

Design and Development of Online Shopping Website System Based on ASP.NET and SQL Server

Tianyu Lan, Muhang Jiang and Jiarui Li

¹University of Science and Technology Liaoning, Liaoning 114004, China

Abstract

This article describes the design and development process of an online shopping website system based on ASP.NET and SQL Server. The article begins with a requirements analysis of the system, including system feasibility analysis, use case analysis, functionality and feature analysis, and operational environment analysis. Next, the article describes the overall design of the system in detail, including the introduction of system functions, the design of front-end functions, and the database design. Subsequently, the article discusses in depth the detailed design of the system, including the implementation of the system home page, user login and registration module, search product module, detailed product information module, shopping cart module and view history order module.

Keywords

ASP.NET, SQL Server, Online shopping website.

1. INTRODUCTION

With the popularity of the Internet and the rise of e-commerce, more and more people tend to shop online. In order to meet this growing market demand, it has become very important to design and develop an online shopping website system based on ASP.NET and SQL Server. This system aims to provide a convenient, aesthetically pleasing and efficient shopping platform that enables users to easily browse and purchase a wide range of products, as well as providing merchants with a powerful e-commerce tool for managing products, processing orders and interacting with customers.

2. SYSTEM ANALYSIS

2.1. System Functions and Characteristics Analysis

This online shopping website can mainly realise the functions of adding shopping cart of products or purchasing directly from the product detail page, checking the history of orders, logging in and registering, and so on.

Ordinary users have the ability to view the product detail page, search for products, view orders, view the shopping cart, operate the products in the shopping cart and other functions.

According to the user's needs can be seen, ordinary users only need to meet the ability to user login and new user registration function, enter the system can be achieved after the product browsing or purchase, view the shopping cart page and view the history of orders page. Therefore, the design of the system functions include user registration, user login, view orders, product view, product search, direct purchase, add to cart, view cart, order and delete, these functions are specifically designed as follows:

1. User registration

User registration is the first step in the use of online shopping malls, that is to say, you must be registered as a user of the system before you can add a shopping cart in the shopping site, purchase and other operations otherwise you can only view product information.

2. user login

The user logs in by correctly submitting information that matches the user's password stored in the database.

3. Viewing orders

Users can view historical order information after logging in.

4. Merchandise viewing

You can view all the products in the homepage, and view them through the product details page, and you can buy and add products to the shopping cart in the product details page under the login status.

5. Product Search

You can do a fuzzy search by typing the word of the product name or category in the search box.

6. Direct purchases

You can buy directly from the product detail page and have the option to buy several at once.

7. Add to cart

You can add items to your shopping cart on the product details page and can choose to add several at once.

8. View Shopping Cart

You can view the items you have previously added to your shopping cart, and you can choose to add several of those items at once.

9. Placement and deletion of orders

After logging in, users can place orders or delete previously added items in their shopping carts.

3. OVERALL SYSTEM DESIGN

3.1. System Functionality Introduction

The main functional modules required for this online shopping website are frontend functional modules.

The front part is used by ordinary registered users and unregistered members, mainly including user registration, user login, product search, product order enquiry, shopping cart enquiry, you can view the product, purchase and other functional modules.

3.2. System front-end function design

The front-end functional design of the system, including the commodity module users can view the display of commodities, commodities of various types of information, the purchase of goods into the shopping cart after the information in the shopping cart can see the shopping information, editing the product, checkout function; login to register the user can log in to browse the product by registering the member information, and like to put in the shopping cart to buy.

3.2.1 Functional Design of Login and Registration Module

User management module through the ordinary user registration page, fill in the standard registration information, become a user of the online shopping site, use the account number

after successful registration, password can log in and then enter the online shopping site shopping. For this module of data through the user registration after the success of the data added to the database, the user in the login interface to enter the account password to verify whether the login is successful. In order to facilitate the user to log in normally for online shopping.

3.2.2 Functional design of the commodity module

Commodity purchase module function is the main module of the front-end commodity module functional design, which allows users to understand the details of the product, inventory and other information. Users through the understanding of the goods they want to buy added to the shopping cart, shopping at any time at any time to edit the product information for viewing. As an online shopping site for the purchase of goods module design, the total price of the purchased goods can be viewed in the shopping cart. Users in the shopping cart can continue to add goods, but also according to their own needs to delete unwanted goods, can be displayed in real time the total price of the product when the order is how much it is convenient for users to understand the shopping information.

