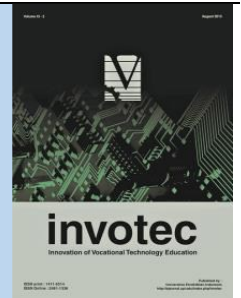




Innovation of Vocational Technology Education

Available online at <http://ejournal.upi.edu/index.php/invotec>



CAKE SALES WEBSITE BASED ON SPRING BOOT AND MYSQL FOR VOCATIONAL STUDENTS

Dhiya Abdurrahman

School of Computer Science and Technology, Nanjing University of Information Science & Technology, China

ARTICLE INFO

Article history:

Received: 12 Apr 2025

Received in revised form: 15 May 2025

Accepted: 27 Jun 2025

Available online: 30 Jun 2025

*Keywords: Cake Order,
MySQL, ReactJS, Spring Boot*

Authors email:

202352200028@nuist.edu.cn

ABSTRACT

The rapid advancement of Internet technology has transformed online shopping into an efficient means of acquiring goods and services. This study aims to design and develop an online cake-ordering platform to enhance the understanding and application of e-commerce systems within vocational education, especially in culinary and information technology programs. The system was developed using ReactJS for the front-end interface, Spring Boot for the back-end services, and MySQL for database management of users and product information. Core e-commerce functionalities were implemented, including user registration and login, product browsing and search, shopping cart integration, order placement, and administrative tools for managing orders and inventory. As part of a project-based learning approach, vocational students were involved in simulating real-world scenarios in digital business development, providing hands-on experience with full-stack web development and system deployment. The platform was designed to offer a smooth and intuitive user experience, supporting efficient order processing for both customers and administrators. The system's use of RESTful architecture and modern web technologies also supports scalability and maintainability, making it suitable for real-world business applications. This study not only validates the chosen technology stack in the context of bakery retail but also highlights its pedagogical impact in vocational education. It fosters student competency in digital entrepreneurship and provides a replicable model for integrating technology-based projects into vocational training programs.

1. Introduction

The global e-commerce industry is undergoing rapid transformation, driven by the widespread adoption of digital technologies and evolving consumer expectations. This trend was further accelerated by global movement restrictions during the COVID-19 pandemic, with e-commerce's share of global retail trade increasing from 14% in 2019 to 17% in 2020 (Bâra, 2023). The shift toward digital shopping channels offers significant benefits—reduced transportation costs, faster transactions, and broader market reach (Kurniawati et al., 2022). One category notably impacted by this shift is cakes and pastries, which were traditionally purchased through physical inspection and tasting. Despite lacking sensory interaction, online cake ordering platforms offer advantages such as product personalization, 24/7 accessibility, delivery convenience, and access to global customer

reviews (Chen, 2021). In countries like the United States and China, internet-based bakery chains (e.g., “21Cake”, “Hollyland”, “Chocolate Magic”) have emerged and expanded successfully by leveraging digital innovation to offer diverse product lines and high-speed delivery (Wang & Li, 2023). These developments highlight the urgency for small and medium enterprises (SMEs) in the bakery industry to adopt effective e-commerce platforms to stay competitive in the digital economy.

Recent studies emphasize the importance of modern frameworks in the development of scalable and efficient e-commerce platforms (Garg & Jain, 2024). Open-source tools like Spring Boot and ReactJS are widely used due to their modularity, flexibility, and active developer communities. Spring Boot supports rapid back-end development by minimizing configuration complexity and enhancing deployment efficiency (Zhou et al., 2022). ReactJS, on the other hand, allows for dynamic and responsive front-end design, promoting real-time interaction without full page reloads, which is essential for user-centered e-commerce applications (Alsaadi & Salah, 2021). In an applied context, Pratama et al. (2021) demonstrated how combining ReactJS with Spring Boot significantly improved system responsiveness and user satisfaction in an online retail prototype. Meanwhile, Singh and Kumar (2020) highlighted the effectiveness of using these frameworks for building single-page applications (SPAs) that provide personalized digital shopping experiences. These studies contribute practical insights into the technical strengths of these tools, although most have centered around general online retail or large-scale systems.

Despite the broad use of open-source frameworks in web application development, current literature lacks in-depth exploration of domain-specific implementations particularly in the context of small and medium-sized enterprises (SMEs) operating within artisanal and vocational sectors such as bakery businesses. Most existing studies focus on general retail systems or large-scale e-commerce platforms, with limited attention to the unique functional requirements of online cake ordering systems, such as real-time product customization, inventory management, and integrated delivery tracking (Lopez & Davis, 2022). Moreover, there is a lack of empirical documentation on the use of modern web development frameworks such as Spring Boot and ReactJS in vocational contexts, especially in culinary-related education and small business training.

To address these gaps, this study aims to design, develop, and evaluate a web-based cake sales system using Spring Boot as the back-end framework and ReactJS as the front-end interface. The purpose of this system is twofold: first, to demonstrate how digital technologies can be applied in a practical and scalable manner to support the operational needs of vocational bakery businesses; and second, to serve as an instructional tool that can be used in vocational education settings. Specifically, the objectives of this research are (1) to build a functional and user-friendly online cake ordering platform that includes key features such as catalog browsing, shopping cart, product customization, and order processing; (2) to showcase the practical implementation of full-stack development tools in the context of vocational IT and culinary education; and (3) to provide a replicable model that supports digital entrepreneurship training in TVET (Technical and Vocational Education and Training) programs. By integrating web development competencies with real-world business applications, this research contributes to enhancing digital literacy, entrepreneurial capacity, and industry relevance in vocational education curricula.

2. Method

This project followed a typical web application development life cycle, including requirements analysis, system design, implementation, and testing. In the requirements phase, three primary user roles for the system were identified (customer, delivery person, administrator), and the functions needed by each were specified. Both functional requirements (specific features the system must provide) and non-functional requirements (performance, security, usability criteria) were defined. Use case modeling was performed to clarify the interactions between each type of user and the system. Figure 2 (Appendix A) in Section 3 summarizes the use case scenarios for the different roles. Based on these requirements, the system architecture was designed as a two-tier web

application separating the front-end client and back-end server. The overall architecture and the E-R diagram of the system were illustrated in Figure 1 and Figure 3 (Appendix A), follows a browser–server model with a RESTful API interface between the ReactJS client and Spring Boot server.

