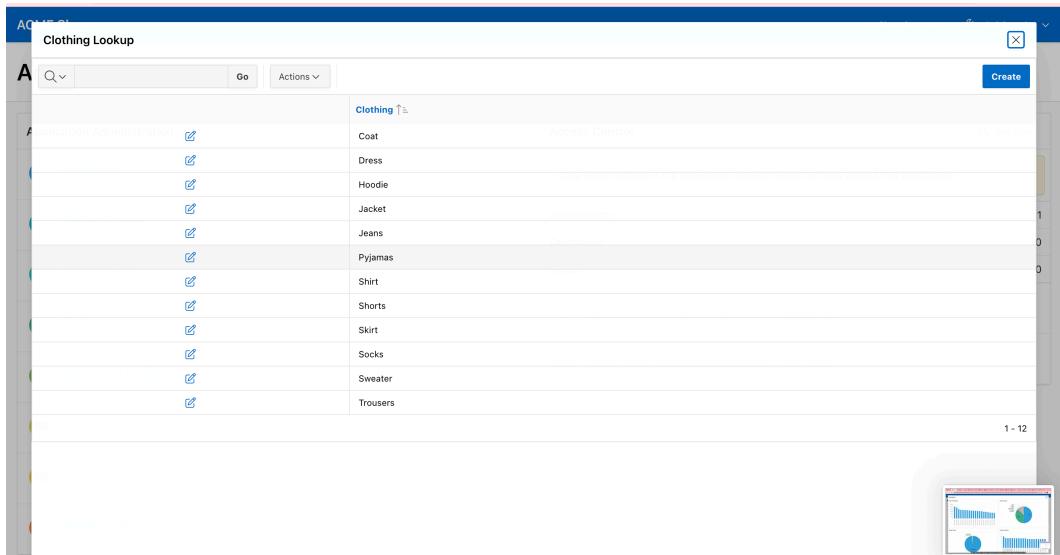


No data found

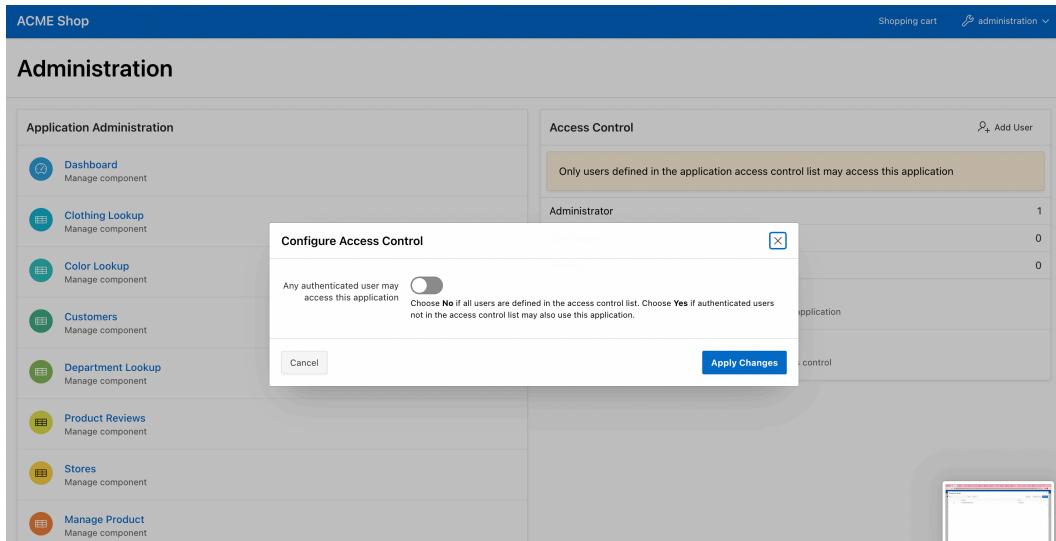
Caption



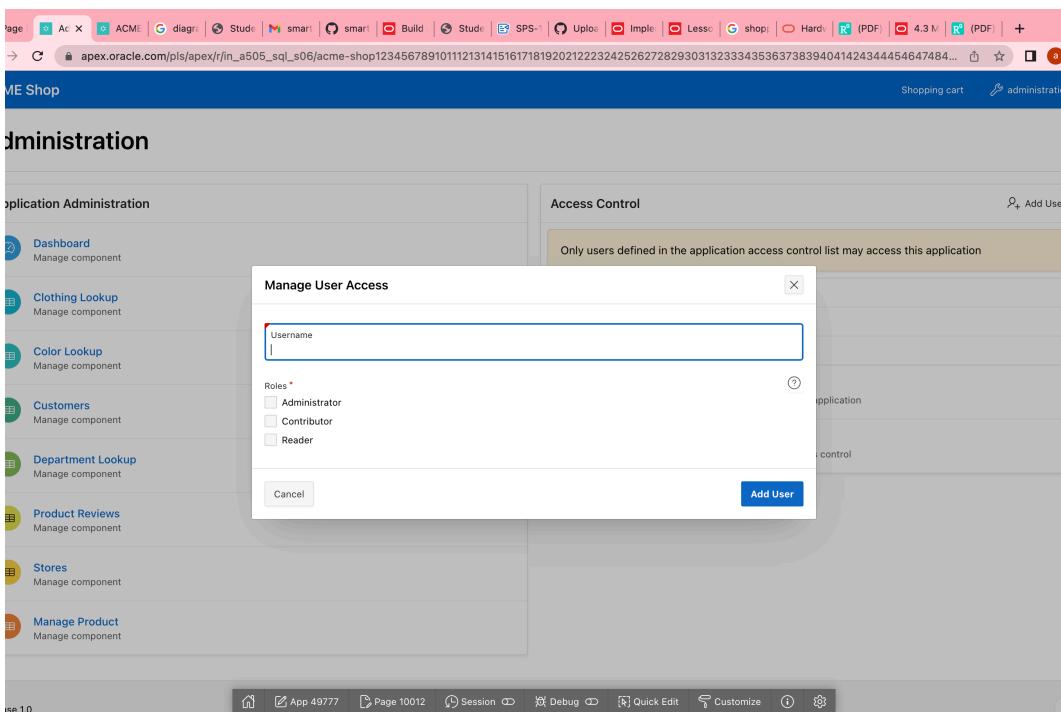
Caption



Caption



Caption



Caption

5 ADVANTAGES AND DISADVANTAGES

1. Shop From Anywhere

You can shop from the comfort of your own home, your workplace, or virtually anywhere. All you need is a device that's connected to the internet and a convenient address where you can receive the goods that you order.

2. Choice

You have a maximum amount of choice when buying products via the internet. You are not limited to what an individual store sells or stocks. If a product exists, then you will almost certainly be able to buy it online somewhere.

3. International Availability

As well as internal markets, shopping online gives you access to international goods. If you can't find the right product in your home country, then you can look abroad.

4. Prices Comparisons and Reviews

You have easy access to reviews of the products that you are interested in buying, enabling you to make more informed buying decisions. You can also easily compare prices at different vendors.

Disadvantages

1. Time Delay

There is typically a time delay in receiving your purchase while you wait for it to be delivered. Buy it in a brick-and-mortar store and you can take it home with you immediately.

2. Product Description

There may be potential problems with the seller when you buy online. They can describe a product wrongly, or in a way that is misleading, as they have almost total control over how a product is portrayed. They may also be more difficult to communicate with than your local store.

3. Pricing

Retailers can also use methods regarding the pricing of products that many buyers can find underhand. For instance, a product can be displayed at a certain price online, only for excessive shipping, or other costs, to be added later on in the buying process.

4. Delivery

You can have problems with delivery when ordering a product online - either it arrives broken, or it doesn't arrive at all. This may happen only rarely, but it can be very frustrating and time-consuming when it does occur.

