**Ecommerce Design**

**DATABASE SCHEMA:**

Customer table => Used to store the details of the customer

Security table => Handles the security details

Product table => Used to store the details of the product to be purchased

Catogory table => Used to store the variety of catogories

Cart table => Used to store the details of the product to be added to the cart

Order table => Used to store the details of the order such as the user info, order date

Payment table => Used to store the details of various payment details

OrderedItem table => Used to store the details of ordered item on the each order

|  |  |  |
| --- | --- | --- |
| **CUSTOMER** | | |
| Email | Varchar(255) | Primary key |
| FirstName | Varchar(255) |  |
| LastName | Varchar(255) |  |
| ContactNumber | Varchar(255) |  |
| Address | Varchar(255) |  |
| City | Varchar(255) |  |
| Country | Varchar(255) |  |
| ZipCode | Varchar(255) |  |
| ProfilePhoto | Varchar(max) |  |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **SECURITY** | | |
| Email | Varchar(255) | Primary Key & Foreign key from Customer table |
| Password | varbinary(max) |  |
| SecurityQuestion | Varchar(max) |  |
| Answer | Varbinary(max) |  |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **PRODUCT** | | |
| ProductId | int | Primary key (Identity) |
| ProductName | Varchar(max) |  |
| ProductImage | Varchar(max) |  |
| StockCount | int |  |
| Price | Decimal(10, 2) |  |
| CategoryId | int | Foreign key from Category table |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **CATEGORY** | | |
| CategoryId | Identity | Primary Key |
| CategoryName | Varchar(max) |  |
| Description | Varchar(max) |  |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **CART** | | |
| CartId | int | Primary key (Identity) |
| BelongsTo | string | Foreign key from Customer table |
| IsEnabled | Bit | Default is 1 |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **CARTITEM** | | |
| CartItemId | int | Primary key (Identity) |
| CartId | int | Foreign key from Cart table |
| ProductId | int | Foreign key from Product table |
| Quantity | int |  |
| TotalPrice | Decimal(10,2) |  |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **ORDER** | | |
| OrderId | Varchar(255) | Primary key |
| CustomerId | Varchar(255) | Foreign key from customer table |
| OrderDate | DateTime |  |
| Status | Varchar(255) | Placed /Delivered/Canceled |
| TotalAmount | Decimal(10, 2) |  |
| PaymentId | int | Foreign key from Payment table |
| ShippingId | int | Foreign key from shipping table |
| CartId | int | Nullable/ Foregin key from Cart table |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **PAYMENT** | | |
| PaymentId | int | Primary Key(Identity) |
| PaymentType | Varchar(255) |  |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **ORDERITEM** | | |
| OrderItemId | int | Primary key (Identity) |
| OrderId | Varchar(255) | Foreign key from Order table |
| ProductId | int | Foreign key from Product table |
| Quantity | int |  |
| TotalPrice | Decimal(10, 2) |  |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

|  |  |  |
| --- | --- | --- |
| **SHIPPING** | | |
| ShippingId | int | Primary key (Identity) |
| Address | Varchar(255) |  |
| City | Varchar(255) |  |
| Country | Varchar(255) |  |
| ZipCode | Varchar(255) |  |
| Landmark | Varchar(255) |  |
| BelongsTo | Varchar(255) | Foreign key from Customer table |
| CreatedBy | Varchar(255) | email or admin |
| CreatedDate | DateTime | Default GETDATE() |
| UpdatedBy | Varchar(255) | email or admin |
| UpdatedDate | DateTime | Default GETDATE() |

**USE CASES**

**1. Customer Registration/Login:**

- If the customer is a new user, they may need to register by providing their personal information, such as name, email address, and password.

- If the customer is a returning user, they can log in using their previously created account credentials.

**2. Browsing Products:**

- Once logged in, the customer can start browsing products on the website or app. They can use search filters, categories, or a search bar to find the products they are interested in.

