Design and Implementation of A Web Based Ordering System

Jingwen Cao

School of Computer and Software, Jincheng College, Sichuan University, Chengdu 611731, China

Abstract: ***Working staff spend much more time using computers than using mobile phones, making it more convenient to order meals on computers compared to using mobile apps. Therefore, a web-based ordering system has been designed. The system adopts a front-end and back-end separation mode. The front-end adopts Vue.js framework, the server uses Node.js, and the database uses MongoDB. This system can effectively meet the ordering needs of users on web pages. The ordering system has been validated and tested to meet user needs and has practical value.***

Keywords: Vue; Mongo DB; front end; Node.js.

1. INTRODUCTION

The society has entered the Internet era. Under the fast-paced life, take out ordering has become an integral part of people's daily life. In real life, there are very few web-based ordering systems, and a very small number of merchants who provide web-based ordering have ordering systems that are significantly different from user needs, resulting in poor user experience. In the Internet era, computers are the most important tool for work. On the job staff spend much more time using computers than using mobile phones. At the same time, on the job staff are also a group with a greater demand for selling meals. Therefore, it is necessary to develop a web-based ordering platform for the group of people who use computers as common work tools. This platform can effectively solve the problem of inconvenient ordering through computers. This article presents the design and implementation of a ordering system based on the Web and Vue frameworks. Recent advancements in artificial intelligence (AI) and machine learning have significantly influenced various domains, including energy optimization, environmental technology, and urban development. Zheng et al. (2024) proposed a novel GWO-SARIMA-LSTM forecasting model integrated with the TRIZ method for urban building energy optimization, demonstrating its effectiveness in predicting energy consumption patterns [1]. In the field of energy storage, Yin et al. (2024) utilized deep learning for crystal system classification in lithium-ion batteries, highlighting its potential to enhance battery design and performance [2]. Meanwhile, Liu et al. (2024) investigated the impact of supply chain digitization on environmental technology development in G7 nations, emphasizing the roles of inflation and consumption patterns in shaping sustainable practices [3]. In urban planning, Tang et al. (2024) conducted a qualitative analysis of regional housing supply and demand imbalances in the US using big data, providing insights into addressing housing market challenges [4]. In graph theory, Yang et al. (2023) introduced HGMatch, a hyperedge-based approach for subgraph matching, which addresses complexities in hypergraph analysis [5]. Peng et al. (2024) advanced 3D vision-language models with Gaussian splatting, offering innovative solutions for multimodal data representation [6]. In finance, Bi and Lian (2024) demonstrated the application of deep learning in portfolio management, enhancing investment strategies through machine learning models [7]. Ren et al. (2025) developed an IoT-based system for 3D pose estimation and motion optimization in athletes, leveraging C3D and OpenPose to improve sports analytics [8]. Zhou et al. (2024) optimized automated garbage recognition models using ResNet-50 and weakly supervised CNNs, contributing to sustainable urban development [9]. Fan et al. (2025) explored incremental learning for retrieval-augmented generation (RAG) models, enhancing their adaptability in dynamic environments [10]. Xu et al. (2025) designed AI-enhanced tools for cross-cultural game design, facilitating collaborative character conceptualization and sketching [11]. Lastly, Zheng et al. (2024) conducted a comparative study of advanced pre-trained models for named entity recognition, providing valuable insights into natural language processing advancements [12].

2. ANALYSIS OF THE ORDERING SYSTEM

2.1 Requirement analysis

During work hours, people spend much more time using computers than using mobile phones. In response to the widespread use of computers by working staff, this user group requires the ability to directly place orders through web pages, thereby avoiding the inconvenience of switching to mobile ordering. Based on the actual analysis results, the ordering system needs to meet the following three basic requirements:

(1) The computer can be used with a browser installed;

(2) Can meet the basic functional needs of users, such as browsing dish information, adding dishes to shopping carts, and placing orders for purchases;

(3) Considering the usage scenario of the ordering system, the operation of the ordering system should be as simple and intuitive as possible to avoid tedious operations that may delay time.

3. DESIGN OF ORDERING SYSTEM

3.1 The main technologies used in the system

3.1.1 Vue Framework

Vue.js - One of the most popular frameworks in front-end project development. Vue is derived from the French word Vue, which means' see ',' view ', and' window ', which aligns well with the philosophy of Vue.js. By using a bidirectional binding data flow pattern, the problem of developers needing to frequently manipulate DOM has been solved [1].