The commodity information module is divided into 3 main sections:

1. Merchandise orders, you can make enquiries about your own historical orders.

2. To browse products, you can click on the product image or product name to enter the product detail page. At the same time, you can input the product name and category you want to search.

3. The shopping cart shows the shopping information of the goods, the user can choose their own shopping cart of the goods they like, you can like the goods and then purchase, you can also delete the goods you do not like, you can edit the goods, modify the number of deletion and other functions, and through the purchase of the goods the unit price of the cumulative calculation of the total price, and display it.

3.3. Database design

3.3.1 Physical structure design

1. User information table, which is used to describe the user's information. As shown in Table 1.

Table 1. User information table (userdata)

Listings	Instructions	Typology	Austerity
Id	User id	int	primary key
username	user ID	nchar (10)	not null
password	user password	nchar (10)	not null
sex	user's gender	nchar (10)	null

2. Commodity information table, which is used to store commodity information. As shown in Table 2.

Table 2. Goods information table (goodsdata)

Listings	Instructions	Typology	Austerity
goodid	Product id	int	primary key
goodsname	trade name	nvarchar(50)	null
goodsimg	Product Image Path	varchar(50)	null
goodsintro	Products	nvarchar(50)	null
goodprice	commodity price	Decimal(18,0)	null
goodstype	Product Type	nvarchar(50)	null
goodcount	Number of commodities	int	null

3.Order Information Table, this table describes all information about the order. As shown in Table 3

Table 3. Order Information Table (T_order)

Listings	Instructions	Typology	Austerity
orderID	Order id	int	primary key
userID	User id	int	not null
goodsID	Product id	int	not null
orderDate	Order Time	datetime	not null
orderStatus	Order Status	int	not null
orderItemPrice	Total Price of Ordered Goods	float	null
orderItemCount	Total number of products ordered	int	null

4.Shopping cart information table, this table describes the information of the products in the shopping cart. As shown in Table 4.

Table 4. Goods Type Information Table (tb_GoodsType)

Listings	Instructions	Typology	Austerity
cartID	Cart id	int	primary key
userID	User id	int	null
goodID	Product id	int	null
count	Number of commodities	int	null

4. DETAILED SYSTEM DESIGN

4.1. System Home Page Implementation

The following figure is the home page of this online shopping site, in the process of writing code using a lot of html knowledge, while in the html page on the basis of the addition of asp.net controls, the fixed dead html page into an interactive front page. The page is written by combining the template page. And reasonable use of the knowledge gained by the application of custom controls, the interface is simple and reasonable layout. As shown in Figure 1.

