Information Systems and Data Modeling – IT1090

Assignment

Title: Online Fashion Store

Batch Number: Y1.S2.WD.CSNE.01.1	Group Number: MLB_WD_CSNE_01.01_06

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

IT22360496	WICKRAMA ARACHCHI D R	Dion ………………………… signature
IT22311122	BINATH N	[illegible] ………………………… signature
IT22324306	PERERA K C S P	[illegible] ………………………… signature
IT22365378	WICKRAMASINGHE W M S L	[illegible] ………………………… signature
IT22323798	KAVEESHA N	[illegible] ………………………… signature

Submitted on: 27/05/2023

Scenario

Customers can skim, pick, and purchase an assortment of style creations, including clothes, shoes, and additions, on the online fashion store. They can peruse the distinct creations on the website, decide the ones they like to purchase, and then add those items to their shopping cart. Clients can enrol on the website to preserve their shopping carts, revisit order histories, and obtain suggestions that are tailored to their affections. To draw and keep customers, the company presents an assortment of deals, upgrades, and loyalty programs in complement to a vast choice of trademarks, creators, and styles. After completing their purchase, shoppers can head to the checkout and pay to utilize a variety of choices, including credit/debit cards, PayPal, and other online payment modes. The online apparel store delivers users a user-friendly format and a tailored purchasing experience. The store also presents a user-friendly content management system, which allows website administrators to virtually control the goods, wares, orders, and customer data. Young people between the ages of 18 and 30 who are inquisitive about fashion and ongoing crazes make up the shop's major target demographic.

Requirements Analysis

1. Introduction

1.1 Motive for the system

The rationale for the system is to give an easy-to-use online fashion store that permits clients to peruse and buy garments, shoes, and extras from distinct brands and styles, as well as get customized suggestions and offers.

1.2 Extent of the system

The web-based store will incorporate many items from distinct brands and styles, with an emphasis on the objective demographic of young people aged 18 to 30 who are keen on design and the latest fads.

1.3 Goals and success standards of the task

The goals of the task are to foster a vigorous and protected Internet business platform that addresses the issues of the objective demographic and sustains the development and progress of the business. Success standards incorporate high consumer satisfaction evaluations, boosted sales income, and a developing client base.

1.4 References

The project will reference existing web-based style stores, best practices of Internet business, and important industry analysis and patterns.

1.5 Outline

The RAD will depict the functional and non-functional necessities of the framework, as well as give framework models and a glossary of significant terms.

2. Current Framework

The existing system is not relevant since the online fashion store is a fresh framework being created from scratch.

3. Proposed Framework

3.1 Outline

The proposed framework will be an internet-based fashion store that permits clients to peruse and buy garments, shoes, and frills from distinct brands and styles. The framework will likewise present customized proposals and offers to clients considering their shopping history and inclinations.

3.2 Functional requirements

3.2.1 Clients ought to have the option to peruse and choose garments, shoes, and extras from various brands and styles on the internet-based fashion store.

3.2.2 Clients can view and look for items on the site.

3.2.3 Clients ought to have the option to add items to their shopping baskets and continue to the checkout.

3.2.4 Clients ought to have the option to make an account on the site to save their shopping baskets, check order histories, and get customized ideas given their shopping history and inclinations.

3.2.5 The framework should offer deals, promotions, and loyalty schemes to draw in and keep clients.

3.2.6 Clients ought to have the option to pay for their items utilizing credit/debit cards, PayPal, and other web-based payment strategies.

3.2.7 The internet-based clothing store ought to give an easy-to-use design and customized shopping experience for clients.

3.2.8 The framework ought to offer an easy-to-use content administration framework that permits managers to oversee items, orders, and client information.

3.3 Non-functional requirements

3.3.1 **Convenience:** The site ought to be not difficult to explore and use, with a clear and natural plan and design.

3.3.2 **Security:** The site should be secure and protect client information from unapproved access.

3.3.3 **Versatility:** The site ought to have the option to deal with many clients and transactions without execution issues.

3.3.4 **Execution:** The site should give quick loading times to guarantee a smooth client experience.

3.4 Framework models

3.4.1 **Situations:** The framework will have a situation where another client peruses the store and makes an order and one more situation where a returning client logs in, sees their order history, and utilizes a loyalty program.

3.4.2 **Use Case Model:** The utilization cases incorporate perusing for items, adding items to a cart, making an account and a purchase, and utilizing loyalty programs.

3.4.3 **Analysis Object Model:** This model depicts the commodities in the framework, including clients, items, orders, and remittance strategies.

3.4.4 **Dynamic Model:** This model portrays the collaborations between the elements in the framework.

3.4.5 **UI:** The UI incorporates models showing the screens and navigational ways representing the series of screens.

4. Glossary

A glossary of significant terms connected with the internet-based style store, including client, item, order, payment technique, and loyalty program. This guarantees consistency in the particularizing and the utilization of the client's phrasing.