3.1.2 MongoDB database

The storage of MongoDB database data is similar to JSON's bson format, which can store complex data types. The syntax of MongoDB database shell command operation is very similar to JavaScript. Whether it is shell command operation of MongoDB database or JSON like data storage, these characteristics are very suitable for the development of this system.

3.2 Functional Module Design

According to the analysis, the system functions include five modules: user module, login and registration module, shopping cart module, product module, and settlement module. The product module is the main module of this system, which includes functions such as browsing dish information by category, collecting dishes, and adding dishes to the shopping cart. The functional analysis is shown in Figure 1.

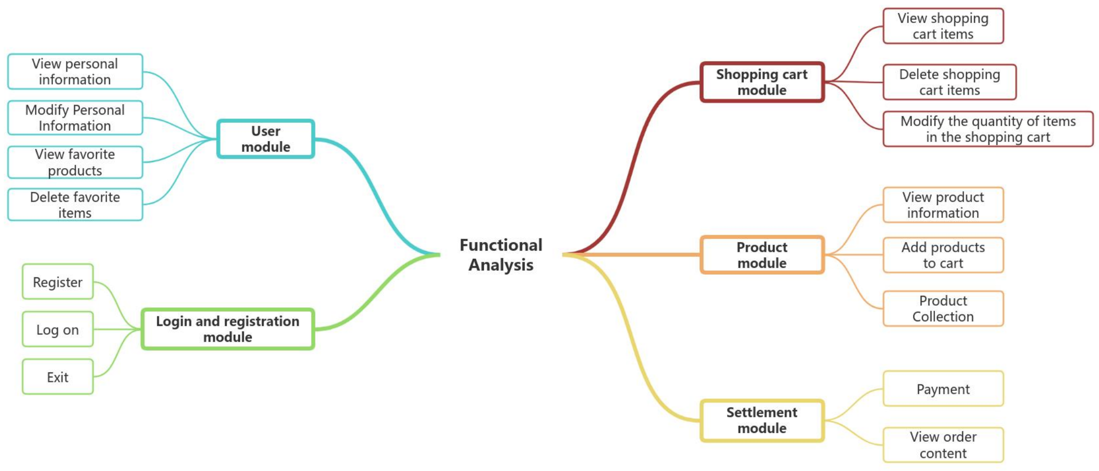


Figure 1: Functional Analysis Module Diagram

3.3 System front-end design

In order to improve the portability of the entire system, the ordering system adopts a front-end and back-end separated system architecture. The core idea of front-end and back-end separation is that the front-end HTML page calls the back-end RESTful API interface through Ajax and interacts through JSON data. The front-end of the ordering system is developed using Visual Studio Code software, mainly designed using the Vue framework and incorporating the Element UI component library. The rapid construction of page structure is completed through the Element UI component library, and the routing router implements page redirection, fully utilizing the Vue.js framework to achieve responsive data binding features through simple APIs. Data is presented on the front-end page through data binding methods such as v-bind.

3.4 System backend design

The backend of this ordering system is developed based on JavaScript language, and the server of the ordering system is written using Node.js+express. Under the premise of front-end and back-end separation, data exchange between the front-end and back-end is carried out through interfaces and JSON format for data transmission, while MongoDB database is used as the database. The design of the backend system mainly uses low-cost technologies to reduce the development cost of the system and make it easier to operate in implementation.

The backend of the ordering system mainly provides the basic operating environment for the system and handles data interaction during user operations. Users can interact with the backend through HTTP requests on the front-end, and the API interface outputs stable and clear JSON data.

3.5 Database Design

The database used is MongoDB. The MongoDB database of this ordering system mainly includes the following collections: users, dishes, shopping carts, and orders.

The user set mainly includes usernames, user passwords, user addresses, ordering phone numbers, etc; The dish collection mainly includes dish names, dish prices, and dish schematics; The shopping cart collection includes dish prices, dish names, dish quantities, and the total amount of dishes. The order collection mainly includes the order username, order contact phone number, order delivery address, and information related to the dishes ordered.

4. IMPLEMENTATION OF ORDERING SYSTEM

4.1 Login and Registration Module

The login and registration module implements user login and user registration functions. Users can register by following these steps:

(1) Enter your phone number and password, with the phone number serving as the default ordering phone number;

(2) Click the "Register" button to submit and trigger the judgment function (btd_zhuce ()) to determine whether the registered content meets the requirements;

(3) The function (btd_zhuce ()) evaluates the input content, and when the password meets the requirement of at least 6 digits, the user registration is successful.