6 APPLICATIONS

- Amazon India Online Shopping.
- Flipkart Online Shopping App.
- Myntra Online Shopping App.

7 CONCLUSION

Technology has made significant progress over the years to provide consumers a better online shopping experience and will continue to do so for years to come. With the rapid growth of products and brands, people have speculated that online shopping will overtake in-store shopping. While this has been the case in some areas, there is still demand for brick and mortar stores in market areas where the consumer feels more comfortable seeing and touching the product being bought. However, the availability of online shopping has produced a more educated consumer that can shop around with relative ease without having to spend a large amount of time. In exchange, online shopping has opened up doors to many small retailers that would never be in business if they had to incur the high cost of owning a brick and mortar store. At the end, it has been a win-win situation for both consumer and sellers.

8 FUTURE SCOPE

Oracle Apps are highly popular with large companies. Experience with them will make you very hireable. This is an opportunity to build upon your developer skills and make yourself marketable as a consultant. Before we launch into the future and scope of eCommerce in India, let us first understand what is e-commerce. To put it simply, electronic commerce refers to the purchase and sale of goods online or via the internet.

Sellers make websites where they display images of their products with price and description. Shoppers who buy the products have multiple payment options like COD, e-wallet, net banking, credit card, and so on. Online sellers have the responsibility of shipping the product to the buyer and ensuring safe and timely delivery.

There are different models of E-Commerce:

- **B2C** – Business to consumer; this refers to the sale of goods to the end-user directly
- **B2B** – a business that sells to another business; for example, office equipment, wholesalers, construction equipment sellers.
- **B2G** – Businesses that sell or deal only with Government organizations.

- **C2B** – this is when a customer creates something that adds value to a business, and the business consumes it. The best example is positive customer reviews.

Today with the high penetration of the internet and mobile devices – currently there are about 776.45 million internet connections in India – it has also become very easy to buy and sell online.

9 APPENDIX

Source code 1

UPDATE

(

```
    SELECT p.product_id,
           p.product_name,
           p.clothing,
           p.color,
           p.department,
           p.product_details
```

```
    FROM products p ) p
```

```
SET p.clothing = Substr(product_name, Instr(product_name, '
',1,1)+1, Instr(product_name, ' ',1, 2)+1 - Instr(product_name, '
',1,1)- 2),
```

p.color =

(

```
    SELECT c.color
```

```
    FROM json_table (p.product_details, '$' COLUMNS ( color
```

```
VARCHAR2(4000) path '$.colour') ) c),
```

p.department =

(

```
    SELECT g.department
```

```
    FROM json_table (p.product_details, '$' COLUMNS
```

```
( department VARCHAR2(4000) path '$.gender') ) g)
```

Source code 2

```
CREATE OR replace PACKAGE manage_orders
AS
```

```
-- create procedure for add a product temporarily
```

```
PROCEDURE add_product (
  p_product IN NUMBER,
  p_quantity IN NUMBER);
```

```
-- create procedure for remove a product temporarily
```

```

PROCEDURE remove_product (
    p_product IN NUMBER);

-- create function to get the number of items in the shopping cart
FUNCTION Get_quantity
RETURN NUMBER;

-- create procedure for validate if a product exists in the shopping cart
FUNCTION Product_exists(
    p_product IN NUMBER)
RETURN NUMBER;

-- create procedure for clear the cart
PROCEDURE clear_cart;

-- create function to validate a customer
FUNCTION Customer_exists(
    p_customer_email IN VARCHAR2)
RETURN NUMBER;

-- create procedure to insert orders
PROCEDURE create_order (
    p_customer      IN VARCHAR2 DEFAULT NULL,
    p_customer_email IN VARCHAR2,
    p_store         IN NUMBER,
    p_order_id      OUT orders.order_id%TYPE,
    p_customer_id   OUT NUMBER );
END manage_orders;