- **Account** - an enrolled client profile on the internet-based style store site that permits clients to get to their shopping basket, order history, and get customized suggestions.
- **Check out** - the way of checking and settling a client's shopping basket, choosing a remittance technique, and putting in an order.
- **Content administration framework** - a framework that permits managers to deal with the content and information on the site, including items, orders, and client data.
- **Credit/Debit card** - a payment strategy that permits clients to pay for their purchases utilizing their credit or debit card data.
- **Client** - an individual who utilizes the store to peruse and buy dresses, shoes, and embellishments.
- **Deal** - a bargain or discount given to clients to urge them to make a purchase.
- **Loyalty program** - a program that rewards clients for their steadfastness to the web-based style store, commonly by offering points or remunerates that can be recovered for future purchases.
- **Online fashion store** - a site that permits clients to peruse and buy dresses, shoes, and frills from various brands and styles.
- **PayPal** - an online payment method that permits clients to pay for their purchases utilizing their PayPal account.
- **Customized shopping experience** - a modified shopping experience for clients that considers their shopping history, inclinations, and different variables.
- **Product** - a dress item, shoe, or frill accessible for purchase on the web-based store site.
- **Shopping basket** - a component of the web-based store that permits clients to add items they wish to buy and survey their determinations before checkout.

- **Objective demographic -** the target group for the web-based store, for this situation young people aged 18 to 30 who are keen on style and the latest fads.

Data Requirements

Client data: This incorporates individual client data, including client ID, name, email address, mailing address, bill-to address, password, payment data, and purchase history. It is used to make client accounts, process orders, give modified suggestions, and provide benefits of the loyalty program. Email and password may be stored as null values as the client may not have created an account yet. The payment information can be only one of the following per client: 'Credit Card', 'Debit Card', 'PayPal', 'Bank Transfer', or 'Cash'. Each client may be managed by one security feature. Each client may be tracked by one or many loyalty programs. Each client may be tracked by one or many site usage features. Each client may be managed by one or many content management features. Each client may make one or many orders.

Item data: This incorporates insights regarding all items accessible on the site, including data such as item ID, item name, brand, style, size, category, description, price, image, comment, rating, and availability. It is used to show the items on the site, allow clients to look at and peruse items, and enable administrators to oversee stock. Each item's name should be unique. The size of the item can only be one of the twelve options stored in the database. The availability of an item can only be either "in stock" or "out of stock". The price of an item cannot be negative. The rating of an item can be only between 0 and 5. Each item may be managed by one or many content management features. Each item may be constituted in one or many orders.

Order data: This incorporates data for all orders set by clients, including the order ID, client ID, item ID, quantity, cost, order date, order time, delivery details, payment details, shopping cart ID, and order status. It is used to process and track orders, update stock, and produce sales reports. The quantity of an order should be above 0. The cost of an order cannot be negative. The order status can only be 'Shipped', 'Processing', or 'Delivered'. Each order may be managed by one or many content management features. Each order must constitute one or many items. Each order must contain one payment. Each order must be made by one client.

Payment data: This incorporates information — payment ID and payment method — connected with payment strategies accessible on the site, like credit or debit cards and PayPal. It is essential to manage payments safely and guarantee consistency with legitimate and administrative necessities. At the end of the day, this information is expected to safely manage exchanges. Each payment may be contained in one order.

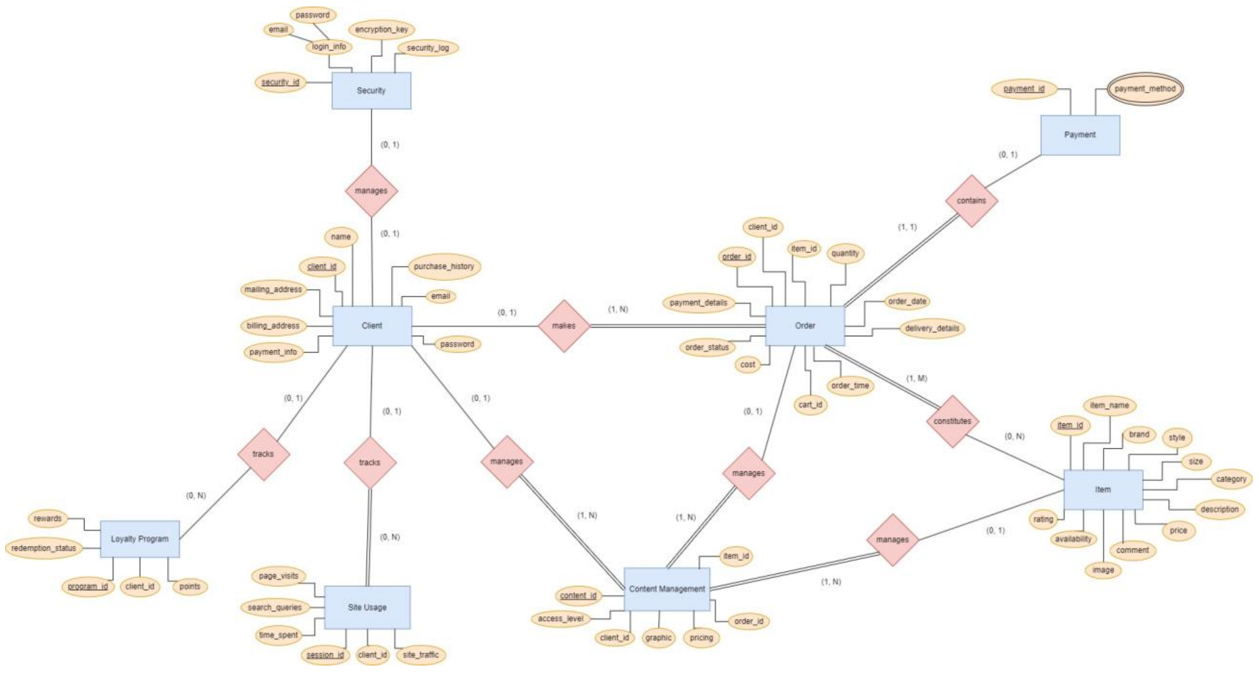
Content administration data: This incorporates information connected with dealing with the substance of the site, like content ID, item ID, order ID, client ID, illustrations, access level, and pricing. It is important to routinely manage and update the site's framework. Each content management feature must manage one client. Each content management feature must manage one order. Each content management feature must manage one item.