For development and implementation, the project used the Java programming language for the back-end logic and JavaScript/JSP for the front-end interface. The back-end was built with Spring Boot 2.x, selected for its ability to accelerate application setup through auto-configuration and its rich ecosystem supporting common web development tasks. Spring Boot allowed us to embed a web server (Tomcat) and quickly expose REST API endpoints using annotated controllers, reducing the amount of boilerplate code needed for configuring the web infrastructure. Spring Boot was integrated with Spring Data JPA for database access, which simplified data handling by allowing repository interfaces to be defined and letting the framework generate SQL queries for standard operations. The database used is MySQL 8.0, a widely used open-source relational database management system. MySQL was chosen for its reliability and performance in handling web application data, as well as its compatibility with Spring Boot via JPA drivers.

The front-end was developed using ReactJS 17 (a recent stable version at the time of development), along with HTML5/CSS and Bootstrap for styling. ReactJS enables the creation of a modular user interface where components render efficiently based on state changes, providing a smooth user experience (Rawat & Mahajan, 2020). React Router was used to manage navigation between pages (views) on the single-page application, and Axios (a promise-based HTTP client) was utilized to communicate with the back-end API. Development was conducted in an iterative manner. The core features for each role (basic product listing, user authentication, order placement) were first implemented, followed by the progressive addition of enhancements (search function, admin management pages, etc.), testing each component as it was built. The project was managed using an agile approach with short cycles of coding, review, and refinement. All development and testing were completed using a standard PC environment (Intel i5 processor, 16GB RAM) running the Spring Boot server locally and the React development server, with Google Chrome used for browser testing.

3. Result and Discussion

3.1 System Overview and Architecture

After implementation, the cake sales website fulfills the identified requirements, providing a functional platform for online cake shopping. The system's overall architecture is a two-tier architecture comprising a client-side user interface and a server-side application, connected via RESTful web services. Figure 1 (Appendix A) shows the overall structure of the system. The ReactJS front-end constitutes the presentation layer, responsible for displaying pages and capturing user interactions, while the Spring Boot back-end constitutes the business logic and data access layer, handling operations and interacting with the MySQL database. This separation of concerns (frontend vs. backend) follows modern web development best practices and proved effective in creating a scalable, maintainable application. By using REST API calls (JSON over HTTP) to communicate between the layers, the system ensures loose coupling – the front-end does not need to know the details of database operations, and the back-end can serve other clients or scale independently if needed. All user requests (such as logging in, searching products, or placing an order) are sent as API requests to the server, which processes them and returns the results to be rendered by the client.

Use case diagrams for each user role were developed during the design phase to clarify the system's functionality. Figure 2 (Appendix A) presents the consolidated use case diagram for the three main actor roles: customer, delivery person, and administrator. As shown, customers have use cases such as Register, Login, Browse Products, Search Products, Add to Cart, Place Order, and View Orders. Delivery personnel have use cases Login and View Assigned Orders (which includes updating the delivery status of orders). Administrators have use cases Manage Users, Manage

Products, and Manage Orders. These diagrams (Figure 2) helped in validating that all necessary interactions were accounted for. They guided the implementation by highlighting, for instance, that only logged-in customers should be able to place orders, and that an administrator is needed to assign orders to delivery staff. In the implemented system, these role-based permissions are enforced in both the front-end (hiding or disabling unauthorized actions in the UI) and the back-end (verifying user roles before processing certain API requests).

3.2 Functional Features by User Role

For end-users (customers), the website offers a seamless online shopping experience for cakes. When a user first visits the site, they are presented with a homepage that lists available cake products in various categories. Guests (not logged in) can browse all products, view product details, and search for cakes by keyword or category. The product detail page displays information such as the cake's name, description, price, and an image. If a guest user attempts to add an item to their shopping cart or proceed to checkout, the system will prompt them to log in or register an account. New users can create an account by providing a username, email, password, and selecting their role (as customer or delivery person) on the registration page. Once registered, a user can log in with their credentials. Upon successful login as a customer, the user gains access to purchase functionality: they can add desired cakes to a virtual shopping cart. The cart page allows the customer to review selected items, adjust quantities, or remove items. The total price is automatically calculated based on the contents of the cart. When the customer is ready to purchase, they proceed to a checkout page.

In the current implementation, the checkout page collects order details and simulates a payment process (for this prototype, it is a placeholder page without integration to real payment gateways). After "submitting" the payment, the order is created in the system, and the customer is shown an order confirmation with an order ID. The customer can then navigate to the "My Orders" section, which displays a list of all orders they have placed, along with their status (e.g., Processing, Out for Delivery, Delivered). This order history page updates in real time or on refresh to show the latest status of each order. For a new order, the status will initially be Processing/Pending. Later updates to the status (such as when the order is assigned to a delivery person or marked delivered) will be visible here. Through this flow, the system fulfills the core requirement of allowing a customer to select products, place an order, and track that order through completion. The user interface for customers was designed to be intuitive and responsive – for example, users receive feedback messages when an item is added to the cart or when an action requires login – thereby providing a positive user experience. (Sample screenshots of the customer interface for browsing, cart, and order information are provided in Appendix A, Figures 4–8.)