**3. Product Page:**

On the product details page, you can display two prominent buttons or options for the customer to choose from:

- "**Buy Now**" Button: This button allows customers who are ready to make an immediate purchase to skip the cart and proceed directly to the checkout process.

- "**Add to Car**t" Button: This button allows customers to add the product to their shopping cart, indicating an intention to purchase but not immediately.

The two buttons can be displayed side by side or as two separate options, making it clear to the customer that they have a choice.

**4. Viewing the Cart:**

- The customer can view the contents of their shopping cart at any time by clicking on a "Cart" or "Shopping Cart" icon.

**5. Managing Cart Items:**

- In the cart view, the customer can review the items they've added, update quantities, or remove items if needed.

**6. Proceed to Checkout:**

- Once the customer is satisfied with their cart, they can proceed to the checkout process.

- This may involve confirming the delivery address, selecting a payment method, and reviewing the order summary.

**WORKFLOW DIAGRAM**

Index

**Order processing:**

**Order success!**

My Orders

View cart items

Order summary

Payment option

Shipping Address

Buy now

**Yes**

**No**

Register

Add to cart

Login

Is registered user?

Browse products

Product view

**User Registration:**

Login

Index

Personal details

Security details

**User registered successfully!**

**API DESIGN**

**Models:**

* Customer
* Security
* Product
* Category
* Cart
* CartItem
* Payment
* Shipping
* Order
* OrderedItem

|  |  |  |
| --- | --- | --- |
| **Customer** | | |
| Email | string | Primary key |
| FirstName | string |  |
| LastName | string |  |
| ContactNumber | string |  |
| Address | string |  |
| City | string |  |
| Country | string |  |
| ZipCode | string |  |
| ProfilePhoto | string |  |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **Product** | | |
| ProductId | int | Primary key |
| ProductName | string |  |
| ProductImage | string |  |
| StockCount | int |  |
| Price | Decimal |  |
| CategoryId | int | Foreign key from Category model |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **Security** | | |
| Email | string | Primary key also Foreign key from Customer model |
| Password | Byte[] |  |
| SecurityQuestion | string |  |
| Answer | Byte[] |  |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **Catogory** | | |
| CatogoryId | int | Primary key |
| CategoryName | string |  |
| Description | string |  |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **Cart** | | |
| CartId | int | Primary key |
| BelongsTo | string | Foregin Key from Customer model |
| IsEnabled | bool | Default is true |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **CartItem** | | |
| CartItemId | int | Primary key |
| CartId | int | Foreign key from Cart model |
| ProductId | int | Foregin Key from Product model |
| Quantity | int |  |
| TotalPrice | Decimal |  |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **Order** | | |
| OrderId | string | Primary key |
| CustomerId | String | Foreign key from customer model |
| OrderDate | DateTime |  |
| Status | string |  |
| TotalAmount | Decimal |  |
| PaymentId | int | Foreign key from Payment model |
| ShippingId | int | Foreign key from Shipping model |
| CartId | int | Nullable/ Foregin key from Cart model |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **Payment** | | |
| PaymentId | int | Primary key |
| PaymentType | string |  |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **OrderItem** | | |
| OrderItemId | int | Primary key |
| OrderId | string |  |
| ProductId | int | Foreign key from Product model |
| Quantity | int |  |
| TotalPrice | Decimal |  |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

|  |  |  |
| --- | --- | --- |
| **Shipping** | | |
| ShippingId | int | Primary key |
| Address | string |  |
| City | string |  |
| Country | string |  |
| ZipCode | string |  |
| Landmark | string |  |
| BelongsTo | string | Foreign key from Customer model |
| CreatedBy | string |  |
| CreatedDate | DateTime |  |
| UpdatedBy | string |  |
| UpdatedDate | DateTime |  |

**Controllers with Action verb:**

**Customer Controller:**

* GET
* GetCustomerByCustomerId(string email)
* GetAllCustomers()
* POST
  + AddCustomer(Customer customer)
* PUT
  + EditCustomer(Customer customer)
* DELETE
  + DeleteCustomerById(string email)