Loyalty program data: This incorporates all loyalty program data, including points acquired by clients, rewards accessible, redemption status, client ID, and program ID. It is used to deal with the loyalty program and give customised benefits given clients' shopping histories and inclinations. The redemption status of a loyalty program can be only one of the five levels stored in the database. The points of a loyalty program cannot be negative. Each loyalty program may track one client.

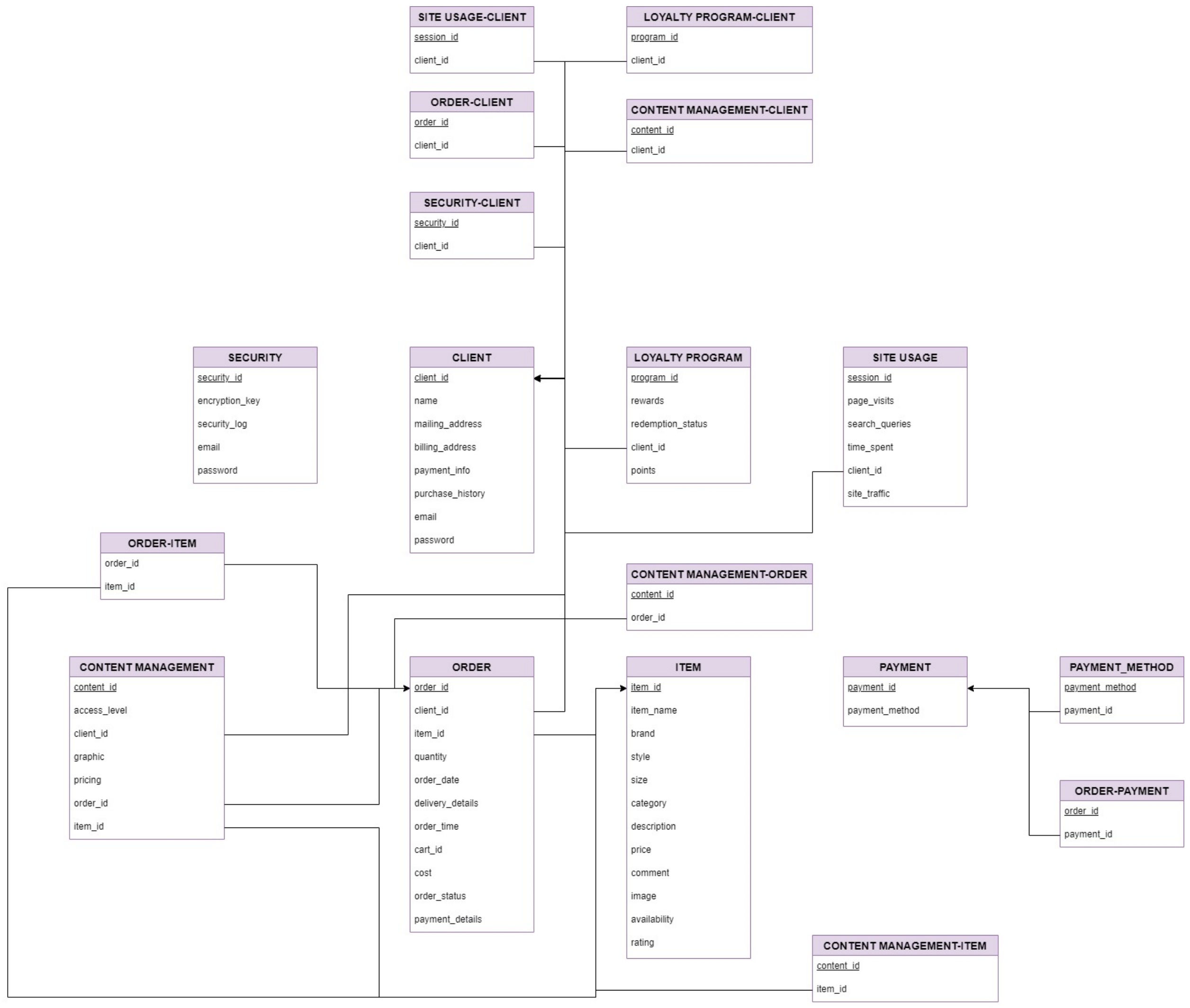
Use-of-site data: This incorporates information about how clients use the site, for example, site traffic (the number of guests), page visits, search inquiries, session ID, client ID, and time spent on each page. This information is used to advance the site for better execution and further develop the client experience. The number of page visits and time spent cannot be negative. The site traffic can only be either 'Low', 'Medium', or 'High'. Each site usage feature must track one client.

Security data: This information is expected to guarantee the security and protection of client information and forestall unapproved access, hacking, or data breakings. It incorporates information connected with safety efforts to safeguard client information from unapproved access, such as login data (email and password), encryption keys, security ID, and security logs. Similar to client data, each email should be unique and in the format of %_@__%.__% where the underscore character (_) represents any single character while the per cent sign character (%) represents a string of zero or more characters. Each password should be at least eight characters long. Each security feature may manage one client.

