IECE

IECE Transactions on Computer Science
http://dx.doi.org/10.62762/TCS.2024.931558

RESEARCH ARTICLE

Development and Implementation of 'Fresh House Grocery' Fresh Food E-commerce System Based on Django Framework

Yudong Wang[1,*], **Xingdong Wang**[1] **and Yuhua Wang**[1]

[1] College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China

Abstract

With the rapid development of Internet technology, people's shopping methods are gradually shifting from offline to online. As an important branch of this, fresh food e-commerce has gradually become a part of people's daily lives. Traditional offline fresh food markets require a large amount of manpower and material resources for operation and maintenance, which is inefficient and has serious waste of resources. Therefore, it is particularly important to develop an efficient and convenient online fresh food shopping platform. This article introduces a 'Fresh House Grocery' fresh food e-commerce system based on the Django framework. This system aims to provide consumers with a convenient and fast online shopping platform and at the same time improve the work efficiency of management personnel.

Keywords: fresh food e-commerce, python, django, SQLite.

Submitted: 25 October 2024
Accepted: 17 December 2024
Published: 12 January 2025

Vol. 2, **No.** 1, 2025.
10.62762/TCS.2024.931558

***Corresponding author:**
✉ Yudong Wang
ytwyd182@163.com

1 Introduction

With the continuous progress of computer technology, intelligent, information-based lifestyle has gradually penetrated into people's daily lives, and online fresh produce shopping platforms have become a new type of choice [1]. Regarding the purchase channels of fresh products, many years ago, Wang et al. [2] mentioned that most consumers in China usually buy fresh food in farmers' markets, but with the successive implementation of the 'agricultural reform and super' policy and the rapid development of e-commerce in major cities, the penetration rate of department stores and supermarkets and fresh e-commerce has gradually increased. Nowadays, the channels for buying fresh products have formed a diversified pattern. Zheng et al. [3] found that the development of our e-commerce system is extremely rapid, and at the same time gathering is also relatively more concentrated, most of them are gathered in the north, Guangzhou, Shenzhen and other more developed areas, such as Dangdang, Dianping, Box Ma Fresh and so on, these brands in the development of e-commerce platforms have been greatly acclaimed and expectations. Under the merchant-to-consumer model of the fresh food e-commerce system, Kollmann et al. [4] research found that the model integrates physical business with online sales, while becoming a channel for companies to divide work among themselves. Dai et al. [5] research on foreign fresh food e-commerce

Citation
Wang, Y., Wang, X., & Wang, Y. (2025). Development and Implementation of 'Fresh House Grocery' Fresh Food E-commerce System Based on Django Framework. *IECE Transactions on Computer Science*, 2(1), 18–25.

has found that the fresh food supply product chain in some foreign developed countries is very sound, in which the characteristic supplier households include a combination of supply and demand, and these two parts have the advantages of purchasing, billing, transport, traffic and other aspects. Therefore, the cost of the merchant-consumer system model is relatively low, and the range of customers served is much wider. Lingyu et al. [6] for the development of our domestic fresh food e-commerce research, found that although our domestic fresh food products are rich in content and have the advantage of production, but we for the origin of the product, logistics and transport, transport preservation and so on are not quite in place, which largely led to our fresh food e-commerce development has been affected. Zou [7] mentioned that the preservation of fresh food and all aspects of transportation and logistics are extremely important, which is not only difficult, but also extremely costly. Rungani et al. [8] mentioned that it is necessary to strengthen the management of the transportation and preservation of fresh products, so as to reduce the cost waste of merchants, improve the utilization rate of funds, and promote the comprehensive development of e-commerce platforms. At the same time, Liu et al. [9] also said that it is necessary to keep up with the progress of the times, so that the offline fresh market and the online e-commerce platform can be organically combined, so that they can complement each other's advantages and find a bright way forward for the development of China's fresh food industry. Shan et al. [10] said that the development of fresh food e-commerce platforms need to actively try to do a combination of practice and theory, more to explore more practice, and also look to the future.

In the context of reviewing the growing scale of the Chinese fresh food market, the issue of improving consumer shopping experience and management efficiency is particularly urgent. This study develops a fresh food e-commerce system based on the Django framework, which aims to build an integrated large-scale fresh food shopping website management system to meet consumer needs.