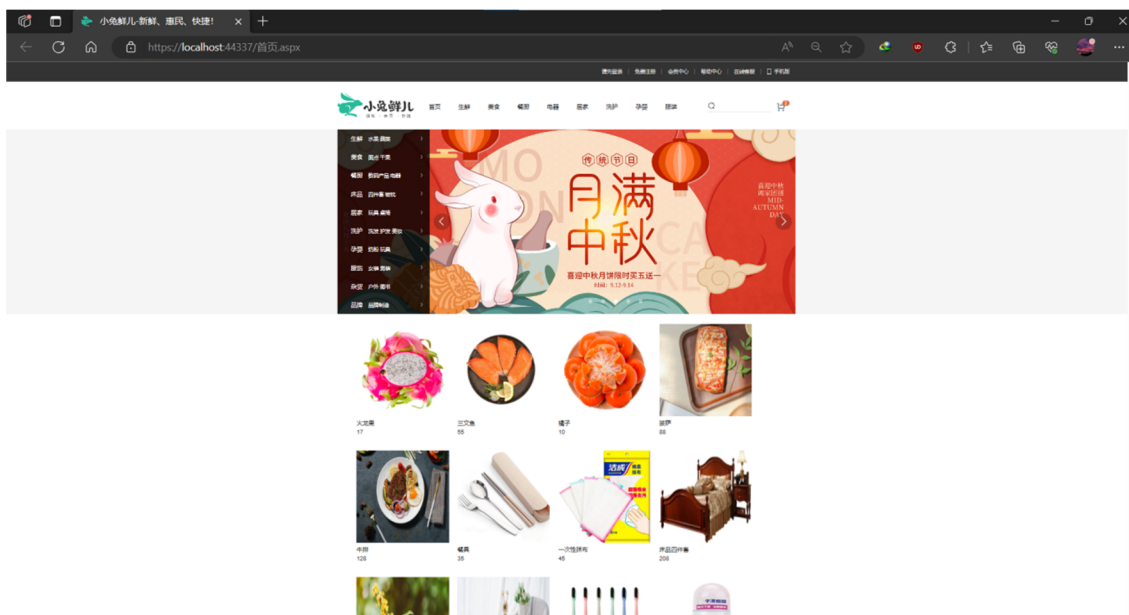


Figure 1. Home Page Interface Diagram

4.2. System Frontend Function Implementation

4.2.1 Product search module

Users can search for products by typing one or several words of product name or product category in the textbox control and then clicking the search button (imagebutton) to perform fuzzy product search. The search box is set in the master page, so you need to pass the input information into the "kind" value first, and then search for the product by getting the "kind" value in the home page. The main code of the search function in the master page is described below:

First, the search terms entered by the user are taken from the text box named txtsearch and stored in a string variable named search.

Use Response.Redirect to pass the user's search term as a query parameter to the parameter named "kind" and redirect to the ". /Home.aspx" page in order to perform the search in the home page.

The code in the first page starts execution by checking if there is a query parameter named "kind". If it exists, the user performs a search and stores the value of the query parameter in a string variable called kind.

Next, construct a SQL query statement to query the database table named goodsdata based on the keywords provided by the user. The query criteria include records whose product name (goodsname) or product type (goodstype) contains the keyword provided by the user.

Use dataoperator.Select(sql) to execute the SQL query and store the result in a SqlDataReader object named dr.

If the query results have rows that match the conditions (i.e., dr.HasRows is true), bind those rows to the data list control named DataList1 to display the search results on the page. Then, close the data reader (dr.Close()) and the database connection (dataoperator.Close()).

If there are no eligible records in the query result, or the user has not performed a search operation (no query parameters), then all product data is queried and bound to DataList1.

Finally, data readers and database connections are closed to ensure that resources are properly freed, regardless of whether a search operation was performed.

The implementation page is shown in Figure 2.



Figure 2. Product Search Page Map

4.2.2 Detailed Product Information Module

Users can click on the home page of the product image or product name to enter the product "details" page goodsdetail.aspx page, the main function of this page is to display the details of the product selected by the user. Which displays the product information mainly through the datalist control to bind the data source for the display of product information, through imagebutton, label, button and other controls to complete the complete display of product information, but also provides a selection of goods to modify the number of buttons, direct purchase and add to cart button, which is the main code is described below.

Get the parameter named "goodsid" from the requested query string and parse it into an integer to be stored in the variable id. Construct a SQL query to select the product records matching id from the database table named goodsdata.

Use `dataoperator.Select(sql)` to execute the SQL query and store the result in a `SqlDataReader` object named `dr`. Use `dr.Read()` to try to read the first row of the query result.

If the query result has rows that match the condition (i.e., `dr.HasRows` is true), then the following actions are performed:

Get information about the product from the data reader, including product image URL (`imgurl`), product name (`name`), product description (`description`), product price (`price`), product quantity (`quantity`).

Set the URL of the product image to the `ImageUrl` property of the `ImageButton1` control to display the product image on the page.

Set the product name, description, price and quantity to the text content of the corresponding text controls (`title`, `miaoshu`, `pricelab`, `number`) respectively.

Finally, close the data reader (`dr.Close()`) and the database connection (`dataoperator.Close()`) to ensure that the resources are released correctly.

The implementation page is shown in Figure 3.