When users log in:

(1) First, enter the registered phone number and password, then click the "Login" button to submit the input content;

(2) Determine whether the input is empty using the btd_login () method;

(3) If it is empty, output the corresponding input prompt information;

(4) If it is not empty, the system will match the information entered by the user with the existing user information in the database. If the match is successful, the user will log in successfully.

4.2 User Module

The user module realizes the basic needs of users for ordering, mainly including the user information interface, user favorites interface, and order query interface. The user information interface provides users with the ability to view and modify their personal information, with the modification function mainly targeting individuals' contact phone numbers and names. The original personal information of this interface is obtained and generated from the database based on the user information filling section in the login module.

After the user logs in successfully, they will be asked to enter information related to ordering, such as name, phone number, delivery address, and gender during delivery. After filling in the required information for name, phone number, and delivery address, they will submit it. After verification, the user can enter the main page to order.

The submission of page information is judged through a method called change ():

(1) If the three items of name, phone number, and delivery address are not empty, the data will be stored in the user information of the backend and routed to the main page of the delivery page;

(2) If any one of the three items, name, phone number, or delivery address, is empty, a prompt message will appear asking for input, and no routing jump will be performed at this time.

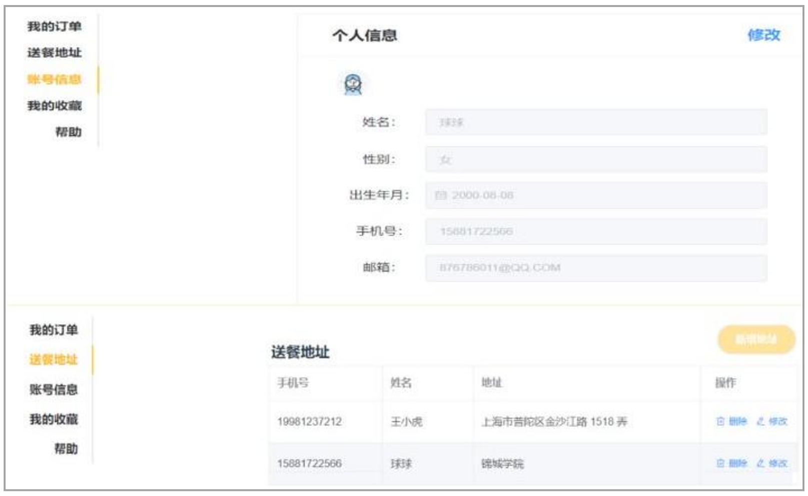
In the user delivery address interface, users can add, delete, and modify their own delivery addresses separately. Users can store more than one address in this section for ordering and selection.

The collection interface is displayed as empty when the user has not performed a collection operation. The interface will only present the corresponding data after the user has performed a collection operation. Figure 2 is a schematic diagram of the user module interface.

The collection module is accessed through:

(1) The user clicks on the dish 'Add to Cart' to trigger the addCart method to save data to the database;

(2) The front-end reads the data related to collections from the database, and displays the dish cards through two steps: data binding and using v-for to render the list.



姓名:	球球
性别:	女
出生年月:	2000-08-08
手机号:	15881722566
邮箱:	876786011@QQ.COM

送餐地址

手机号	姓名	地址	操作
19981237212	王小虎	上海市普陀区金沙江路 1518 弄	删除 修改
15881722566	球球	锦城学院	删除 修改

Figure 2: User Module Interface Diagram

4.3 Product module

The product module is the most frequently browsed module for users, and it is also the main functional module of the ordering system. This module mainly implements the functions of browsing dish information and adding dishes to the shopping cart. The product module utilizes the Element UI component library to present dish cards. The data of dishes is stored in the backend database. The frontend first retrieves the corresponding dish data from the backend data and stores it in a list array. Secondly, Vue.js uses v-bind to bind the data to the HTML tags that need to be bound. Then, it uses v-for to traverse the entire array of data and generate a corresponding number of dish cards based on the number of items in the array.

When adding a shopping cart, users complete the following process:

(1) Click on the event to trigger the addCart method and transfer the corresponding dish information to the shopping cart data in the backend database;

(2) After the data is transmitted to the backend, the add Cart function is used for conditional judgment;

(3) If the added dish name is equal to the existing dish names in the data storage, no duplicate addition will be made, only the quantity of the dish will be modified. Otherwise, the dish will be added to the shopping cart. When the user performs a collection operation, the add Collect method is triggered to determine whether the name is duplicate before adding the corresponding dish data to the user's collection data. If it is duplicate, no new collection will be added, and if it is duplicate with the dish name in the data, no duplicate addition will be made.