2 Technical Introduction and Architecture Deployment

2.1 System technology

The system uses Python as the programming language, Django as the back-end framework, the database management system uses mysql and redis, the use of session objects used to store the user's page cache, while the use of celery distributed task queue asynchronous processing framework to perform some of the tasks, such as sending authentication mail. File storage using fastdfs distributed file storage technology, with nginx used for distributed storage of pictures and other static files, with rapid call, stability and other characteristics. The front-end page is designed and implemented using HTML5, CSS and JavaScript. PyCharm as the integrated development environment provides a wealth of development tools and debugging features, which greatly improves the development efficiency.

2.2 System architecture deployment

Users through the browser for nginx scheduling, divided into dynamic requests and static requests for, if the static request, nginx implements several parts of the function, including forwarding the request to the corresponding application server to achieve load balancing, processing the project's static files, according to the user's request to go to the static page

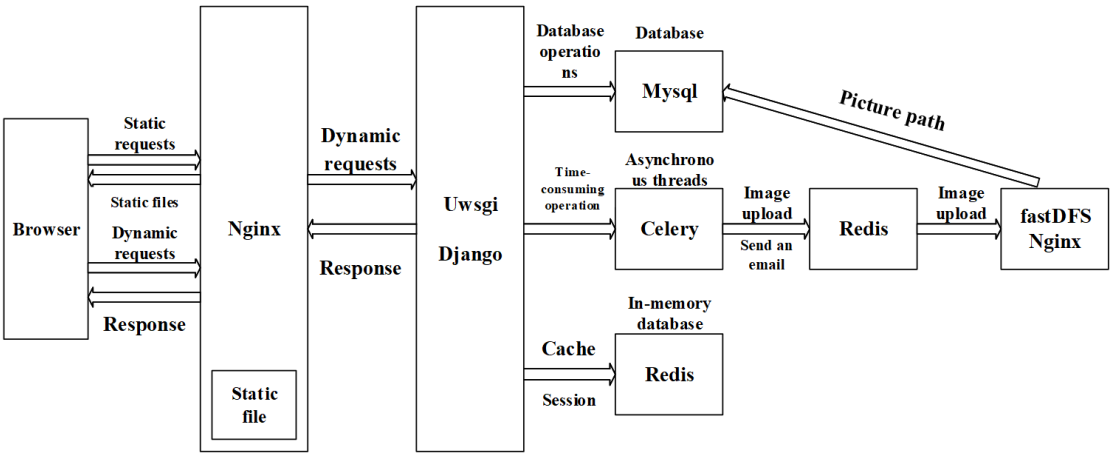

Figure 1. Deployment diagram of system architecture.

servers to find nginx to return to the static home page, so as to answer. For dynamic requests, nginx will convey the command to uWSGI for the application function call, through the django processing and then answer, which in the application server has mysql database, fastdfs distributed file system, redis storage system deployment. The system architecture deployment diagram is shown in Figure 1.

3 Requirements Analysis

3.1 Economic Feasibility

As a country with a large population, food is a major concern for us. People in China have a unique pursuit of delicious food on the tip of their tongues, whether it is the ancient delicacies or today's nutritionally balanced catering, all of which show our unique understanding of food, thus indicating that food is a key concern for us. The development of the fresh food e-commerce system is the offline fresh food shopping market moved to the line, the site through the display of a large number of colourful commodities, as well as detailed commodity profiles, thus attracting the attention of customers to increase the consumer's desire to shop, thus promoting the economic development of the online shopping mall. In addition, we can also maintain cooperation with offline individual users, by providing a large number of fresh products to individual users, to promote the development of online and offline as one.

3.2 Technical Feasibility

The development of the fresh food e-commerce site 'Fresh Food Shop' was done using Django, which includes both front-end and back-end, the front-end needs to meet the functionality requirements of the users and also requires that the users are easy to get started, while the back-end needs to create a complete database and write an administrator's management functions.

The website was developed using Python Web for the front-end and SQLite for the back-end. Using the most popular web production technology Django, combined with HTML code, can quickly complete the site features, without the tedious to compile, can be very simple to write the code, SQLite it takes up very few resources, running speed is very fast, the most important thing is that it is still open source, most of the small and medium-sized web development have chosen SQLite as a web site database. It makes full use of the characteristics of the technology, analyses the technical winds that can occur, explores whether there are technical problems and ways to reduce the risk.