Delivery personnel have a dedicated interface to manage the orders assigned to them. When a user with a delivery role logs into the system, they are taken to a delivery dashboard page. This page lists all cake orders that have been assigned by an administrator to that particular delivery person and are awaiting delivery. For each order, the delivery person can view key details: the order ID, the customer's name and delivery address, contact information, and the list of items to deliver (with quantities). As the delivery person progresses with their deliveries, they are responsible for updating the status of each order on this dashboard. For example, when they start delivering an order, they might update the status to Out for Delivery, and once the order is handed over to the customer, they mark it as Delivered. The system allows the delivery person to input additional details such as the actual delivery date/time or any issues encountered. These updates immediately reflect in the central database, which means the customer's "My Orders" view will also show the updated status. By enabling delivery staff to update order statuses in real time, the website ensures that customers are kept informed and that administrators have oversight of delivery operations. This contributes to transparency and trust in the service. In the tests, the delivery interface successfully allowed drivers to manage multiple orders and perform status updates without confusion. (Appendix

A contains example interface images such as a delivery dashboard screenshot, labeled as Figure 9.)

The administrator role has full control over managing the platform's data and supervising the order fulfillment process. Upon logging in as an administrator, the user is presented with an admin panel that typically has three main management sections: user management, product management, and order management. In user management, the administrator can view a list of all registered users (both customers and delivery personnel). They can perform actions like activating/deactivating a user account or removing a user (for instance, if a user is fraudulent or no longer needed). For the current system, user management is limited to viewing and deletion; if a user is deleted, the system is designed to handle any dependent data (e.g., orders by that user) by either preventing deletion or by cascading the removal safely. However, in practice, deleting users who have existing orders might be restricted for data integrity. In product management, the administrator can add new cake products to the catalog by providing details such as the product name, category, price, description, and an image. They can also edit details of existing products (for example, update the price or stock availability) and remove products that are no longer for sale. Whenever the product list is modified, those changes are immediately available to customers browsing the site (since the front-end fetches the product list from the database on each page load or via refresh).

The admin interface for product management was implemented to include form validations (e.g., ensuring no required field is left blank when adding a product) to maintain data quality. In order management, the administrator can see all orders placed by all customers. Orders can be searched by Order ID if needed, which is useful in a scenario with many orders in the system. The admin can view each order's details and current status. A crucial function here is assigning orders to delivery personnel: when a new order comes in, it initially has no delivery person assigned. The admin selects an available delivery person from a dropdown list and assigns them to the order. Once assigned, the system updates the order's status to indicate it is ready for delivery and records who is responsible. The assigned delivery person will then see this order in their dashboard. Additionally, the admin can mark orders as completed if necessary or update payment status if manual confirmation is needed (in the system, it is assumed that payment is completed online by the customer at checkout, so the orders are automatically marked as paid.).

Through these capabilities, the administrator role closes the loop in the order processing workflow, overseeing everything from user accounts and inventory to the final delivery of orders. The admin functions were tested with multiple orders and users to ensure that all operations behave correctly and data consistency is maintained. For instance, the system prevents an order from being assigned to more than one delivery person at a time, and it restricts actions that could corrupt data (such as deleting a product that is part of an active order). Figures 10–12 in Appendix A provide sample visuals of the admin dashboard interfaces for user, product, and order management.

Overall, the implemented website achieves the intended functionality for all three user roles. Customers can conveniently register and order cakes online, delivery staff can efficiently handle their delivery tasks, and the administrator can manage the platform's content and operations. The successful completion of these features demonstrates that the chosen technology stack was appropriate for the task. Notably, the use of Spring Boot for the back-end and ReactJS for the front-end was effective in meeting the system requirements. The front-end React components provided a responsive and interactive user experience, while the back-end Spring Boot services reliably handled business logic and data transactions. The integration between the two via RESTful APIs functioned smoothly, indicating that decoupling the front-end and back-end was a sound decision. This result aligns with the trend in web development towards single-page applications (SPAs) consuming RESTful services, which is known to improve scalability and maintainability of web systems (as each side can be developed and scaled independently).

3.3 Performance, Scalability and Security Considerations

Beyond functional correctness, the system was evaluated against several non-functional requirements, including performance, scalability, usability, and security. In terms of performance, the website was observed to be responsive during testing. Typical page loads and operations (such as querying the product list or submitting an order) complete within a couple of seconds on a modest server configuration, which is acceptable for a prototype. Multiple simultaneous users performing different actions (e.g., several customers browsing products and placing orders while the administrator assigns deliveries) were simulated to assess how the system handles concurrency. The Spring Boot back-end, running on a local server, was able to handle concurrent requests without errors or significant slowdowns. For a large-scale deployment, further performance testing and optimization would be needed, but the results are promising for the targeted scope (a single store's online sales). The scalability of the system is supported by its architecture: because the client and server are separate, The server could be scaled vertically (more resources on one machine) or horizontally (multiple server instances behind a load balancer) without changing the front end. The stateless nature of REST APIs means each request is independent, which also aids in scaling out. Moreover, adding new features or product categories is straightforward due to the modular design – for example, the system can be extended to include promotional discounts or additional product attributes without overhauling the existing structure. In the current implementation, if the number of products grows or new types of items are added, the same API and front-end components can accommodate them by design, indicating good maintainability and scalability for future growth.

The usability and interface design were important considerations given the vocational education context of this project. A modern, responsive web design was adopted using the Bootstrap framework alongside custom CSS, ensuring the site is mobile-friendly and works on various screen sizes. The layout and color scheme remain consistent across pages to provide a unified user experience. Clarity and simplicity were prioritized in the UI – navigation menus are straightforward, and important actions (like “Add to Cart” or “Checkout”) are clearly indicated by buttons. During user testing sessions, participants found the interface intuitive, requiring minimal instruction to use the main features. This positive feedback suggests the website meets its usability goals, making it suitable for end-users who may not be tech-savvy.

In terms of security, the application follows standard web security practices appropriate for an e-commerce prototype. The system uses role-based access control: certain actions and pages are protected and only accessible to authenticated users of the correct role. For example, an administrator page cannot be accessed by a customer account (the back-end checks the user's role on each admin API call), and if a non-logged-in user attempts to access a restricted page, they are redirected to the login page. Passwords are stored in the database in hashed form for protection (using a one-way hashing algorithm via Spring Security's utilities).