**Security Controller:**

* GET
* GetSecurityDetailsByCustomerId(string email)
* POST
* AddSecurityDetails(Security security)
* PUT
* EditSecurityDetails(Security security)
* DELETE
* DeleteSecurityDetailsById(Security security)

**Product Controller:**

* GET
* GetAllProducts()
* GetProductByProductId(string email)
* GetProductsByCategoryId(int categoryId)
* POST
* AddProduct(Product product)
* PUT
* EditProduct(Product product)
* DELETE
* DeleteProductById(int productId)

**Category Controller:**

* GET
* GetAllCategories()
* GetCategoryByCategoryId(int categoryId)
* POST
* AddCategory(Category category)
* PUT
* EditCategory(Category category)
* DELETE
* DeleteCategoryById(int categoryId)

**Cart Controller:**

* GET
* GetAllCarts()
* GetCartByCartId(int cartId)
* GetCartByCustomerId(string email)
* POST
* AddToCart(Cart cart)
* PUT
* EditCart(Cart cart)
* DELETE
* DeleteCart(int cartId)

**CartItem Controller:**

* GET
* GetAllCartItemsByCartId(int cartId)
* GetCartItemByCartItemId(int cartItemId)
* POST
* AddCartItem(CartItem cartItem)
* PUT
* EditCartItem(CartItem cartItem)
* DELETE
* DeleteCartItem(int cartItemId)

**Shipping Controller:**

* GET
* GetShippingByShippingId(int shippingId)
* GetShippingByCustomerId(string email)
* POST
* AddShipping(Shipping shipping)
* PUT
* EditShipping(Shipping shipping)
* DELETE
* DeleteShipping(int shippingId)

**Payment Controller**

* GET
* GetPaymentByPaymentId(int paymentId)
* POST
* AddPayment(Payment payment)
* PUT
* EditPayment(Payment payment)
* DELETE
* DeletePayment(int paymentId)

**Order Controller**

* GET
* GetAllOrders()
* GetOrderByOrderId(string orderId)
* GetOrdersByCustomerId(string email)
* POST
* AddOrder(Order order)
* PUT
* EditOrder(Order orderd)
* DELETE
* DeleteOrder(string orderId)

**OrderItem Controller**

* GET
* GetOrderItem(int orderId)
* GetOrderItemsByOrderId(string orderId)
* POST
* AddOrderItem(OrderItem orderItem)
* PUT
* EditOrderItem(OrderItem orderItem)
* DELETE
* DeleteOrderItem(int orderItemId)

**Sample Data:**

To do:

* Store 5 customer detail
* Store 4 category detail
* Store 3 Products on each
* Store 2 Shipping address

**Test Case:**

* Consider srimani has purchased a product via cart mode
* Ordered 2 products with 1 quantity via Cash on delivery mode

**Customer table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Email** | **FirstName** | **LastName** | **ContactNumber** | **Address** | **City** |
| srimani@gmail.com | Srimani | Ravikumar | 9677899682 | Lalgudi | Trichy |
| suresh@gmail.com | Suresh | kumar | 7904954042 | Marakadai | Chennai |
| nirmal@gmail.com | Nirmal | Raj | 6383775155 | Jeeyapuram | Madurai |
| shri@gmail.com | Shri | Dev | 9894785369 | Woraiyur | Karur |
| sanjay@gmail.com | Sanjay | Loganathan | 9487914272 | Kulumani | Theni |

|  |  |  |
| --- | --- | --- |
| **Country** | **ZipCode** | **ProfilePhoto** |
| India | 632123 | D:\MyDocumentaries\Images\ProfilePhoto\Srimani.jpg |
| China | 651231 | D:\MyDocumentaries\Images\ProfilePhoto\Suresh.jpg |
| Japan | 543879 | D:\MyDocumentaries\Images\ProfilePhoto\Nirmal.jpg |
| Thailand | 443789 | D:\MyDocumentaries\Images\ProfilePhoto\Shridev.jpg |
| France | 557908 | D:\MyDocumentaries\Images\ProfilePhoto\Sanjay.jpg |