3.3 Operational Feasibility

When constructing the website system, the concept of humanised design is fully integrated to ensure that the design of the operating system can be adapted to the daily use habits of the users and the management needs of the administrators, so as to guarantee the high efficiency of the system operation. In addition, the system is equipped with complete documentation and operation manuals to help users and managers quickly familiarise themselves with the system operation. With the increasing popularity of smart technology, electronic devices have become an indispensable part of people's daily life, especially for convenient access and operation of online shopping malls. Based on this, the ease of operation of this system has significant practical value.

4 System Analysis and Overall Design

4.1 System Function Analysis

The system function analysis mainly divides the system into two modules: the user function module and the administrator function module. Users of the user function module can register, log in, browse products, add to the shopping cart, place orders and pay, query orders and other operations. When registering, you need to fill in your personal details, and after logging in, you can view a personalized user center and manage your profile and shopping history. The shopping cart function makes it easy for users to purchase goods and supports a variety of payment methods to ensure safe and convenient transactions. The administrator of the Administrator function module is responsible for managing user information, product information, order information, and shopping cart information. It can add, delete, modify and check users, update product inventory, price and description in real time, process order status, and provide personalized recommendations based on user cart information to ensure the efficient operation of the mall.

4.2 System Use Case Analysis

Use case analysis can help us understand the main objects of the system and the corresponding functions of these objects, so as to facilitate a clearer understanding of the system construction. The use case diagram is the most indispensable part of the entire website. The use case diagram is mainly to show the users of the software, and the users of the software

who can use those features. As shown in Figure 2.

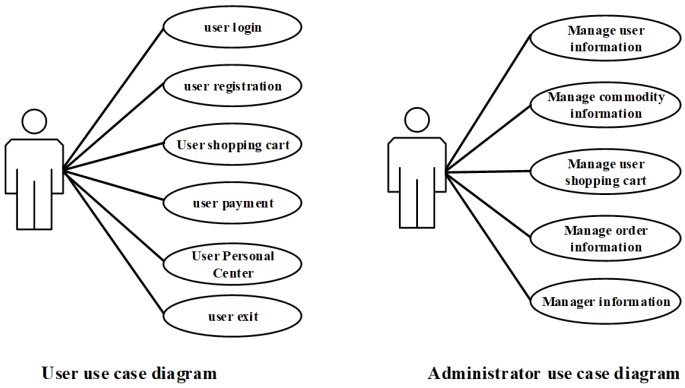

Figure 2. Use case diagram for the system.

Consumers are the most important ingredient in the whole system, and consumers need to register as users to log in to the platform to view products and add them to the shopping cart. Therefore, consumers have more functions, including registration and login, shopping cart, payment, order, user personal information and other information. After the administrator logs in, the main function is to manage user information, product information, user shopping cart information, order information, manager information, etc.

4.3 Overall Design of The System

In the overall design phase, a module hierarchy diagram is used to represent the functional modules of the system and the connections between them. Through the analysis of use cases and the division of functional modules, it is clear that the functions that need to be implemented in this system and the connection between the corresponding module programs. Therefore, for the design of the website function module, the website is divided into two main modules, namely the front-end user module and the back-end management module. In the two theme modules, relevant small modules are added according to user needs, and each small module is linked to each other to form a complete functional module, forming a module design that meets the functional needs of the website. The two main modules should also be connected to each other to ensure the integrity of the system.

The user module is the user-facing part, and the administrator module is responsible for the smooth operation of the entire marketplace, as shown in Figure 3. In addition, database design was carried out, including conceptual structure design, logical