4.4 Shopping Cart Module

The shopping cart module data mainly relies on the user's addition of shopping cart operations in the product module. This interface reads the dish data in this section and displays it in the shopping cart. The addition, subtraction, and deletion of the number of items in the shopping cart are implemented on this interface. By using the plus and reduce methods, the obtained dish names are matched with the data in the shopping cart of the database. Clicking on the+and - numbers respectively triggers an increase or decrease in the dish number. When the number of dishes is 1, it is not allowed to reduce the number of dishes. The delete operation triggered by the "delete" button will completely delete the corresponding dish data from the shopping cart.

4.5 Settlement module

The settlement interface is mainly used for visualizing data, which is achieved through the v-bind data binding of the Vue.js framework. On the settlement interface, users can only see their contact name, phone number, order address, settlement amount, and other information, but they cannot modify this information on this interface. The dish data in the settlement interface is obtained from the shopping cart data in the backend to the frontend, and the data binding and data list rendering are generated through Vue.js' v-bind and v-for methods, thereby displaying the specific dishes that the user settles for in the order.

5. CONCLUSION

This system is a web-based ordering system developed using Vue framework, Node.js, and Mongo DB database for research and development. The completed system can meet the expected needs of user groups using computers to order on web pages. According to verification, the system only requires a browser to run, and the web-based functional pages are well implemented. The entire ordering system has low development costs and strong application value.

REFERENCES

[1] Zheng, S., Liu, S., Zhang, Z., Gu, D., Xia, C., Pang, H., & Ampaw, E. M. (2024). Triz method for urban building energy optimization: Gwo-sarima-lstm forecasting model. arXiv preprint arXiv:2410.15283.
[2] Yin, Y., Xu, G., Xie, Y., Luo, Y., Wei, Z., & Li, Z. (2024). Utilizing Deep Learning for Crystal System Classification in Lithium - Ion Batteries. Journal of Theory and Practice of Engineering Science, 4(03), 199–206. https://doi.org/10.53469/jtpes.2024.04(03).19.
[3] Liu, H., Li, N., Zhao, S., Xue, P., Zhu, C., & He, Y. (2024). The impact of supply chain and digitization on the development of environmental technologies: Unveiling the role of inflation and consumption in G7 nations. Energy Economics, 108165.
[4] Tang, Y., Zhao, S., & Yanjun, C. (2024). Regional Housing Supply and Demand Imbalance Qualitative Analysis in US based on Big Data.

[5] Yang, Z., Zhang, W., Lin, X., Zhang, Y., & Li, S. (2023, April). HGMatch: A Match-by-Hyperedge Approach for Subgraph Matching on Hypergraphs. In 2023 IEEE 39th International Conference on Data Engineering (ICDE) (pp. 2063-2076). IEEE.

[6] Peng, Q., Planche, B., Gao, Z., Zheng, M., Choudhuri, A., Chen, T., ... & Wu, Z. (2024). 3d vision-language gaussian splatting. arXiv preprint arXiv:2410.07577.

[7] Bi, S., & Lian, Y. (2024). Advanced portfolio management in finance using deep learning and artificial intelligence techniques: Enhancing investment strategies through machine learning models. Journal of Artificial Intelligence Research, 4(1), 233-298.

[8] Ren, F., Ren, C., & Lyu, T. (2025). Iot-based 3d pose estimation and motion optimization for athletes: Application of c3d and openpose. Alexandria Engineering Journal, 115, 210-221.

[9] Zhou, Y., Wang, Z., Zheng, S., Zhou, L., Dai, L., Luo, H., ... & Sui, M. (2024). Optimization of automated garbage recognition model based on resnet-50 and weakly supervised cnn for sustainable urban development. Alexandria Engineering Journal, 108, 415-427.

[10] Fan, Y., Wang, Y., Liu, L., Tang, X., Sun, N., & Yu, Z. (2025). Research on the Online Update Method for Retrieval-Augmented Generation (RAG) Model with Incremental Learning. arXiv preprint arXiv:2501.07063.

[11] Xu, Y., Shan, X., Lin, Y. S., & Wang, J. (2025). AI-Enhanced Tools for Cross-Cultural Game Design: Supporting Online Character Conceptualization and Collaborative Sketching. In International Conference on Human-Computer Interaction (pp. 429-446). Springer, Cham.

[12] Zheng, Z., Cang, Y., Yang, W., Tian, Q., & Sun, D. (2024). Named entity recognition: A comparative study of advanced pre-trained model. Journal of Computer Technology and Software, 3(5).