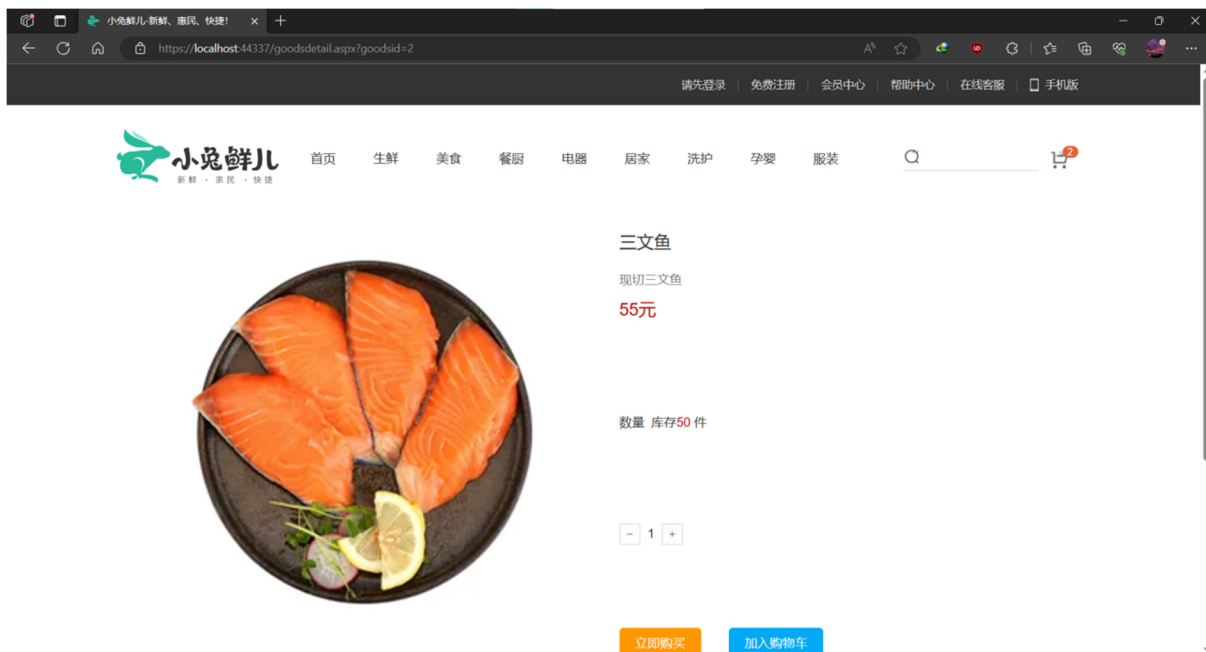


Figure 3. Product details page view

4.2.3 Shopping Cart Module Implementation

Logged in users through the top right corner of the personal home page can click to enter the "shopping cart" `shoppingcart.aspx` page, the main function of this page is to view the shopping cart related information. Users can click `Modify` to modify the number of items purchased, click `delete` to delete the shopping cart record, click `Order` to purchase this product and generate orders, generated orders can be viewed in `My Orders`, shopping cart information through the `datalist` control bound to the data source to display the total amount of `Lable` display, which is the main code is described below.

Get the user's username from the `Session` and store it in a string variable named `user`. Construct a SQL query statement `sqlid` to select the user ID (`id`) that matches the current user by username from the database table named `userdata`. Use `dataoperator.Select(sqlid)` to

execute the SQL query and store the result in a SqlDataReader object named `getid`. Use `getid.Read()` to try to read the first row of the query result.

If `getid.HasRows` is true, it means that a user record matching the current user has been found, and the user's ID (`id`) is converted to an integer and stored in the variable `userid`. If `getid.HasRows` is false, it means that no user record matching the current user is found, and the message "User does not exist" will be output to the page.

Construct a SQL query statement `sql` for selecting the cart items and associated product information that match the user ID (`userid`) from the shopping cart table named `T_cart` and the products data table named `goodsdata`. Close the `getid` data reader and the database connection (`getid.Close()` and `dataoperator.Close()`). Use `dataoperator.Select(sql)` to execute the SQL query and store the result in a SqlDataReader object named `dr`. Check that `dr.HasRows` is true, and if there are shopping cart items and related product information, bind that data to a data list control called `DataList3` to display the cart contents on the page.

Finally, close the `dr` data reader and the database connection (`dr.Close()` and `dataoperator.Close()`).

The implementation page is shown in Figure 4.