3.4 Comparison with Related Works and Discussions

The results of this project can be viewed in light of other similar systems and existing literature on web development frameworks. The approach demonstrates that open-source frameworks and a RESTful architecture can be utilized to rapidly develop a functional e-commerce platform for a specific domain (in this case, cake sales). This finding is consistent with the experiences reported by other researchers and developers. For instance, Wu (2022) developed an e-commerce management system using Spring Boot for the back-end and a JavaScript-based front-end (Vue.js in that case), achieving a stable platform for managing orders and inventory (Wu, 2022). The success of that system and ours highlights the versatility of Spring Boot for a range of e-commerce applications. Another study by Tennakoon et al. (2022) presented “Cake-Hut,” a novel online cake ordering platform built with a different technology stack (Node.js for the server and MongoDB for the database) (Tennakoon et al., 2022). Despite the differences in implementation, Cake-Hut provides comparable functionality to the system – allowing customers to browse and order cakes online and

supporting administrative oversight – which underscores that there are multiple viable approaches to implementing such applications. The Spring Boot/MySQL solution achieves similar outcomes to the Node.js/MongoDB solution by Tennakoon et al., reinforcing the idea that modern web frameworks, whether using Java or JavaScript on the server side, can effectively serve the needs of small-scale e-commerce.

One advantage of the implementation is the clear separation of concerns through an API-centric design, which is a practice increasingly adopted in the industry. By using ReactJS to consume a REST API provided by Spring Boot, a smooth user experience was created, paving the way for future enhancements (such as creating a mobile app or integrating third-party services) without needing to redesign the core system. This aligns with the concept of decoupled front-end/back-end architectures discussed in recent web engineering literature, where such architectures facilitate easier maintenance and scaling. The findings of this study also reflect the benefits noted by Hubli & Jaiswal (2023) regarding Spring Boot's productivity and integration capabilities – the framework's built-in support for data access and security allowed us to focus on application logic rather than boilerplate code (Hubli & Jaiswal, 2023).

Through the development of this project, a few areas that present challenges or opportunities for further improvement were identified. For example, implementing real online payment integration was beyond the initial scope, so the current system uses a placeholder payment page. In a production scenario, integrating with secure payment gateways (such as Stripe or PayPal) would be essential. This involves additional security considerations (handling payment credentials, SSL certificates, etc.) and backend workflow to confirm payment transactions. Similarly, the implemented search feature is basic, supporting only exact matches on product name or category. More advanced search capabilities could improve user experience – such as fuzzy search (to handle typos), filtering by price range or popularity, and sorting options. These features have been explored in other e-commerce projects and could be integrated using available libraries or search engine services. Another potential enhancement is adding real-time interactivity, like a chat support widget or live order status updates using web sockets, to further engage users; this could draw on technologies like Spring Boot's WebSocket support or third-party services.

Despite these limitations, the developed website meets the core objectives and provides a solid foundation for a fully featured online cake shop. It demonstrates the feasibility of bringing a traditional business (cake retail) into the online realm using contemporary web development tools. From an educational perspective, this project shows how knowledge of software engineering and web technologies can be applied to solve real-world problems in the field of vocational technology education. Students and practitioners can learn from this case study that by using a well-chosen set of frameworks, even complex-sounding applications like an e-commerce system can be built within a limited timeframe. With further refinement and testing (particularly focusing on security hardening and performance optimization), this platform could be deployed in a real production environment. It could also be extended with a companion mobile application (since the backend APIs could be reused), thereby expanding its reach and convenience.

In summary, the results confirm that the combination of Spring Boot, ReactJS, and MySQL is a powerful and effective stack for developing e-commerce web applications. This work contributes to the growing body of evidence that modern full-stack development frameworks allow for rapid development without sacrificing scalability or user experience. The successful implementation of the system within a limited scope illustrates the potential of current web technologies in rapidly developing niche-market e-commerce solutions. Future developers can build upon this project, whether to create similar online ordering systems for other products or to enhance the system with additional features and integrations, supporting the ongoing innovation in vocational and consumer-oriented web applications.

Suggestions for further development in a vocational context include integrating real payment gateways to simulate actual business transactions, implementing advanced product filtering to reflect real e-commerce systems, and extending the platform with mobile access or real-time features like live chat and notifications. These additions could deepen students' exposure to professional digital practices and enhance curriculum relevance. From an educational perspective, this project makes a significant contribution to vocational learning by bridging web development competencies with business and customer service skills. It provides students with practical exposure to software engineering workflows, role-based system design, and user experience principles that are highly relevant to modern vocational careers.

Future iterations of this project could be integrated into vocational curricula as a capstone project or module-based training resource. Instructors may use the system to teach project management, UI/UX design, and full-stack development using open-source tools. Additionally, the platform could be extended to include collaborative features such as student group development tasks, performance analytics, or instructor assessment dashboards to simulate workplace scenarios in digital entrepreneurship and retail industries.

4. Conclusion

The rapid advancement of Internet technology has transformed online shopping into an efficient means of purchasing goods and services. This study presents the design and implementation of an online cake-ordering platform using a ReactJS front end, Spring Boot back end, and MySQL database. The system offers integrated e-commerce functionalities, including user registration and login, product catalog browsing, shopping cart management, order placement, and administrative features for inventory and logistics control. The architecture adopts a RESTful API approach, allowing a clear separation between front-end and back-end operations, thus enhancing scalability and maintainability. The project was developed as part of a vocational education initiative, providing students in culinary and information technology programs with hands-on experience in developing a digital business platform. Open-source tools were chosen to reduce development costs while ensuring flexibility and upgrade potential through community support. Evaluation of the system confirmed that it meets functional requirements: customers can order cakes online with ease, while administrators manage products and orders efficiently. The platform also serves as a model for implementing digital entrepreneurship projects in vocational education, reinforcing students' competencies in applied web development and business process management. In conclusion, this project contributes not only to digital transformation in the bakery retail sector but also to vocational education through real-world, project-based learning. Future development may include integration with secure payment gateways, AI-powered product recommendations, and enhanced user interaction features. Embedding this platform into vocational training modules could further enrich teaching practices and support the development of job-ready digital skills among students.