|  |  |  |  |
| --- | --- | --- | --- |
| **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| srimani@gmail.com | GetDate() | srimani@gmail.com | GetDate() |
| suresh@gmail.com | GetDate() | suresh@gmail.com | GetDate() |
| nirmal@gmail.com | GetDate() | nirmal@gmail.com | GetDate() |
| shti@gmail.com | GetDate() | shti@gmail.com | GetDate() |
| sanjay@gmail.com | GetDate() | sanjay@gmail.com | GetDate() |

**Security table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Email** | **Password** | **SecurityQuestion** | **Answer** |
| srimani@gmail.com | srimani | What is your mother's maiden name? | Shanthi |
| suresh@gmail.com | suresh | What is your favorite movie? | Ayan |
| nirmal@gmail.com | nirmal | What is your favorite vacation spot? | Yercaud |
| shri@gmail.com | shri | What is your favorite food? | Noodles |
| sanjay@gmail.com | sanjay | What is the name of your first school? | Bishop heber |

|  |  |  |  |
| --- | --- | --- | --- |
| **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| srimani@gmail.com | GetDate() | srimani@gmail.com | GetDate() |
| suresh@gmail.com | GetDate() | suresh@gmail.com | GetDate() |
| nirmal@gmail.com | GetDate() | nirmal@gmail.com | GetDate() |
| shri@gmail.com | GetDate() | shri@gmail.com | GetDate() |
| sanjay@gmail.com | GetDate() | sanjay@gmail.com | GetDate() |

**Category table:**

|  |  |  |
| --- | --- | --- |
| **CategoryId** | **CategoryName** | **Description** |
| 1 | Apparel | Include all clothing items related to sports, such as jerseys, uniforms, sports-specific clothing (e.g., soccer shorts, basketball jerseys), compression wear, sports bras, socks, and athletic jackets. |
| 2 | Footwear | Include all types of sports shoes, including cleats (for various sports), tennis shoes, running shoes, golf shoes, hiking boots, ice skates, and rollerblades. |
| 3 | Equipment | Include all sports equipment, such as balls (for different sports), rackets and paddles, bats and clubs, helmets, pads, gloves, goggles, swim caps, and other protective gear. |
| 4 | Accessories | Include sports-related accessories like water bottles, sports bags, sports watches and trackers, nutrition supplements, and any other accessory items. |
| 5 | Nutrition and Hydration | Includes both nutrition supplements and sports drinks, emphasizing the importance of staying hydrated and well-nourished during sports and fitness activities. |

|  |  |  |  |
| --- | --- | --- | --- |
| **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| Admin | GetDate() | Admin | GetDate() |
| Admin | GetDate() | Admin | GetDate() |
| Admin | GetDate() | Admin | GetDate() |
| Admin | GetDate() | Admin | GetDate() |
| Admin | GetDate() | Admin | GetDate() |

**Shipping table:**

* One customer may have more than one shipping address

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ShippingId** | **Address** | **City** | **Country** | **ZipCode** | **Landmark** |
| 1 | Lalgudi | Trichy | India | 632123 | Sivan temple |
| 2 | Marakadai | Chennai | China | 651231 | A2 Police station |
| 3 | Jeeyapuram | Madurai | Japan | 543879 | Medical shop |
| 4 | Woraiyur | Karur | Thailand | 443789 | BC Supermarket |
| 5 | Kulumani | Theni | France | 557908 | St.Joseph’s College |
| 6 | Kingston street | Wahington | United States | 12401 | Fit-X Gym |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BelongsTo** | **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| srimani@gmail.com | srimani@gmail.com | GetDate() | srimani@gmail.com | GetDate() |
| suresh@gmail.com | suresh@gmail.com | GetDate() | suresh@gmail.com | GetDate() |
| nirmal@gmail.com | nirmal@gmail.com | GetDate() | nirmal@gmail.com | GetDate() |
| shri@gmail.com | shri@gmail.com | GetDate() | shri@gmail.com | GetDate() |
| sanjay@gmail.com | sanjay@gmail.com | GetDate() | sanjay@gmail.com | GetDate() |