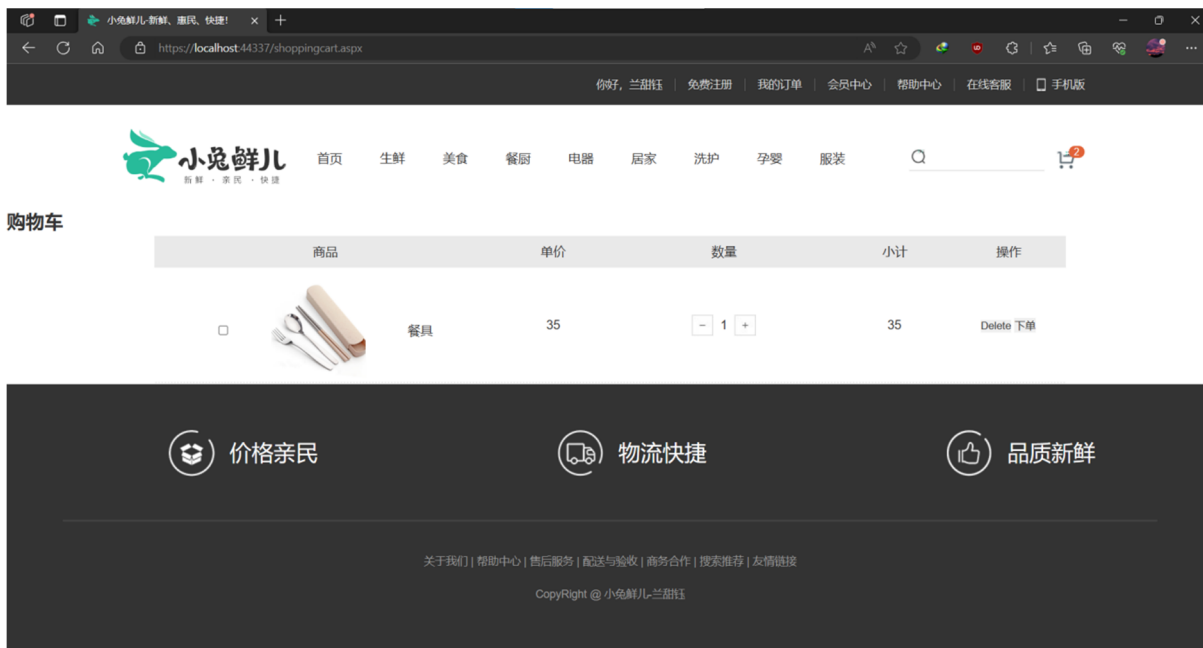


Figure 4. Shopping Cart Page Diagram

4.2.4 View Order History Module

Logged in users through the My Orders to view the history of orders module, the main function of this module is to display all the history of orders, the user can view the home page of the relevant information of each order, such as the total price of the order, the order of goods, the number of goods ordered, the order status, order number, order, order date, etc., the history of the order information through the datalist control binds the data source to display the information of its The main code is described below:

Get the user's username from the Session and store it in a string variable named `user`. Construct a SQL query statement `sqlid` to select the user ID (`id`) that matches the current user by username from the database table named `userdata`. Use `dataoperator.Select(sqlid)` to execute the SQL query and store the result in a SqlDataReader object named `getid`. Use `getid.Read()` to try to read the first row of the query result. If `getid.HasRows` is true, it means that a

user record matching the current user has been found, and the user's ID (id) is converted to an integer and stored in the variable userid.

The getid data reader and the database connection are then closed (getid.Close() and dataoperator.Close()).Construct a SQL query statement select_cartsql to select the order items and associated product information that match the user ID (userid) from the orders table named T_order and the products data table named goodsdata. Use dataoperator.Select(select_cartsql) to execute the SQL query and store the result in a SqlDataReader object named dr. Check that dr.HasRows is true, and if there are order items and related product information, bind this data to a data list control named DataList2 to display the user's order information on the page.

Finally, close the dr data reader and the database connection (dr.Close() and dataoperator.Close()).

The implementation page is shown in Figure 5.