Conflicts of Interest

The authors declare no conflict of interest regarding the publication of the paper.

References

- Alsaadi, H., & Salah, A. (2021). Building modern front-end interfaces using ReactJS: A study on user performance and satisfaction. *International Journal of Human-Computer Interaction*, 37(15), 1487–1496.
- Bâra, A. (2023). Digital transformation in global retailing: A post-pandemic acceleration. *Journal of Retail and Consumer Services*, 71, 103192.
- Chen. (2021). Online consumer preferences for bakery products: The impact of service quality and user experience. *Journal of Foodservice Business Research*, 24(6), 565–584.
- Garg, V., & Jain, A. (2024). Scalable Data Integration Techniques for Multi-Retailer E-Commerce Platforms. *International Journal of Computer Science and Engineering*, 13(2), 525-570.

- Hu, M., Wen, C., & Huang, C. (2020). Design and development of graduation thesis management system based on Spring Boot. *Computer Programming Skills and Maintenance*, 2020(7), 92–93.
- Hubli, S. C., & Jaiswal, D. R. C. (2023). Efficient Backend Development with Spring Boot: A Comprehensive Overview. In *International Journal for Research in Applied Science and Engineering Technology* (Vol. 11, Issue 11, pp. 759–765).
- Kurniawati, R., et al. (2022). Adoption of digital platforms by SMEs during COVID-19: Case from food and beverage industry. *International Journal of Information Management Data Insights*, 2(2), 100073.
- Lopez, J., & Davis, E. (2022). Customizing e-commerce platforms for SMEs in the food sector: A review of current practices and limitations. *Journal of Small Business and Enterprise Development*, 29(5), 819–835.
- Pratama, A., et al. (2021). Integration of Spring Boot and ReactJS in web-based commerce applications: A usability evaluation. *Journal of Information Systems Engineering and Management*, 6(4), em0182.
- Rawat, P., & Mahajan, A. N. (2020). ReactJS: A Modern Web Development Framework. In *International Journal of Innovative Science and Research Technology* (Vol. 5, Issue 11, pp. 698–702).
- Singh, R., & Kumar, M. (2020). Developing personalized web applications using single-page architecture and component-based frameworks. *International Journal of Web Engineering and Technology*, 15(3), 223–238.
- Tennakoon, R.P.C. Sankalpani, K.A.V. Prabuddhi, D.S. Balage, D.I. De Silva, & Dulanji Cooray. (2022). Cake-Hut: A Novel E-Commerce Platform for Booking and Purchasing Cakes Online. In *International Journal of Engineering and Management Research* (Vol. 12, Issue 5, pp. 129–134).
- Wu, H. (2022). Commerce Middle Office Management System Based on Springboot. In *International Journal of Advanced Network, Monitoring and Controls* (Vol. 7, Issue 2, pp. 32–45). *International Journal of Advanced Network, Monitoring and Controls*.
- Wang, Y., & Li, T. (2023). Digital bakery business models in China: Case studies and competitive strategies. *Asia Pacific Journal of Marketing and Logistics*, 35(1), 134–152.
- Zhou, X., Li, Y., & Wang, H. (2022). A scalable e-commerce framework using Spring Boot and RESTful APIs. *Journal of Systems and Software*, 186, 111229.

Tables

The following tables present the structured data schema used in the development of the Cake Sales Website. These database tables define the core entities, relationships, and attributes that support the functionality of the system, including user management, product cataloging, order processing, and address handling. Each table is normalized to ensure data consistency and integrity throughout the application.

Table 1. User Information Table

Serial number	The name of the field	type	Primary/foreign keys	Field explanations
1	id	int	Primary key	User ID
2	email_id	varchar(255)	-	Mailbox
3	first_name	varchar(255)	-	Name
4	last_name	varchar(50)	-	Name
5	password	varchar(255)	-	Password
6	phone_no	varchar(255)	-	Telephone Number
7	role	varchar(255)	-	User roles
8	address_id	varchar(255)	-	Address

Table 1 stores user account information, including email, hashed passwords, user roles, and registration dates. It supports role-based access to distinguish between customers, administrators, and delivery personnel.

Table 2. Product Information Table

serial number	The name of the field	type	Primary/foreign keys	Field explanations
1	id	int AI PK	Primary key	Auto-increment ID
2	description	varchar(255)	-	Product Description
3	image_name	varchar(255)	-	Picture title
4	price	decimal(19,2)	-	Price
5	quantity	varchar(255)	-	Quantity
6	title	varchar(255)	-	Product Title
7	category_id	int	-	Category ID

Table 2 contains data related to the cakes being sold, including product names, categories, prices, descriptions, and image URLs.

Table 3. Order Table

serial number	The name of the field	type	Primary/foreign n keys	Field explanations
1	id	int AI PK	Primary key	Auto-increment ID
2	delivery_assigned	varchar(255)	-	Distribute deliveries
3	delivery_date	varchar(255)	-	Delivery date

4	delivery_person_id	int	foreign key	Delivery person ID
5	delivery_statuses	varchar(255)	-	Delivery status
6	delivery_time	varchar(255)	-	Delivery time
7	order_date	varchar(255)	-	Order time
8	order_id	varchar(255)	foreign key	Order ID
9	quantity	int	-	Quantity
10	product_id	int	foreign key	Product ID

Table 3 tracks orders placed by customers, including order ID, customer ID, order status, total amount, and assigned delivery personnel.

Table 4 Product Category Information Table

serial number	The name of the field	type	Primary/foreign keys	Field explanations
1	id	int AI PK	Primary key	Auto-increment ID
2	description	varchar(255)	-	Description
3	title	varchar(255)	-	Product Title

Table 4 defines the available product categories in the system, allowing products to be grouped and searched efficiently.