**Product table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ProductId** | **ProductName** | **ProductImage** | **StockCount** |
| 1 | Nike Men's Dri-FIT Running Shirt | D:\MyDocumentaries\Images\ProductImages\Apparel\Nike Men's Dri-FIT Running Shirt.jpeg | 150 |
| 2 | Under Armour Women's HeatGear Leggings | D:\MyDocumentaries\Images\ProductImages\Apparel\Adidas Men's Dri-FIT Running Track Pants.jpg | 50 |
| 3 | Under Armour Men's Challenger II Training Shorts | D:\MyDocumentaries\Images\ProductImages\Apparel\Under Armour Men's Challenger II Training Shorts.jpg | 200 |
| 4 | Nike Air Zoom Pegasus 38 Running Shoes | D:\MyDocumentaries\Images\ProductImages\Footwear\Nike Air Zoom Pegasus 38 Running Shoes.jpg | 200 |
| 5 | Adidas Men's Ultraboost 21 Running Shoes | D:\MyDocumentaries\Images\ProductImages\Footwear\Adidas Men's Ultraboost 21 Running Shoes.jpg | 200 |
| 6 | Reebok Women's CrossFit Nano X Training Shoes | D:\MyDocumentaries\Images\ProductImages\Footwear\Reebok Women's CrossFit Nano X Training Shoes.jpg | 200 |
| 7 | Spalding NBA Official Basketball | D:\MyDocumentaries\Images\ProductImages\Equipment\Spalding NBA Official Basketball.jpg | 100 |
| 8 | Wilson Pro Staff Tennis Racket | D:\MyDocumentaries\Images\ProductImages\Equipment\Wilson Pro Staff Tennis Racket.jpg | 100 |
| 9 | Everlast Heavy Bag and Boxing Gloves Set | D:\MyDocumentaries\Images\ProductImages\Equipment\Everlast Heavy Bag and Boxing Gloves Set.jpg | 100 |
| 10 | Oakley Radar EV Path Sunglasses | D:\MyDocumentaries\Images\ProductImages\Accessories\Oakley Radar EV Path Sunglasses.png | 2999 |
| 11 | Fitbit Charge 5 Fitness Tracker | D:\MyDocumentaries\Images\ProductImages\Accessories\Fitbit Charge 5 Fitness Tracker.jpg | 500 |
| 12 | Garmin Forerunner 245 Music GPS Watch | D:\MyDocumentaries\Images\ProductImages\Accessories\TX Excellence Yoga Mate.jpg | 6999 |
| 13 | Fast and Up | D:\\MyDocumentaries\\Images\\ProductImages\\Nutrition and Hydration\\Fast and Up.jpg | 500 |
| 14 | Muscleblaze Creatine Monohydrate | D:\MyDocumentaries\Images\ProductImages\Nutrition and Hydration\\Muscleblaze Creatine Monohydrate.jpg | 899 |
| 15 | Asitis Peanut Butter | D:\\MyDocumentaries\\Images\\ProductImages\\Nutrition and Hydration\\Asitis Peanut Butter.jpg | 8766 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Price** | **CategoryId** | **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| 888 | 1 | Admin | GetDate() | Admin | GetDate() |
| 3999 | 1 | Admin | GetDate() | Admin | GetDate() |
| 2111 | 1 | Admin | GetDate() | Admin | GetDate() |
| 5990 | 2 | Admin | GetDate() | Admin | GetDate() |
| 6000 | 2 | Admin | GetDate() | Admin | GetDate() |
| 1499 | 2 | Admin | GetDate() | Admin | GetDate() |
| 500 | 3 | Admin | GetDate() | Admin | GetDate() |
| 900 | 3 | Admin | GetDate() | Admin | GetDate() |
| 100 | 3 | Admin | GetDate() | Admin | GetDate() |
| 40 | 4 | Admin | GetDate() | Admin | GetDate() |
| 100 | 4 | Admin | GetDate() | Admin | GetDate() |
| 100 | 4 | Admin | GetDate() | Admin | GetDate() |
| 700 | 5 | Admin | GetDate() | Admin | GetDate() |
| 7676 | 5 | Admin | GetDate() | Admin | GetDate() |
| 823 | 5 | Admin | GetDate() | Admin | GetDate() |