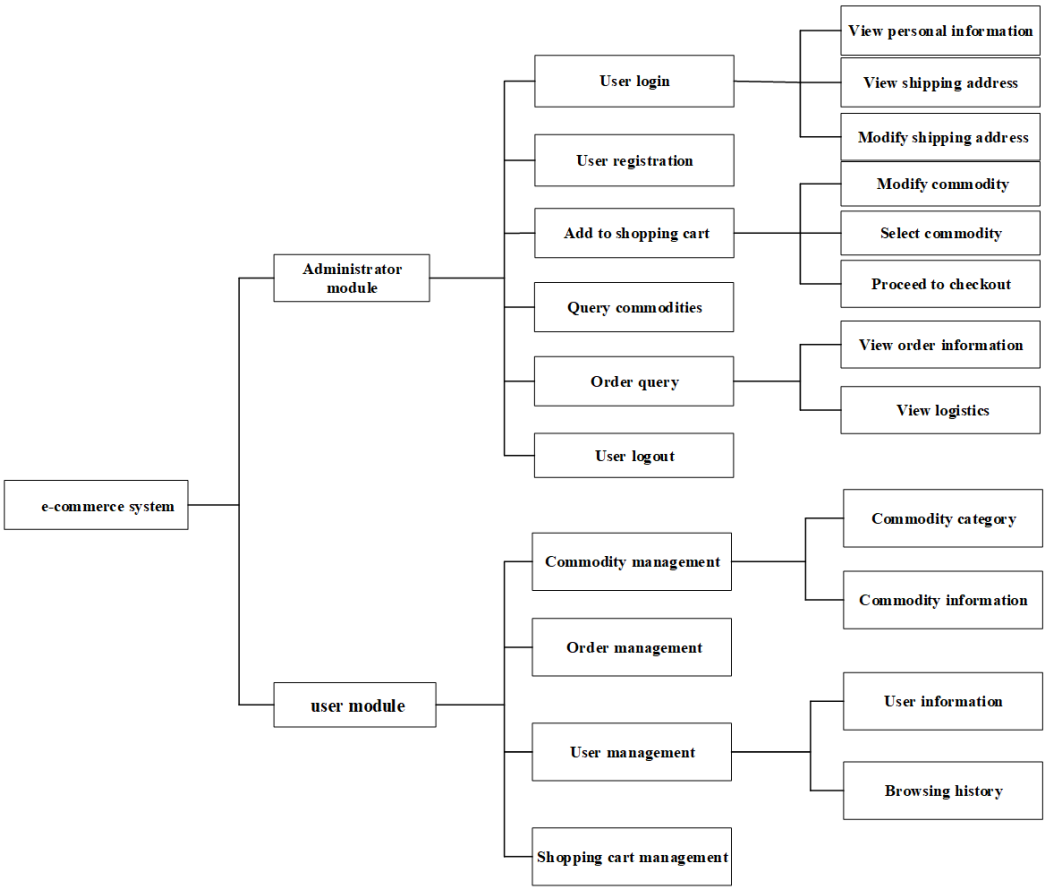

Figure 3. Design of the overall system.

structure design, and physical structure design of the database. The interconnection of websites operates through the interconnection of entity objects, so that the data is linked to each other.

5 System Implementation

In the system implementation stage, according to the requirements of the overall design, each functional module is gradually realized. In the implementation process, the performance and user experience of the system are fully considered to ensure that the system can run stably and meet the needs of users. The website is mainly written using the Django framework, the user requests a connection through the browser, and when the user's URL is obtained, it will match it through the urls.py file, find the corresponding View (view), call the method in the View function, and access the database data through Models, the database here uses the small and medium-sized database SQLite database that comes with Django. After that, the data is returned to the View, and if the views are needed, the HTML page can be generated using the Context passed to the Template. The front-end page of the website realizes the user's login, registration and exit, product inquiry, product information pagination display, shopping cart function, order and other functions. The back-end mainly realizes the administrator background management function, including user management, product information management, shopping cart management, and order management. Figure 4 shows the dependencies of the model relationship.

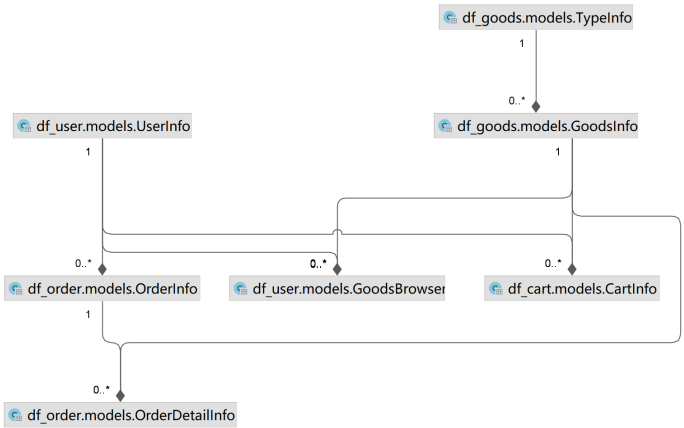

Figure 4. Diagram of model dependencies.

