Project Title: Manga Selling Website

The Manga Selling Website is a web-based application designed to manage manga store operations efficiently. The system is modular and follows the MVC architecture for flexibility and scalability, making it compatible with the ASP.NET Core MVC framework.

Core Modules

Category Management – Handles creation, updates, and management of categories.
Product Management – Handles creation, updates, and management of products.
Order Processing – Facilitates order submission, processing, and tracking.
Customer Management – Manages customer profiles and interactions.
Inventory Management – Manages stock levels and inventory tracking.
User Management – Manages user authentication, authorization, and profiles.
Shopping Cart – Manages the shopping cart for customers.

Assumptions-

The application will be deployed locally during development using a relational database (MySQL).
Security mechanisms will include role-based authentication.
ORM framework (Entity Framework for .NET) will handle database interactions.
No containerization will be used for local deployment.
Module-Level Design

Category Management Module

Purpose: Handles operations for category management.
Controller:
CategoryController
createCategory(categoryData)
updateCategory(categoryId, categoryData)
getCategoryDetails(categoryId)
deleteCategory(categoryId)

Model:
Entity: Category
Attributes:

categoryId (PK)
categoryName (VARCHAR)
displayOrder (INT)

Product Management Module

Purpose: Handles operations for product management.
Controller:
ProductController
createProduct(productData)
updateProduct(productId, productData)
getProductDetails(productId)
deleteProduct(productId)

Model:
Entity: Product
Attributes:
productId (PK)
title (VARCHAR)
author (VARCHAR)
description (TEXT)
ISBN (VARCHAR)
listPrice (DECIMAL)
price (DECIMAL)
price50 (DECIMAL)
price100 (DECIMAL)

Order Processing Module

Purpose: Facilitates order management.
Controller:
OrderController
submitOrder(orderData)
processOrder(orderId, status)
getOrderDetails(orderId)

Model:
Entity: Order
Attributes:
orderId (PK)

customerId (FK)
orderAmount (DECIMAL)
orderStatus (ENUM)
orderDate (DATE)
deliveryDate (DATE)

Customer Management Module

Purpose: Manages customer profiles and data.
Controller:
CustomerController
addCustomer(customerData)
updateCustomer(customerId, customerData)
getCustomerDetails(customerId)

Model:
Entity: Customer
Attributes:
customerId (PK)
name (VARCHAR)
email (VARCHAR)
phone (VARCHAR)
address (TEXT)

Inventory Management Module

Purpose: Manages stock levels and inventory tracking.
Controller:
InventoryController
updateStock(productId, quantity)
getStockDetails(productId)

Model:
Entity: Inventory
Attributes:
inventoryId (PK)
productId (FK)
quantity (INT)

User Management Module

Purpose: Manages authentication and role-based access control.
Controller:
UserController
registerUser(userData)
loginUser(username, password)
getUserProfile(userId)

Model:
Entity: User
Attributes:
userId (PK)
username (VARCHAR)
password (VARCHAR, Encrypted)
role (ENUM)

Shopping Cart Module

Purpose: Manages the shopping cart for customers.
Controller:
ShoppingCartController
addItemToCart(cartItemData)
updateCartItem(cartItemId, cartItemData)
removeItemFromCart(cartItemId)
getCartDetails(customerId)

Model:
Entity: ShoppingCart
Attributes:
cartId (PK)
customerId (FK)
totalAmount (DECIMAL)
Entity: CartItem
Attributes:
cartItemId (PK)
cartId (FK)
productId (FK)
quantity (INT)
price (DECIMAL)

Database Schema Table Definitions

Category Table:

```
CREATE TABLE Category (
  categoryId INT PRIMARY KEY AUTO_INCREMENT,
  categoryName VARCHAR(100),
  displayOrder INT
);
```

Product Table:

```
CREATE TABLE Product (
  productId INT PRIMARY KEY AUTO_INCREMENT,
  title VARCHAR(100),
  author VARCHAR(100),
  description TEXT,
  ISBN VARCHAR(20),
  listPrice DECIMAL(10, 2),
  price DECIMAL(10, 2),
  price50 DECIMAL(10, 2),
  price100 DECIMAL(10, 2)
);
```

Order Table:

```
CREATE TABLE Order (
  orderId INT PRIMARY KEY AUTO_INCREMENT,
  customerId INT,
  orderAmount DECIMAL(10, 2),
  orderStatus ENUM('PENDING', 'SHIPPED', 'DELIVERED', 'CANCELLED'),
  orderDate DATE,
  deliveryDate DATE,
  FOREIGN KEY (customerId) REFERENCES Customer(customerId)
);
```

Customer Table:

```
CREATE TABLE Customer (
  customerId INT PRIMARY KEY AUTO_INCREMENT,
```

```
  name VARCHAR(100),
  email VARCHAR(100),
  phone VARCHAR(15),
  address TEXT
);
```

Inventory Table:

```
CREATE TABLE Inventory (
  inventoryId INT PRIMARY KEY AUTO_INCREMENT,
  productId INT,
  quantity INT,
  FOREIGN KEY (productId) REFERENCES Product(productId)
);
```

User Table:

```
CREATE TABLE User (
  userId INT PRIMARY KEY AUTO_INCREMENT,
  username VARCHAR(50) UNIQUE,
  password VARCHAR(255),
  role ENUM('ADMIN', 'USER')
);
```

ShoppingCart Table:

```
CREATE TABLE ShoppingCart (
  cartId INT PRIMARY KEY AUTO_INCREMENT,
  customerId INT,
  totalAmount DECIMAL(10, 2),
  FOREIGN KEY (customerId) REFERENCES Customer(customerId)
);
```

CartItem Table:

```
CREATE TABLE CartItem (
  cartItemId INT PRIMARY KEY AUTO_INCREMENT,
  cartId INT,
  productId INT,
  quantity INT,
```

```
  price DECIMAL(10, 2),
  FOREIGN KEY (cartId) REFERENCES ShoppingCart(cartId),
  FOREIGN KEY (productId) REFERENCES Product(productId)
);
```

Conclusion

The Manga Selling Website is a comprehensive web-based application designed to streamline manga store operations. It employs a modular architecture following the MVC pattern, ensuring flexibility and scalability. The system includes core modules for managing categories, products, orders, customers, inventory, users, and shopping carts. It leverages ASP.NET Core MVC and Entity Framework for efficient development and database interactions. The database schema is well-defined, supporting robust data management and ensuring referential integrity. Overall, the Manga Selling Website aims to enhance operational efficiency, improve customer experience, and provide secure, role-based access control.