```

```

CREATE OR replace PACKAGE BODY manage_orders
AS
    PROCEDURE add_product (p_product IN NUMBER,
                           p_quantity IN NUMBER)
    IS
        BEGIN
            IF NOT apex_collection.collection_exists (p_collection_name => 'PRODUCTS')
            THEN
                apex_collection.create_collection(p_collection_name => 'PRODUCTS');
            END IF;

            apex_collection.add_member(p_collection_name => 'PRODUCTS',
                                      p_n001 => p_product,
                                      p_n002 => p_quantity);
        END add_product;

    PROCEDURE remove_product (p_product IN NUMBER)
    IS
        l_id NUMBER;
        BEGIN
            IF apex_collection.Collection_exists (p_collection_name => 'PRODUCTS')
            THEN
                SELECT seq_id
                INTO l_id
                FROM apex_collections a
                WHERE collection_name = 'PRODUCTS'
                    AND a.n001 = p_product;

                apex_collection.delete_member(p_collection_name => 'PRODUCTS',
                                              p_seq => l_id);
            END IF;

```

```

END remove_product;

FUNCTION get_quantity
RETURN NUMBER
IS
    l_items NUMBER := 0;
BEGIN
    IF apex_collection.collection_exists (p_collection_name => 'PRODUCTS')
    THEN
        SELECT SUM(n002)
        INTO l_items
        FROM apex_collections a
        WHERE collection_name = 'PRODUCTS';
    END IF;

    RETURN l_items;
END get_quantity;

FUNCTION product_exists(p_product IN NUMBER)
RETURN NUMBER
IS
    l_quantity NUMBER;
BEGIN
    IF apex_collection.collection_exists (p_collection_name => 'PRODUCTS')
    THEN
        SELECT a.n002
        INTO l_quantity
        FROM apex_collections a
        WHERE collection_name = 'PRODUCTS'
            AND a.n001 = p_product;

        RETURN l_quantity;
    ELSE
        RETURN 0;
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        RETURN 0;
END product_exists;

PROCEDURE clear_cart
IS
BEGIN
    IF apex_collection.collection_exists (p_collection_name => 'PRODUCTS')
    THEN
        apex_collection.truncate_collection(p_collection_name => 'PRODUCTS');
    END IF;
END clear_cart;

FUNCTION customer_exists(p_customer_email IN VARCHAR2)
RETURN NUMBER
IS
    l_customer customers.customer_id%TYPE;
BEGIN
    SELECT customer_id
    INTO l_customer
    FROM customers
    WHERE email_address = p_customer_email;

    RETURN l_customer;

```

```

EXCEPTION
  WHEN no_data_found THEN
    RETURN 0;
END customer_exists;

PROCEDURE create_order (p_customer      IN VARCHAR2,
                       p_customer_email IN VARCHAR2,
                       p_store          IN NUMBER,
                       p_order_id       OUT orders.order_id%TYPE,
                       p_customer_id    OUT NUMBER)
IS
BEGIN
  p_customer_id := customer_exists(p_customer_email);

  IF p_customer_id = 0 THEN
    INSERT INTO customers
      (full_name,
       email_address)
    VALUES  (p_customer,
              p_customer_email)
    returning customer_id INTO p_customer_id;
  END IF;

  INSERT INTO orders
    (order_datetime,
     customer_id,
     store_id,
     order_status)
  VALUES  (SYSDATE,
           p_customer_id,
           p_store,
           'OPEN')
  returning order_id INTO p_order_id;

  IF apex_collection.collection_exists (p_collection_name => 'PRODUCTS')
  THEN
    INSERT INTO order_items
      (order_id,
       line_item_id,
       product_id,
       unit_price,
       quantity)
    VALUES  (p_order_id,
              seq_id,
              p.product_id,
              p.unit_price,
              n002
    SELECT p_order_id,
           seq_id,
           p.product_id,
           p.unit_price,
           n002
    FROM  apex_collections a,
          products p
    WHERE collection_name = 'PRODUCTS'
      AND p.product_id = a.n001;
  END IF;

  apex_collection.delete_collection(p_collection_name => 'PRODUCTS');
END create_order;
END manage_orders;

```