Table 5 Shopping Cart Table

serial number	The name of the field	type	Primary/foreign keys	Field explanations
1	id	int AI PK	Primary key	Auto-increment ID
2	quantity	int	-	Quantity
3	product_id	int	foreign key	Product ID
4	user_id	int	foreign key	User ID

Table 5 manages temporary cart data, storing the relationship between users and the products they intend to purchase before checkout.

Table 6 Address Information Table

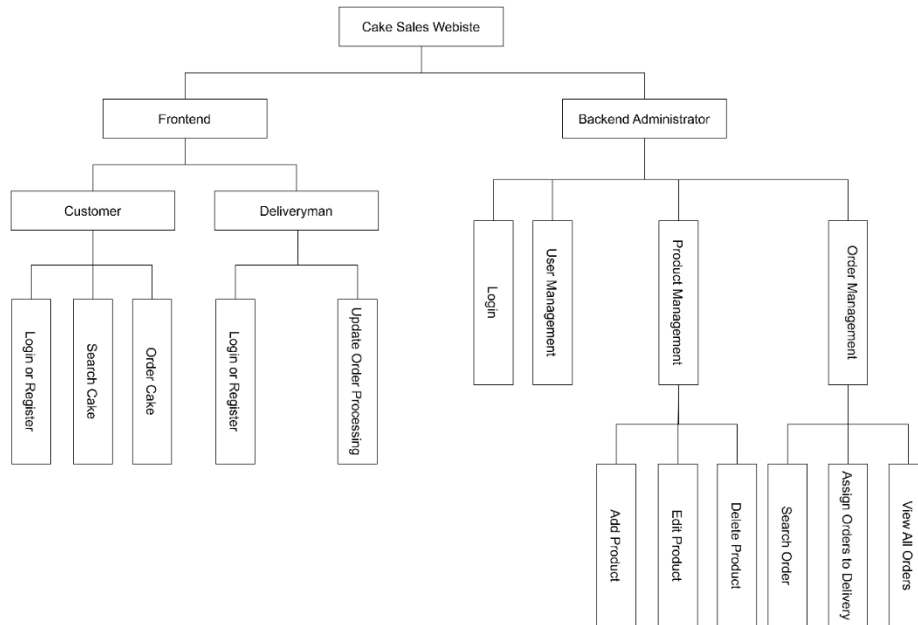
serial number	The name of the field	type	Primary/foreign keys	Field explanations
1	id	int AI PK	Primary key	Auto-increment ID
2	city	varchar(255)	-	City
3	pincode	int	-	Zip code
4	street	varchar(255)	-	Street

Table 6 contains user-provided shipping address details, including city, street, and postal codes, used during the order fulfillment process.

Appendix A

The following figures illustrate the core architectural and functional design of the Cake Sales Website. These diagrams were referenced in the main text to support the explanation of system architecture, data modeling, and user interaction design.

Figure 1. The overall structure of the system



This diagram presents the two-tier architecture separating the front-end (ReactJS) from the back-end (Spring Boot), with data persistence handled by a MySQL database.

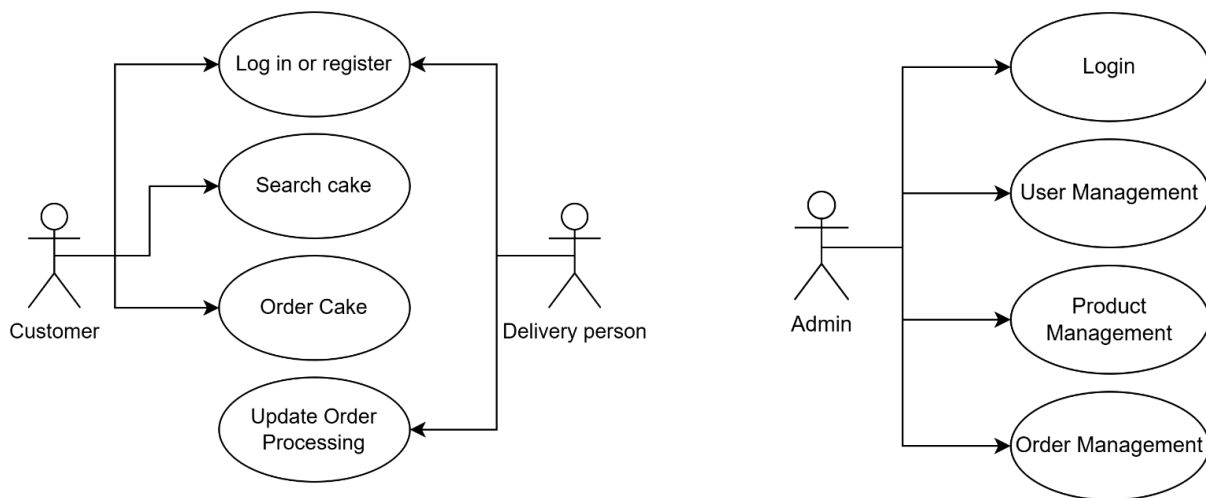


Figure 2. Use Case Diagram

This diagram depicts the interactions between system actors (customer, delivery personnel, administrator) and the available use cases for each role.