ER Diagram


password
email
encryption_key
login_info
security_log
security_id
Security
(0, 1)
manages
(0, 1)
payment_id
payment_method
Payment
(0, 1)
contains
(1, 1)
client_id
item_id
quantity
order_id
name
client_id
purchase_history
email
mailing_address
payment_details
order_date
billing_address
Client
(0, 1)
makes
(1, N)
Order
delivery_details
payment_info
password
order_status
cost
order_time
cart_id
(1, M)
item_name
(0, 1)
(0, 1)
(0, 1)
(0, 1)
constitutes
item_id
brand
style
(0, N)
tracks
tracks
manages
manages
size
category
Item
description
(0, N)
(1, N)
(1, N)
rating
price
availability
comment
image
(0, N)
manages
(0, 1)
rewards
Loyalty Program
redemption_status
program_id
client_id
points
page_visits
search_queries
Site Usage
time_spent
session_id
client_id
site_traffic
item_id
content_id
Content Management
(1, N)
access_level
order_id
client_id
graphic
pricing

Relation Schema



Database Tables with Sample Data

```
-- Create the CLIENT table

CREATE TABLE CLIENT (

  client_id INT NOT NULL,

  name VARCHAR(50) NOT NULL,

  mailing_address VARCHAR(320) NOT NULL,

  billing_address VARCHAR(320) NOT NULL,

  payment_info VARCHAR(255) NOT NULL,

  purchase_history VARCHAR(255) NOT NULL,

  email VARCHAR(100),

  password VARCHAR(256)

  CONSTRAINT CLIENT_PK PRIMARY KEY (client_id)

);



ALTER TABLE CLIENT

ADD CONSTRAINT payment_info_CHK CHECK (payment_info IN ('Credit Card', 'Debit Card',
'PayPal', 'Bank Transfer', 'Cash'))



-- Insert sample data into the CLIENT table
```

```
INSERT INTO CLIENT (client_id, name, mailing_address, billing_address, payment_info,
purchase_history, email, password)

VALUES

 (186, 'Brad Steffen', '3394 Wines Lane', '2055 Capitol Avenue', 'Credit Card', 'Fatigue, Bomber
Jacket, Boxers', 'bradsteffen@gmail.com', '9ijqX1Va4pkI2HZ'),

 (479, 'Anissa Kitchens', '1682 Meadowcrest Lane', '4997 Longview Avenue', 'Credit Card',
'Bikini, Flip Flops', 'anissakitchens@outlook.com', 'mNDQEa2Q01qJ8vH'),

 (699, 'Bridger Proudfoot', '1291 Ralph Drive', '1075 Center Street', 'Cash', 'Flannel Shirt',
'bridgerproudfoot@gmail.com', 'fvAIw3OpHwHhCU3'),

 (896, 'Kendal Alberts', '1979 Crummit Lane', '4301 Pyramid Valley Road', 'Bank Transfer',
'Burka', 'kendalalberts@yahoo.com', 'Llrp7yL49m5jGO8'),

 (998, 'Sommer Cannon', '3730 Upton Avenue', '3452 Hide A Way Road', 'PayPal', 'Leather Jacket,
Jacket', 'sommercannon@icloud.com', 'Sg9PVaLE6RjduKo');


-- Create the ITEM table

CREATE TABLE ITEM (

  item_id INT NOT NULL,

  item_name VARCHAR(256) NOT NULL,

  brand VARCHAR(30) NOT NULL,

  style VARCHAR(255) NOT NULL,

  size VARCHAR(3) NOT NULL,
  category VARCHAR(60) NOT NULL,

  description VARCHAR(300) NOT NULL,

  price DECIMAL(10, 2) NOT NULL,

  comment VARCHAR(900) NOT NULL,

  image VARCHAR(255) NOT NULL,

  availability VARCHAR(12) NOT NULL,

  rating DECIMAL(3, 2) NOT NULL

  CONSTRAINT ITEM_PK PRIMARY KEY (item_id)

);


ALTER TABLE ITEM

ADD CONSTRAINT item_name_CHK UNIQUE (item_name)

ALTER TABLE ITEM

ADD CONSTRAINT size_CHK CHECK (size IN ('XXS', 'XS', 'S', 'M', 'L', 'XL', 'XXL', '3XL',
'4XL', '5XL', '6XL', '7XL'))

ALTER TABLE ITEM

ADD CONSTRAINT availability_CHK CHECK (availability IN ('In Stock', 'Out of Stock'))

ALTER TABLE ITEM

ADD CONSTRAINT ITEM_CK_PRICE CHECK (price >= 0)
```

```
ALTER TABLE ITEM

ADD CONSTRAINT ITEM_CK_RATING CHECK (rating >= 0 AND rating <= 5)



-- Insert sample data into the ITEM table

INSERT INTO ITEM (item_id, item_name, brand, style, size, category, description, price, comment, image, availability, rating)

VALUES

  (6, 'Shoe', 'StyleBazaar', 'Warm', 'M', 'Shoe', 'Simplicity at its finest, a timeless silhouette and premium materials for versatility.', 57.99, 'Awesome community you have here guys, a lot of friendly people and a lot of good suggestions.', 'ShoeStyleBazaar.jpg', 'In Stock', 4.5),

  (34, 'Camisole', 'BuonChic', 'Wintery', 'M', 'Undergarment', 'Camisoles are essential for layering under blazers and cardigans.', 26.99, 'I enjoy your artwork, you have unique style.', 'CamisoleBuonChic.jpg', 'In Stock', 4.0),

  (43, 'Jacket', 'StyleBazaar', 'Bright', 'L', 'Outerwear', 'The only limit is your imagination, making this jacket a wardrobe staple.', 24.99, 'Can you believe this? This post is masterpiece. Details, illustrations, facts – everything is on the right place.', 'JacketStyleBazaar.jpg', 'Out of Stock', 3.8),