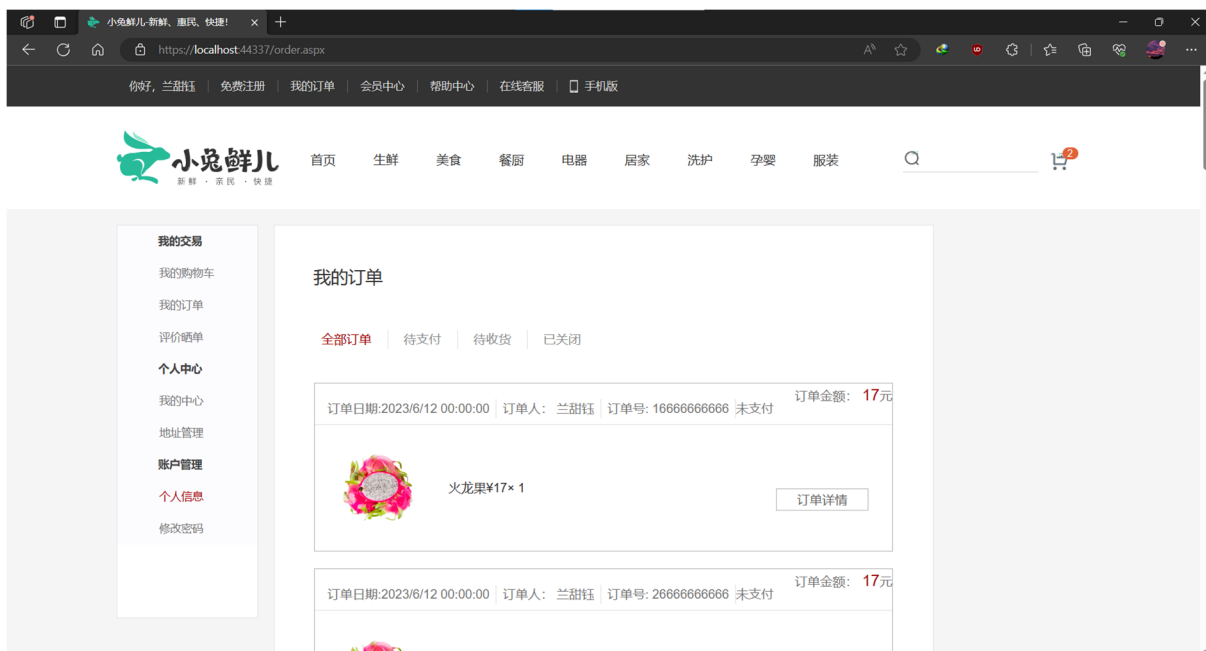


Figure 5. My Orders Page Diagram

5. CONCLUSION

In conclusion, this paper details the design and development process of an online shopping website system based on ASP.NET and SQL Server. Through the system's requirement analysis, overall design and detailed design, we have successfully implemented functional modules such as user login, product search, shopping cart management and historical order view. The system not only has good user interactivity and database management capabilities, but also provides users with a convenient online shopping experience.

Through this project, we have accumulated rich experience in web development, deepened our understanding of ASP.NET and SQL Server, and laid a solid foundation for future web application development. In addition, the system provides a useful example for other developers to learn and learn from for online shopping websites in the e-commerce field. In the growing e-commerce market, such a system is expected to provide more development opportunities for enterprises and more convenience and choices for consumers. Therefore, the online shopping website system presented in this paper has important practical significance and potential business value.

ACKNOWLEDGMENTS

I would like to thank everyone on the team for their help.

REFERENCES

- [1] Song Yafei. "Application of ASP.NET technology in enterprise website design[J]. Wireless Internet Technology,2022,19(04):89-90.
- [2] Chen Xiaotang. Design and implementation of a project management system based on ASP.NET[J]. Microprocessor,2022,43(02):34-37.
- [3] Wu Xiaojing. Design and Implementation of Experimental Teaching Management System for Higher Vocational Colleges and Universities Based on ASP.NET [D]. Guizhou University, 2021.DOI:10.27047/d.cnki.ggudu.2021.000333.
- [4] J. Zhu Xiangyu,Xie Xiaofang,Pan Taotao et al. Development and design of teaching material management system for colleges and universities based on ASP.NET[J]. Computer Knowledge and Technology,2021,17(23):72-74.DOI:10.14004/j.cnki.ckt.2021.2391.