5.1 Front-end User Module Implementation

5.1.1 User Login Module

The user login module uses the login_handle function to submit the form through AJAX, determine the user's password and jump to it, set the cookie value and expiration time, and provide the functions of remembering the password and forgetting the password. The user can log in after successful registration, enter the login page, enter the account password at the time of registration, and then log in, and at the same time have the function of remembering the password and forgetting the password, the main purpose of which is to improve the function of the system to facilitate the user to operate. If the account or password entered is incorrect, a corresponding prompt will appear, and you can enter the password again to log in, as shown in Figure 5.

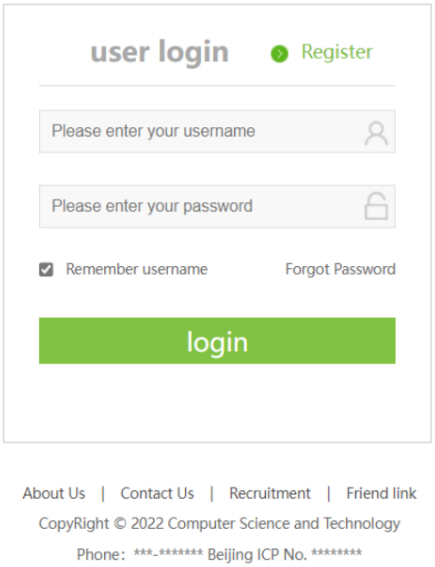

Figure 5. Login page for the user.

5.1.2 Main Page Module

Entering the main page of the website is the theme design of the entire website, and all the front-end functions can be accessed from here. The main page is divided into four parts from top to bottom, namely the user navigation bar, the search section, the product navigation bar and the product display part, and they can carry out different functional operations respectively, as shown in Figure 6 the main page of the website.

5.1.3 Payment Function Module

The payment module uses the order function, enters the cycle time, stores the goods-id in the cart-ids of the shopping cart into the cart for later display to the user through the loop, and also calls the variable cart.count of the user's clicks, so as to calculate the total value of the goods in the shopping cart, and the code formula used is total_price = total_price + float(cart.count) × float(cart.goods.gprice), and finally the function

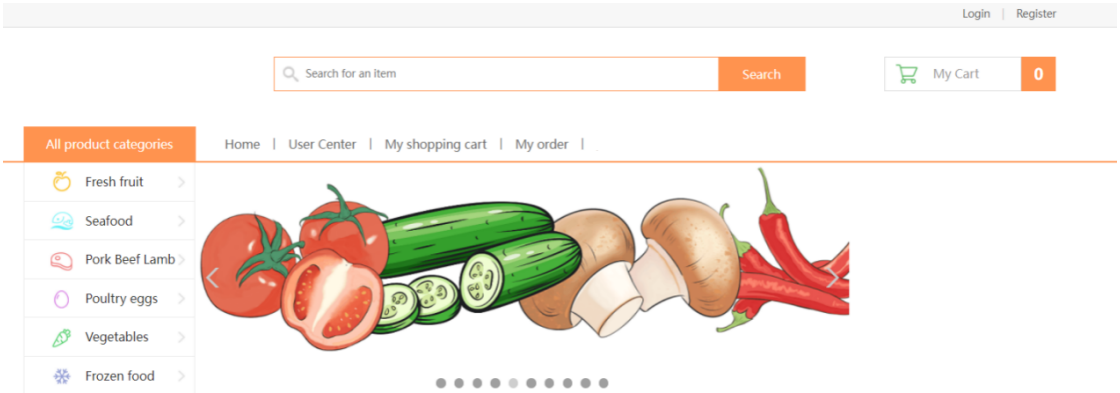

Figure 6. Main page of the site.