  (55, 'Ball Gown', 'PrimeElegance', 'Rich', 'M', 'Evening Dress', 'This ball gown is a blank canvas for creativity and imagination.', 174.99, 'Please don’t even mention it, I had so much trouble with it that I don’t even want to talk about it.', 'BallGownPrimeElegance.jpg', 'In Stock', 4.2),

  (62, 'Briefs', 'ShopFlaire', 'Warm', 'M', 'Undergarment', 'Minimalist briefs with clean lines, premium cotton, and a breezy feel.', 17.99, 'Is it me or comments section is flooded heavily with spam?', 'BriefsShopFlaire.jpg', 'In Stock', 4.7);
```

```
-- Create the ORDER_TABLE table

CREATE TABLE ORDER_TABLE (

  order_id INT NOT NULL,

  client_id INT NOT NULL,

  item_id INT NOT NULL,

  quantity INT NOT NULL,

  order_date DATE NOT NULL,

  delivery_details VARCHAR(255) NOT NULL,

  order_time TIME NOT NULL,

  cart_id INT NOT NULL,

  cost DECIMAL(10, 2) NOT NULL,

  order_status VARCHAR(10) NOT NULL,

  payment_details VARCHAR(255) NOT NULL,

  CONSTRAINT ORDER_TABLE_PK PRIMARY KEY (order_id),

  CONSTRAINT ORDER_TABLE_FK1 FOREIGN KEY (client_id) REFERENCES
CLIENT(client_id),

  CONSTRAINT ORDER_TABLE_FK2 FOREIGN KEY (item_id) REFERENCES
ITEM(item_id)

);
```

```sql
ALTER TABLE ORDER_TABLE

ADD CONSTRAINT quantity_CHK CHECK (quantity > 0)

ALTER TABLE ORDER_TABLE

ADD CONSTRAINT cost_CHK CHECK (cost >= 0)

ALTER TABLE ORDER_TABLE

ADD CONSTRAINT order_status_CHK CHECK (order_status IN ('Shipped', 'Processing', 'Delivered'))



-- Insert sample data into the ORDER_TABLE table

INSERT INTO ORDER_TABLE (order_id, client_id, item_id, quantity, order_date, delivery_details, order_time, cart_id, cost, order_status, payment_details)

VALUES

  (598, 186, 6, 622, '2023-06-05', '1626 Austin Secret Lane, Beaver, Utah', '11:15:39', 3, 8.99, 'Shipped', 'VISA, 4532013237505773, 5/2025, 524'),

  (922, 479, 34, 550, '2023-10-04', '3187 Quincy Street, Philadelphia, Pennsylvania', '12:40:34', 4, 54.99, 'Delivered', 'VISA, 4556261543959942, 4/2027, 593'),

  (594, 699, 43, 751, '2023-07-24', '2056 Crummit Lane, Nelson, Nebraska', '13:35:49', 1, 67.99, 'Processing', 'VISA, 4485018332328820, 3/2026, 648'),

  (257, 896, 55, 123, '2023-01-26', '1863 Essex Court, White River Junction, Vermont', '13:50:19', 4, 80.99, 'Shipped', 'VISA, 4019680417296555, 9/2025, 925'),

  (494, 998, 62, 343, '2023-03-28', '2574 Brentwood Drive, Auston, Texas', '15:20:14', 10, 62.39, 'Delivered', 'VISA, 4024007116538199, 8/2028, 976');
```

```
-- Create the PAYMENT table

CREATE TABLE PAYMENT (

  payment_id INT NOT NULL,

  payment_method VARCHAR(255) NOT NULL,

  CONSTRAINT PAYMENT_PK PRIMARY KEY (payment_id),

);



-- Insert sample data into the PAYMENT table

INSERT INTO PAYMENT (payment_id, payment_method)

VALUES

  (856, 'Mastercard'),

  (692, 'PayPal'),

  (754, 'Apple Pay'),

  (324, 'Google Pay'),

  (821, 'Amazon Pay');



-- Create the SECURITY table

CREATE TABLE SECURITY (

  security_id INT NOT NULL,
  encryption_key VARCHAR(255) NOT NULL,

  security_log VARCHAR(255) NOT NULL,

  email VARCHAR(100) NOT NULL,

  password VARCHAR(256) NOT NULL

  CONSTRAINT SECURITY_PK PRIMARY KEY (security_id)

);


ALTER TABLE SECURITY

ADD CONSTRAINT security_email_CHK UNIQUE (email)

ALTER TABLE SECURITY

ADD CONSTRAINT security_email_CHK2 CHECK (email LIKE '%_@__%.__%')

ALTER TABLE SECURITY

ADD CONSTRAINT security_password_CHK CHECK (LEN(password) >= 8)


-- Insert sample data into the SECURITY table

INSERT INTO SECURITY (security_id, encryption_key, security_log, email, password)

VALUES

  (5692, '@NcRfUjX', 'User "admin" successfully logged in from IP address 192.168.0.100',
'gebin87104@gmail.com', '6p8bzPRlpzFkMyP'),
  (5601, 'UkXp2s5v', 'Access denied for user "guest" from IP address 10.0.0.25. Invalid password attempt.', 'lowax94799@gmail.com', 'QV9ER1nKsakiZfJ'),

  (3367, 'dRgUjXn2', 'Detected suspicious activity from IP address 172.16.0.50. Security system has blocked the connection.', 'yecehib179@outlook.com', 'YxMLHgbWKnNpbYp'),

  (3283, 'J@NcRfTj', 'Critical error occurred. Application crashed unexpectedly. Error code: 500.', 'xacixi2000@yahoo.com', 'H9Pmg3dpioAwvGr'),