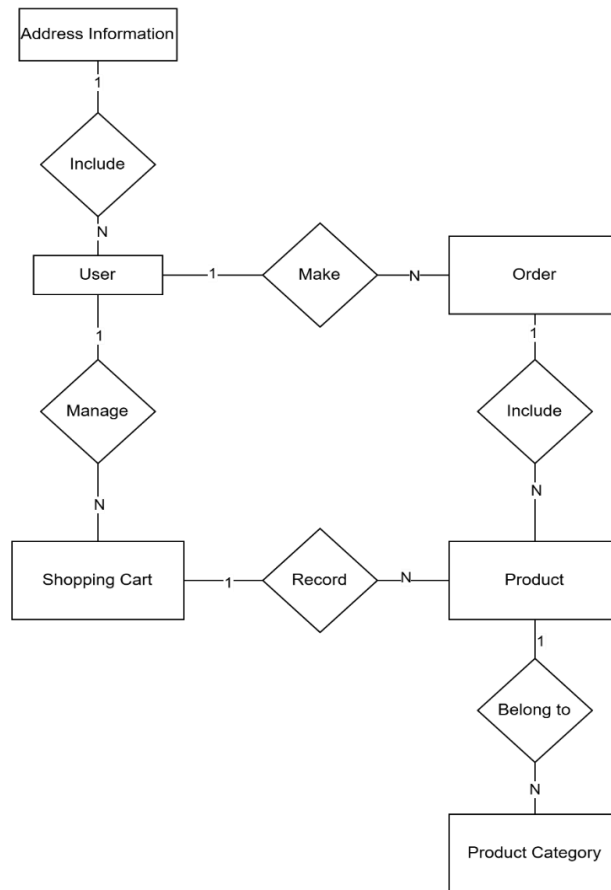


Figure 3. Overall E-R diagram of the system

This diagram outlines the relationships between key entities in the database, such as Users, Products, Orders, and Addresses.

The following figures demonstrate the implemented user interface of the Cake Sales Website. These screenshots provide a visual representation of the user experience and operational workflow for each system role. Sample screenshots of the customer interface for browsing, cart, and order information are provided in Appendix A, Figures 4–8. Figures 9–12 provide sample visuals of the admin dashboard interfaces for user, product, and order management.

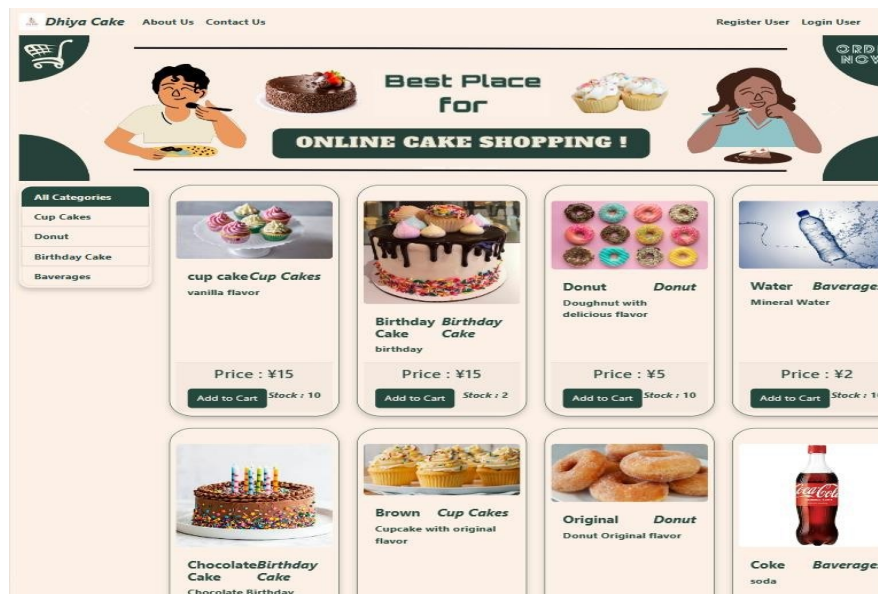


Figure 4. Homepage Interface

Displays available cake products and allows users to browse and search without logging in.

The figure consists of three screenshots of the Dhiya Cake web application interface, showing user registration and login forms.

Top Screenshot: Add User Form

The top screenshot shows the 'Add User' form. The form is titled 'Add User' and is located in the center of the page. The form fields are:

- User Role (Dropdown menu with 'Select Role' selected)
- First Name (Text input field)
- Last Name (Text input field)
- Email Id (Text input field)
- Password (Text input field)
- Mobile No (Text input field)
- Street (Text input field)
- City (Text input field)
- Pincode (Text input field)

A 'Register User' button is located at the bottom of the form. The page header includes the Dhiya Cake logo, 'About Us', 'Contact Us', 'Register User', and 'Login User' links.

Middle Screenshot: User Login Form

The middle screenshot shows the 'User Login' form. The form is titled 'User Login' and is located in the center of the page. The form fields are:

- User Role (Dropdown menu with 'Select Role' selected)
- Email Id (Text input field)
- Password (Text input field)

A 'Login' button is located at the bottom of the form. The page header includes the Dhiya Cake logo, 'About Us', 'Contact Us', 'Register User', and 'Login User' links.

Bottom Screenshot: User Login Form (Expanded)

The bottom screenshot shows the 'User Login' form with the 'User Role' dropdown menu expanded. The dropdown menu lists the following roles:

- Select Role
- Admin
- Customer
- Delivery Person

The form fields and the 'Login' button are also visible. The page header includes the Dhiya Cake logo, 'About Us', 'Contact Us', 'Register User', and 'Login User' links.

Figure 5. User Registration and Login Interfaces

Shows the registration and login forms, including role selection and validation.