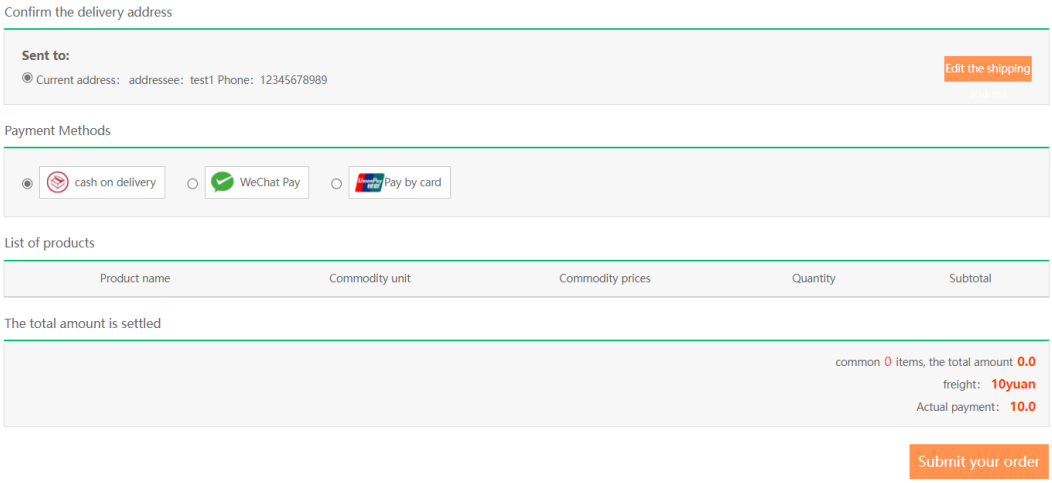

Figure 7. Payment page.

render returns the order information. The payment function is particularly important, including the basic content such as the amount of the product, the user's delivery address, the user's personal information, the payment method, and the product list. Only by filling in the correct delivery address and making the payment, the administrator can receive the order information, send the goods and update the order information. Figure 7 shows the details.

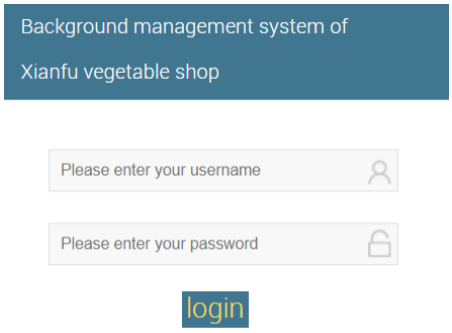

Figure 8. Login page for the administrator.

5.2 Implemented by The Background Management Module

5.2.1 Administrator Login Module

Administrator login we use Django comes with the background management system, in the preparation of the project can be written on the administrator's account password to write an application to create a super-user command python manage.py create superuser, and then enter the appropriate super-user name, as well as the password, the mailbox can be. This is shown in Figure 8.

5.2.2 Merchandise Management Module

An important element of the product management part is to view the number of clicks of each product through the background, through the detail function, when the user clicks on the product, the program will call the request GoodsInfo.objects.get to get the product name, and then through the initial variable of goods.gclick to calculate the number of clicks of each product repeatedly. In the back-end product

management module, administrators can add, modify, and delete products and product types, as well as recommend products. This is shown in Figure 9.

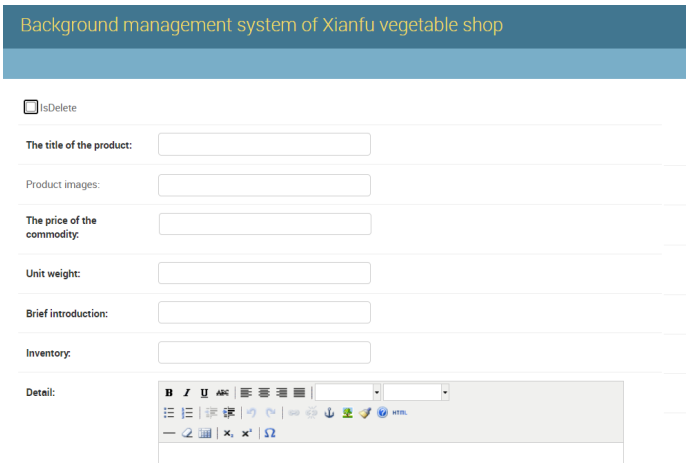

Figure 9. The management page of the product.

5.2.3 *Order Management Module*

The order management module calls @transaction.atomic transaction to manage the order, when the information of the order is obtained, the total price of the order obtained from the front-end is calculated by order_info.ototal, in order to facilitate the customer, the system divides the shopping cart into large and small shopping carts, and the foreign key is associated with order_ detail.order binds small orders to large orders, and finally judges the inventory, compares the purchase amount with the amount of goods.gkucun to judge the inventory, and if any link is wrong, it is determined by transaction.savepoint_rollback (tran_id) to cancel all transactions.

The order management module is divided into two large modules, which include orders and order details, and the orders provided by the user can be managed by the administrator, including the generated order number, user information, time and total price and other information, as shown in Figure 10.