  (6198, '&E)H@McQ', 'User "john.doe" modified sensitive data in the database. Changes logged for auditing purposes.', 'sovomar476@gmail.com', 'aFKicOc9SAzSeAr');


-- Create the LOYALTY_PROGRAM table

CREATE TABLE LOYALTY_PROGRAM (

  program_id INT NOT NULL,

  rewards VARCHAR(255) NOT NULL,

  redemption_status VARCHAR(255) NOT NULL,

  client_id INT NOT NULL,

  points INT NOT NULL,

  CONSTRAINT LOYALTY_PROGRAM_PK PRIMARY KEY (program_id),

  CONSTRAINT LOYALTY_PROGRAM_FK FOREIGN KEY (client_id) REFERENCES CLIENT(client_id)

);
```

```
ALTER TABLE LOYALTY_PROGRAM

ADD CONSTRAINT redemption_status_CHK CHECK (redemption_status IN ('Redeemed', 'Not Redeemed', 'Partially Redeemed', 'Pending', 'Expired'))

ALTER TABLE LOYALTY_PROGRAM

ADD CONSTRAINT points_CHK CHECK (points >= 0)



-- Insert sample data into the LOYALTY_PROGRAM table

INSERT INTO LOYALTY_PROGRAM (program_id, rewards, redemption_status, client_id, points)

VALUES

  (10, '10% off next purchase', 'Redeemed', 186, 3463),

  (25, 'Free shipping on orders over $50', 'Not Redeemed', 479, 4262),

  (37, '$20 gift card', 'Partially Redeemed', 699, 3526),

  (49, 'Exclusive early access to new collections', 'Pending', 896, 2362),

  (68, 'VIP styling session', 'Expired', 998, 8574);



-- Create the SITE_USAGE table

CREATE TABLE SITE_USAGE (

  session_id INT NOT NULL,

  page_visits INT NOT NULL,
  search_queries VARCHAR(255) NOT NULL,

  time_spent INT NOT NULL,

  client_id INT NOT NULL,

  site_traffic VARCHAR(255) NOT NULL,

  CONSTRAINT SITE_USAGE_PK PRIMARY KEY (session_id),

  CONSTRAINT SITE_USAGE_FK FOREIGN KEY (client_id) REFERENCES
CLIENT(client_id)

);



ALTER TABLE SITE_USAGE

ADD CONSTRAINT page_visits_CHK CHECK (page_visits >= 0)

ALTER TABLE SITE_USAGE

ADD CONSTRAINT time_spent_CHK CHECK (time_spent >= 0)

ALTER TABLE SITE_USAGE

ADD CONSTRAINT site_traffic_CHK CHECK (site_traffic IN ('Low', 'Medium', 'High'))



-- Insert sample data into the SITE_USAGE table

INSERT INTO SITE_USAGE (session_id, page_visits, search_queries, time_spent, client_id,
site_traffic)

VALUES
```

```
  (20576, 13, 'summer dresses', 62, 186, 'High'),

  (49055, 4, 'men sneakers', 72, 479, 'Medium'),

  (38563, 7, 'handbags', 56, 699, 'Low'),

  (92357, 15, 'kids clothing', 512, 896, 'High'),

  (458469, 7, 'beauty products', 195, 998, 'Medium');


-- Create the CONTENT_MANAGEMENT table

CREATE TABLE CONTENT_MANAGEMENT (

  content_id INT NOT NULL,

  access_level VARCHAR(255) NOT NULL,

  client_id INT NOT NULL,

  graphic VARCHAR(255) NOT NULL,

  pricing VARCHAR(255) NOT NULL,

  order_id INT NOT NULL,

  item_id INT NOT NULL,

  CONSTRAINT CONTENT_MANAGEMENT_PK PRIMARY KEY (content_id),

  CONSTRAINT CONTENT_MANAGEMENT_FK1 FOREIGN KEY (client_id)
REFERENCES CLIENT(client_id),

  CONSTRAINT CONTENT_MANAGEMENT_FK2 FOREIGN KEY (order_id)
REFERENCES ORDER_TABLE(order_id),
  CONSTRAINT CONTENT_MANAGEMENT_FK3 FOREIGN KEY (item_id) REFERENCES
ITEM(item_id)

);



-- Insert sample data into the CONTENT_MANAGEMENT table

INSERT INTO CONTENT_MANAGEMENT (content_id, access_level, client_id, graphic,
pricing, order_id, item_id)

VALUES

  (39587, 'Admin', 186, 'banner1.jpg', 'Sale: $19.99', 598, 6),

  (20944, 'Manager', 479, 'promo_image.png', 'New Arrival: $49.99', 922, 34),

  (34005, 'User', 699, 'product_image.jpg', 'Clearance: $9.99', 594, 43),

  (40925, 'Admin', 896, 'logo.png', 'Limited Time Offer: $39.99', 257, 55),

  (55946, 'Employee', 998, 'background_image.jpg', 'Bundle Deal: $79.99', 494, 62);



-- Create the PAYMENT_METHOD table

CREATE TABLE PAYMENT_METHOD (

  payment_method VARCHAR(255) NOT NULL,

  payment_id INT NOT NULL,

  CONSTRAINT PAYMENT_METHOD_PK PRIMARY KEY (payment_method),
  CONSTRAINT PAYMENT_METHOD_FK FOREIGN KEY (payment_id) REFERENCES PAYMENT(payment_id)
);


-- Insert sample data into the PAYMENT_METHOD table
INSERT INTO PAYMENT_METHOD (payment_method, payment_id)
VALUES
  ('Mastercard', 856),
  ('PayPal', 692),
  ('Apple Pay', 754),
  ('Google Pay', 324),
  ('Amazon Pay', 821);


-- Create the SITE_USAGE_CLIENT table
CREATE TABLE SITE_USAGE_CLIENT (
  session_id INT NOT NULL,
  client_id INT NOT NULL,