**Payment table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PaymentId** | **PaymentType** | **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| 1 | Cash on Delivery | Admin | GetDate() | Admin | GetDate() |
| 2 | Credit Card | Admin | GetDate() | Admin | GetDate() |
| 3 | Debit Card | Admin | GetDate() | Admin | GetDate() |
| 4 | Internet Banking | Admin | GetDate() | Admin | GetDate() |
| 5 | UPI | Admin | GetDate() | Admin | GetDate() |
| 6 | Wallet | Admin | GetDate() | Admin | GetDate() |

**Cart table:**

|  |  |  |
| --- | --- | --- |
| **CartId** | **BelongsTo** | **IsEnabled** |
| 1 | srimani@gmail.com | 0 |
| 2 | suresh@gmail.com | 0 |
| 3 | nirmal@gmail.com | 1 |
| 4 | shri@gmail.com | 1 |
| 5 | sanjay@gmail.com | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| srimani@gmail.com | GetDate() | srimani@gmail.com | GetDate() |
| suresh@gmail.com | GetDate() | suresh@gmail.com | GetDate() |
| nirmal@gmail.com | GetDate() | nirmal@gmail.com | GetDate() |
| shri@gmail.com | GetDate() | shri@gmail.com | GetDate() |
| sanjay@gmail.com | GetDate() | sanjay@gmail.com | GetDate() |

**CartItem table:**

* Temporary storage
* Used for smothering the user interface for the separate session/user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CartItemId** | **CartId** | **ProductId** | **Quantity** | **TotalPrice** |
| 1 | 1 | 1 | 1 | 888 |
| 2 | 1 | 2 | 1 | 3999 |
| 3 | 1 | 3 | 2 | 4222 |
| 4 | 2 | 1 | 5 | 29950 |
| 5 | 2 | 2 | 1 | 6000 |

|  |  |  |  |
| --- | --- | --- | --- |
| **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| srimani@gmail.com | GetDate() | srimani@gmail.com | GetDate() |
| suresh@gmail.com | GetDate() | suresh@gmail.com | GetDate() |
| nirmal@gmail.com | GetDate() | nirmal@gmail.com | GetDate() |
| shri@gmail.com | GetDate() | shri@gmail.com | GetDate() |
| sanjay@gmail.com | GetDate() | sanjay@gmail.com | GetDate() |

**Order table:**

* Here first two orders are placed via cart mode
* The third order is placed via buy now mode
* This item handled in the generic collection untill the order confirm

|  |  |  |
| --- | --- | --- |
| **OrderId** | **CustomerId** | **OrderDate** |
| SPRTZN-20230828155812-F594-0001 | srimani@gmail.com | 2023-08-28 15:58:12.357 |
| SPRTZN-20230828155829-F4BF-0002 | suresh@gmail.com | 2023-08-28 15:58:29.727 |
| SPRTZN-20230828155829-F11F-0003 | nirmal@gmail.com | 2023-08-30 15:58:29.727 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Status** | **TotalAmount** | **PaymentId** | **ShippingId** | **CartId** |
| Placed/Cancelled/Delivered | 9109 | 1 | 6 | 1 |
| Placed/Cancelled/Delivered | 35950 | 1 | 2 | 2 |
| Placed/Cancelled/Delivered | 100 | 2 | 5 | Null |

|  |  |  |  |
| --- | --- | --- | --- |
| **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| srimani@gmail.com | GetDate() | srimani@gmail.com | GetDate() |
| suresh@gmail.com | GetDate() | suresh@gmail.com | GetDate() |
| nirmal@gmail.com | GetDate() | suresh@gmail.com | GetDate() |