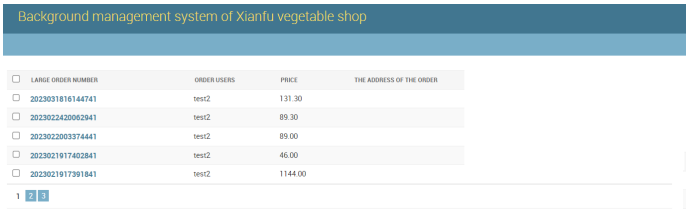

Figure 10. The Order Management page.

6 Conclusion

Through the research and analysis of the existing online fresh food market, it is found that the development of offline fresh food market has entered into a bottleneck, there are many defects that need to be improved, and the development of fresh food enterprises also needs to be further optimised. Therefore, the construction of a high degree of information integration of fresh food website will become the trend of future development. Through the establishment of online fresh food mall, to provide a high-quality and efficient platform for consumers and fresh food merchants, to achieve a close relationship between consumers and merchants, while providing more promotional opportunities for online merchants, in order to promote the development of online shopping mall. On the technical side, the focus should be on optimizing the performance and user experience of the system. Further optimization solutions include considering the use of a distributed file system, storing images and other information in the FastDFS system, and working with Nginx and FastDFS to reduce the pressure on the server. and page data caching, the data used by the page is stored in the cache, and when the data is used again, it is obtained from the cache first, and if it cannot be obtained, it is then queried to the database to reduce the number of queries in the database. Through the optimization of these two aspects, as well as paying attention to the development trend and technological innovation in the e-commerce field, we will explore more application scenarios and functional modules to meet the changing needs of users.

Conflicts of Interest

The author declare no conflicts of interest.

Funding

This work was supported without any funding.

References

[1] Xu, M., Kim, J., Kahhat, R., & Allenby, B. (2008, May). Market dynamics and environmental impacts of e-commerce: A case study on book retailing. In *2008 IEEE International Symposium on Electronics and the Environment* (pp. 1-6). IEEE. [CrossRef]

[2] Wang, H. H., Han, X., Jiang, Y., & Wu, G. (2022). Revealed consumers' preferences for fresh produce attributes in Chinese online markets: A case of domestic and imported apples. *Plos one*, 17(6), e0270257. [CrossRef]

[3] Zheng, Q., Wang, M., & Yang, F. (2021). Optimal channel strategy for a fresh produce E-commerce supply chain. *Sustainability*, 13(11), 6057. [CrossRef]

[4] Kollmann, T., & Hasel, M. (2008). Cross-channel cooperation: on the collaborative integration of online and offline business models of e-entrepreneurs and traditional SMEs. *International Journal of Entrepreneurship and Small Business*, 6(2), 212-229. [CrossRef]

[5] Dai, X., Chen, Y., Zhang, C., He, Y., & Li, J. (2023). Technological revolution in the field: Green development of Chinese agriculture driven by digital information technology (DIT). *Agriculture*, 13(1), 199. [CrossRef]

[6] Lingyu, M., Lauren, C., & Zhijie, D. (2019, May). Strategic development of fresh e-commerce with respect to new retail. In *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)* (pp. 373-378). IEEE. [CrossRef]

[7] Zou, B. (2019). Feasibility Study on Building a Mobile E-commerce platform for Fresh Agricultural Products in China under the Background of Internet Plus. *Ekoloji Dergisi*, (107).

[8] Rungani, E. C., & Potgieter, M. (2018). The impact of financial support on the success of small, medium and micro enterprises in the Eastern Cape province. *Acta Commercii*, 18(1), 1-12. [CrossRef]

[9] Liu, Y., Wang, J., Shao, X., & Li, J. (2018). The current situation and development trend of China's tourism e-commerce. *International Journal of Tourism Sciences*, 18(4), 312-324. [CrossRef]

[10] Shan, H., Bai, D., Fan, X., Shi, J., Li, Y., & Yang, S. (2023). Enabling roles of integration and resilience for sustainable supply Chain performance: an empirical study on China's E-commerce platforms. *Applied Economics*, 55(60), 7079-7093. [CrossRef]

Yudong Wang Currently majoring in electronic Information in College of Information Science and Engineering, Henan University of Technology. (Email: ytwyd182@163.com)

Xingdong Wang Currently working at the School of Information Science and Engineering, Henan University of Technology. His current research interests include remote sensing, GIS(geographic information system). (Email: zkywxd@163.com)

Yuhua Wang urrently working at the School of Information Science and Engineering, Henan University of Technology. Her current research interests include Information security, data analysis. (Email: 9109348@qq.com)