  CONSTRAINT SITE_USAGE_CLIENT_PK PRIMARY KEY (session_id),
  CONSTRAINT SITE_USAGE_CLIENT_FK FOREIGN KEY (client_id) REFERENCES CLIENT(client_id)
```

```
);


-- Insert sample data into the SITE_USAGE_CLIENT table

INSERT INTO SITE_USAGE_CLIENT (session_id, client_id)

VALUES

  (10247, 186),

  (21928, 479),

  (30289, 699),

  (40974, 896),

  (50219, 998);


-- Create the LOYALTY_PROGRAM_CLIENT table

CREATE TABLE LOYALTY_PROGRAM_CLIENT (

  program_id INT NOT NULL,

  client_id INT NOT NULL,

  CONSTRAINT LOYALTY_PROGRAM_CLIENT_PK PRIMARY KEY (program_id),

  CONSTRAINT LOYALTY_PROGRAM_CLIENT_FK FOREIGN KEY (client_id)
REFERENCES CLIENT(client_id)

);
```

```
-- Insert sample data into the LOYALTY_PROGRAM_CLIENT table

INSERT INTO LOYALTY_PROGRAM_CLIENT (program_id, client_id)

VALUES

  (10, 186),

  (25, 479),

  (37, 699),

  (49, 896),

  (68, 998);



-- Create the ORDER_CLIENT table

CREATE TABLE ORDER_CLIENT (

  order_id INT NOT NULL,

  client_id INT NOT NULL,

  CONSTRAINT ORDER_CLIENT_PK PRIMARY KEY (order_id),

  CONSTRAINT ORDER_CLIENT_FK FOREIGN KEY (client_id) REFERENCES
CLIENT(client_id)

);
```

```
-- Insert sample data into the ORDER_CLIENT table

INSERT INTO ORDER_CLIENT (order_id, client_id)

VALUES

  (1098, 186),

  (2436, 479),

  (3255, 699),

  (5374, 896),

  (5266, 998);


-- Create the CONTENT_MANAGEMENT_CLIENT table

CREATE TABLE CONTENT_MANAGEMENT_CLIENT (

  content_id INT NOT NULL,

  client_id INT NOT NULL,

  CONSTRAINT CONTENT_MANAGEMENT_CLIENT_PK PRIMARY KEY (content_id),

  CONSTRAINT CONTENT_MANAGEMENT_CLIENT_FK FOREIGN KEY (client_id)
REFERENCES CLIENT(client_id)

);


-- Insert sample data into the CONTENT_MANAGEMENT_CLIENT table
```

```
INSERT INTO CONTENT_MANAGEMENT_CLIENT (content_id, client_id)
VALUES
  (39587, 186),
  (20944, 479),
  (34005, 699),
  (40925, 896),
  (55946, 998);


-- Create the SECURITY_CLIENT table
CREATE TABLE SECURITY_CLIENT (
  security_id INT NOT NULL,
  client_id INT NOT NULL,
  CONSTRAINT SECURITY_CLIENT_PK PRIMARY KEY (security_id),
  CONSTRAINT SECURITY_CLIENT_FK FOREIGN KEY (client_id) REFERENCES
CLIENT(client_id)
);


-- Insert sample data into the SECURITY_CLIENT table
INSERT INTO SECURITY_CLIENT (security_id, client_id)
```

```
VALUES
  (5692, 186),
  (5601, 479),
  (3367, 699),
  (3283, 896),
  (6198, 998);


-- Create the ORDER_ITEM table
CREATE TABLE ORDER_ITEM (
  order_id INT NOT NULL,
  item_id INT NOT NULL,
  CONSTRAINT ORDER_ITEM_FK1 FOREIGN KEY (order_id) REFERENCES ORDER_TABLE(order_id),
  CONSTRAINT ORDER_ITEM_FK2 FOREIGN KEY (item_id) REFERENCES ITEM(item_id)
);


-- Insert sample data into the ORDER_ITEM table
INSERT INTO ORDER_ITEM (order_id, item_id)
VALUES
```

```
  (598, 6),

  (922, 34),

  (594, 43),

  (257, 55),

  (494, 62);


-- Create the CONTENT_MANAGEMENT_ORDER table

CREATE TABLE CONTENT_MANAGEMENT_ORDER (

  content_id INT NOT NULL,

  order_id INT NOT NULL,

  CONSTRAINT CONTENT_MANAGEMENT_ORDER_PK PRIMARY KEY (content_id),

  CONSTRAINT CONTENT_MANAGEMENT_ORDER_FK FOREIGN KEY (order_id)
REFERENCES ORDER_TABLE(order_id)

);


-- Insert sample data into the CONTENT_MANAGEMENT_ORDER table

INSERT INTO CONTENT_MANAGEMENT_ORDER (content_id, order_id)

VALUES

  (39587, 598),
  (20944, 922),

  (34005, 594),

  (40925, 257),

  (55946, 494);


-- Create the ORDER_PAYMENT table

CREATE TABLE ORDER_PAYMENT (

  order_id INT NOT NULL,

  payment_id INT NOT NULL,

  CONSTRAINT ORDER_PAYMENT_PK PRIMARY KEY (order_id),

  CONSTRAINT ORDER_PAYMENT_FK FOREIGN KEY (payment_id) REFERENCES
PAYMENT(payment_id)

);


-- Insert sample data into the ORDER_PAYMENT table

INSERT INTO ORDER_PAYMENT (order_id, payment_id)

VALUES

  (598, 856),

  (922, 692),
  (594, 754),

  (257, 324),

  (494, 821);


-- Create the CONTENT_MANAGEMENT_ITEM table

CREATE TABLE CONTENT_MANAGEMENT_ITEM (

  content_id INT NOT NULL,

  item_id INT NOT NULL,

  CONSTRAINT CONTENT_MANAGEMENT_ITEM_PK PRIMARY KEY (content_id),

  CONSTRAINT CONTENT_MANAGEMENT_ITEM_FK FOREIGN KEY (item_id)
REFERENCES ITEM(item_id)

);


-- Insert sample data into the CONTENT_MANAGEMENT_ITEM table

INSERT INTO CONTENT_MANAGEMENT_ITEM (content_id, item_id)

VALUES

  (39587, 6),

  (20944, 34),

  (34005, 43),
```