**OrderItem table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderItemId** | **OrderId** | **ProductId** | **Quantity** | **TotalPrice** |
| 1 | SPRTZN-20230828155812-F594-0001 | 1 | 1 | 888 |
| 2 | SPRTZN-20230828155812-F594-0001 | 2 | 1 | 3999 |
| 3 | SPRTZN-20230828155812-F594-0001 | 3 | 2 | 4222 |
| 4 | SPRTZN-20230828155829-F4BF-0002 | 1 | 5 | 29950 |
| 5 | SPRTZN-20230828155829-F4BF-0002 | 2 | 1 | 6000 |

|  |  |  |  |
| --- | --- | --- | --- |
| **CreatedBy** | **CreatedDate** | **UpdatedBy** | **UpdatedDate** |
| srimani@gmail.com | GetDate() | srimani@gmail.com | GetDate() |
| suresh@gmail.com | GetDate() | suresh@gmail.com | GetDate() |
| nirmal@gmail.com | GetDate() | nirmal@gmail.com | GetDate() |
| shri@gmail.com | GetDate() | shri@gmail.com | GetDate() |
| sanjay@gmail.com | GetDate() | sanjay@gmail.com | GetDate() |

**ER Diagram:**

**Statements:**

**Customer - Security**

* A customer can have single security details, and each secutity details will be associated to single customer.

**Customer - Cart**

* A customer can have more carts but not at same time, but each cart will be taken by single customer.

**Customer - Order**

* A customer will place many orders, but each order will be placed by single customer.

**Customer - Shipping**

* A customer may have many shipping address, each shipping address will belongs to distinct customer

**Product - Category**

* A product will belongs to single category, but each category will have many products.

**Product - CartItem**

* A product will be contained in many cart items, but each cart item should contain single product.

**Cart - CartItem**

* A cart may contain many cart items, but each cart item will belongs to single cart.

**Cart - Order**

* A cart will belongs to distict order, and each order having unique cart with it.

**Order - OrderItem**

* An order contains many ordered items, but each ordered item will belongs to unique order.

**Order - Payment**

* It is an one-way relation from order to payment
* Each order will be placed with unique payment mode.

**Order-Shipping**

* An order will be placed with unique shipping address, a shipping address can be used for multiple orders

**Repository Layer Computation:**

* Calculate the total price of the cart before order to be placed.
* Store the details of the product and quantity in a collection while ordering via **‘BUY NOW’** option. **Done in stored procedure**
* Storing the CartItems into the OrderItem table

**Customer Repository:**

1. GetAllCustomer()
2. GetCustomerByCustomerID(string email)
3. CreateCustomer(Customer customer)
4. UpdateCustomer(Customer customer)
5. DeleteCustomerByCustomerID(string email)

**Security Repository:**

1. GetSecurityDetailsByCustomerID(string email)
2. AddSecurityDetails(Security security)
3. UpdateSecurityDetails(Security security)
4. DeleteSecurityDetailsByCustomerID(string email)

**Category Repository:**

1. GetAllCategories()
2. GetCategoryByCategoryID(int categoryID)
3. AddNewCategory(Category category)
4. UpdateCategory(Category category)
5. DeleteCategoryByCategoryID(int categoryID) // remove products
6. DeleteAllCategories()

**Product Repository:**

1. GetAllProducts()
2. GetAllProductsByCategoryID(int categoryID)
3. AddProduct(Product product)
4. UpdateProduct(Product product)
5. DeleteProductByProductID(int productID)
6. DeleteAllProducts()
7. CalculateTotalAmountByQuantity(int productID,int quantity)
8. UpdateStockCount(int productID,int quantityPurchased)