(40925, 55),

(55946, 62);

Performance Requirements

Scalability: The infrastructure and architecture of the system should be scalable so that it can manage more users and transactions at once without experiencing performance degradation.

Responsiveness: For a seamless user experience, the website should load quickly, with quick page loads, quick search results, and instant changes when adding things to the shopping cart.

Security: By using strong security methods like encryption, secure payment gateways, and protection against typical web vulnerabilities, the system should prioritise the security of user data and payment information.

Search and Filtering: The system should offer effective search and filtering capabilities so that users may identify things quickly depending on their preferences. The system's search performance can be improved by using indexing techniques and implementing optimised search algorithms.

Payment Processing: To manage online transactions smoothly, the system should have dependable and secure payment gateways. Fast and secure payment processing is necessary to protect customer payment information and expedite transaction completion.

Inventory Control: To track product availability in real-time, the system must have effective inventory control. This helps decrease consumer unhappiness, avoid overselling and selling out-of-stock items, and ensure proper order fulfilment.

Caching and Content Distribution: By lowering server load and speeding up the distribution of content to users, caching, and content delivery networks (CDNs) can enhance system performance. Content delivery can be accelerated by using CDN servers located closer to the user's location and caching frequently accessed data.

Analytics and Monitoring: Analytics and monitoring technologies can offer perceptions of user behaviour, website performance, and system bottlenecks, and performance monitoring mechanisms can track and examine user behaviour, system performance, and potential problems. Performance optimisation should be done to find and fix any bottlenecks or performance problems.

Availability: To guarantee that users can always access the website, the system should strive for high availability. Redundancy and fault-tolerant design can accomplish this, including load balancing, backup servers, and disaster recovery plans.

Mobile Optimisation: Because they offer a fluid and user-friendly experience across various screen sizes and resolutions, mobile optimisation and mobile responsiveness are crucial for online shopping. Both should be optimised in the system.

Integration with Third-Party Services: To guarantee smooth processing and undersized disruption, integration with third-party services should be taken into consideration. To enable efficient data flow and transaction processing, different systems, such as payment gateways and stock administration systems, should be linked. It is likewise important to consider the performance and reliability of these integrations.

Security Requirements

Secure Payment Handling: Secure payment handling should guarantee that all payment exchanges are managed safely, and that susceptible payment data is encrypted during transmission to safeguard it from unapproved access. Safety efforts should also incorporate compliance with payment gateway integration exercises. This will guarantee the security of sensitive payment data like credit or debit card details and PayPal account data.

Client Validation and Approval: The framework should have strong client verification systems to check the identity of clients and forestall unapproved access to client records and individual data. These incorporate robust password guidelines, multifaceted verification, and session administration controls. Access control measures should likewise be enforced to guarantee clients can only access the suitable resources and activities established for their positions and privileges.

Insurance of Client Information: The system should use proper safety efforts to safeguard client information and comply with security guidelines, like the General Data Protection Regulation (GDPR), and best practices to safeguard client information. This incorporates conducting information encryption, secure storage, and stringent access controls to forestall unapproved access, information breaks, or information leaks.

Secure Shopping Cart and Checkout: The shopping basket and checkout procedure should be secure to safeguard client payment data and forestall alteration or interception of susceptible data. This can be accomplished through secure coding practices, encryption, and routine security testing.

Standard Security Reviews and Updates: Customary security reviews and updates should be conducted to address any recognised weaknesses or security shortcomings. Software updates and fixes should be modernised to address any known weaknesses and safeguard against potential security breaches. Malevolent exercises should be prevented through firewalls, intrusion detection systems, and standard security reviews.

Information Backup and Catastrophe Recuperation: The system should have routine data backup methods set up to guarantee the accessibility and reliability of client information.

Moreover, a catastrophe recovery plan should be enforced to relieve the effects of any potential framework failures or data breaks.

Incident Response and Monitoring: The framework should have strategies set up to observe and react to security occurrences like uncommon client activity or dubious transactions. It should likewise have vigorous logging and monitoring capacities to identify and react to any security happenings or dubious activities, for example, access logs, system logs, and intrusion detection and prevention systems.