**Cart Repository:**

1. GetAllCarts()
2. GetAllActiveCarts(bool isEnabled=true)
3. GetAllOrderedCarts(bool isEnabled=false)
4. GetAllCartsByCustomerID(string email)
5. GetAllOrderedCartsByCustomerID(string email,bool isEnabled=false)
6. GetActiveCartByCustomerID(string email,bool isEnabled=true)
7. GetCartByCartID(int cartID)
8. AddNewCart(Cart cart)
9. UpdateCart(Cart cart)
10. DeleteAllCarts()
11. DeleteAllOrderedCarts(bool isEnabled=false)
12. DeleteAllActiveCarts(bool isEnabled=true)
13. DeleteAllCartsByCustomerID(string email)
14. DeleteAllOrderedCartsByCustomerID(string email,isEnabled=false)
15. DeleteActiveCartByCustomerID(string email,isEnabled=true)

**CartItem Repository:**

1. GetAllCartItems()
2. GetAllCartItemsByCartID(int cartID)
3. GetCartItemByCartItemID(int cartItemID)
4. AddNewCartItem(CartItem cartItem)
5. UpdateCartItem(CartItem cartItem)
6. DeleteAllCartItems()
7. DeleteAllCartItemsByCartID(int CartID
8. DeleteCartItemByCartItemID(int cartItemID)
9. EvaluateCartTotal(int cartID)
10. InsertCartItem(int cartID,int productID,int quantity,string email)

**Order Repository:**

1. GetAllOrders()
2. GetAllOrderByCustomerID(string customerID)
3. GetAllOrdersViaCartMode() //check for not null in cartID column
4. GetAllOrdersViaCartModeByCustomerID(string email) //check for not null in cartID column
5. GetAllOrdersViaDirectMode() //check for null in cartID column
6. GetAllOrdersViaDirectModeByCustomerID(string email) //check for null in cartID column
7. GetAllOrdersByPaymentMode(int paymentID)
8. AddNewOrder(Order order)
9. UpdateOrder(Order order)
10. DeleteAllOrders()
11. DeleteAllOrdersByCustomerID(string email)
12. DeleteOrderByOrderID(string orderID)
13. GenerateOrderID()
14. GetOrderSerialNumber()
15. PlaceOrderViaCartMode(int cartID,string customerID,int paymentID,int shippingID)
16. PlaceOrderViaDirectPurchase(string customerID,int productID,int quantity,int paymentID,int shippingID)

**OrderItem Repository:**

1. GetAllOrderedItems()
2. GetAllOrderedItemsByOrderID(string orderID)
3. AddOrderedItemsFromCart(string orderID,int cartID)
4. AddOrderedItem(string orderID,int productID,int quantity)
5. UpdateOrderItem(OrderItem orderedItem)
6. DeleteAllOrderedItems()
7. DeleteAllOrderedItemsByOrderID(string orderID)
8. DeleteOrderedItemByOrderedItemID(int orderItemID)
9. InsertOrderItemsFromCartItems(string email,int cartID)
10. InsertOrderItem(stirng email,int productID,int quantity)

**Payment Repository:**

1. GetAllPaymentModes()
2. GetPaymentModeByPaymentID(int paymentID)
3. AddPaymentMode(Payment payment)
4. UpdatePaymentMode(Payment payment)
5. DeletePaymentMode(int paymentID)
6. DeleteAllPaymentModes()

**Shipping Repository:**

1. GetAllShippingAddresses()
2. GetAllShippingAddressesByCustomerID(string email)
3. GetShippingAddressByShippingID(int shippingID)
4. AddNewShippingAddressToCustomer(string email)
5. UpdateShippingAddress(Shipping shippingl)
6. DeleteAllShippingAddresses()
7. DeleteAllShippingAddressesByCustomerID(string email)
8. DeleteShippingAddressByShippingID(int shippingID)