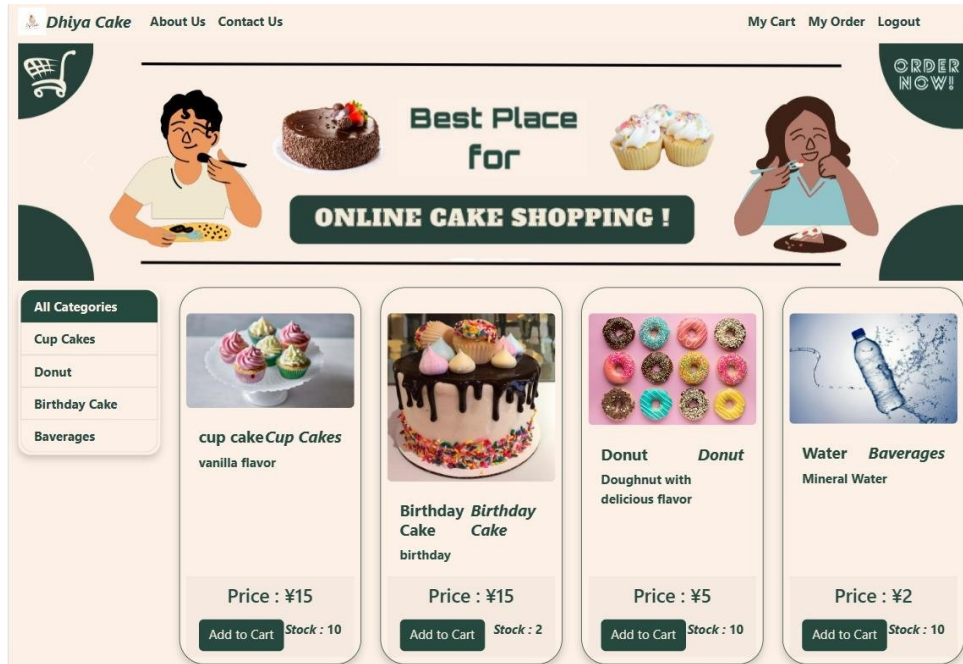


Figure 6. Customer Homepage

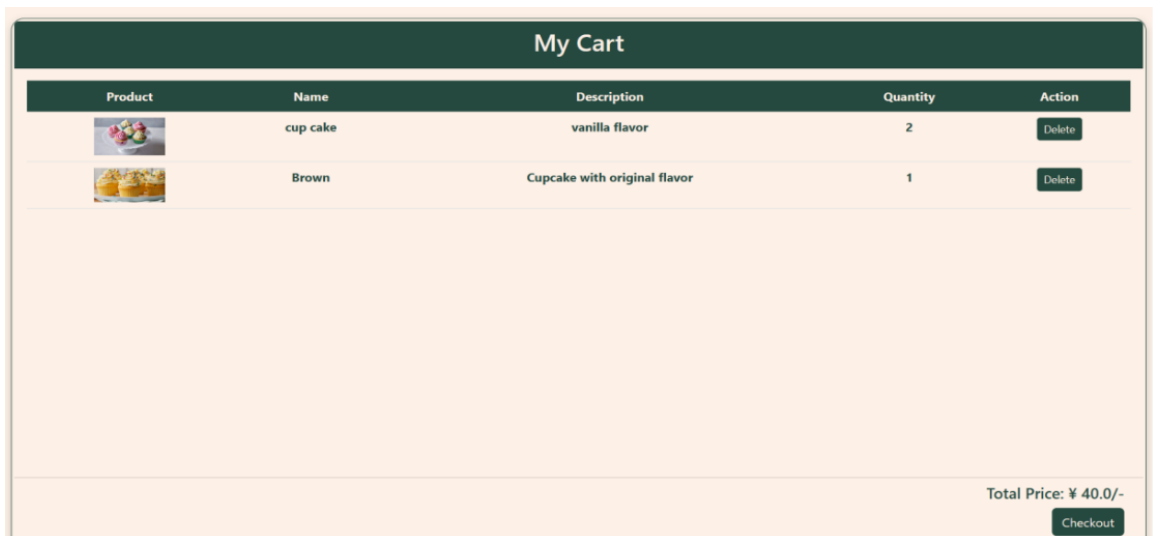


Figure 7. Shopping Cart and Checkout Interfaces

Illustrates how customers manage their selected items and proceed to place an order.



Dhiya Cake

About Us

Contact Us

My Cart

My Order

Logout

My Orders

Order Id	Product	Name	Description	Quantity	Total Price	Order Date	Delivery Date	Delivery Status	Delivery Person	Delivery Mobile No
ZP6MTHRACM		cup cake	vanilla flavor	1	15.0	09-04-2023 10:58	Pending	Pending	Pending	Pending
DZQRXEYXFE		Coke	soda	1	5.0	13-04-2023 16:27	2023-05-12 Afternoon	Delivered	ryan	13232534553
FZVOCZQGLK		Chocolate Cake	Chocolate Birthday cake	1	25.0	13-04-2023 16:36	2023-04-14 Afternoon	Delivered	ryan	13232534553
GHCJOP4APT		cup cake	vanilla flavor	2	30.0	26-04-2023 13:41	Pending	Pending	Pending	Pending
GHCJOP4APT		Brown	Cupcake with original flavor	1	10.0	26-04-2023 13:41	Pending	Pending	Pending	Pending

Figure 8. Customer Order History Page
Presents a list of orders placed by the customer, including status information.



Dhiya Cake

About Us

Contact Us

My Deliveries

Update Order Delivery

Logout

My Deliveries

Order Id	Product	Name	Description	Quantity	Total Price	Customer Name	Street	City	Pin code	Mobile No.	Order Date	Delivery Date	Delivery Status	Delivery Person	Delivery Mobile No
ZP6MTHRACM		cup cake	vanilla flavor	1	15.0	david pierre	Nanjing	Nanjing	12000	12345567789	09-04-2023 10:58	Pending	Pending	ryan	13232534553
DZQRXEYXFE		Coke	soda	1	5.0	david pierre	Nanjing	Nanjing	12000	12345567789	13-04-2023 16:27	2023-05-12 Afternoon	Delivered	ryan	13232534553
FZVOCZQGLK		Chocolate Cake	Chocolate Birthday cake	1	25.0	david pierre	Nanjing	Nanjing	12000	12345567789	13-04-2023 16:36	2023-04-14 Afternoon	Delivered	ryan	13232534553

Search Customer Orders

Order Id	Product	Name	Description	Quantity	Total Price	Customer Name	Street	City	Pin code	Mobile No.	Order Date	Delivery Date	Delivery Status	Delivery Person	Delivery Mobile No
ZP6MTHRACM		cup cake	vanilla flavor	1	15.0	david pierre	Nanjing	Nanjing	12000	12345567789	09-04-2023 10:58	Pending	Pending	ryan	13232534553

Update Delivery Status

Order Id

Select Delivery Date

Delivery Time

Delivery Status

Update Delivery Status

ZP6MTHRACM

yyyy/mm/日

Select Delivery Time

Select Delivery Status

Figure 9. Delivery Dashboard
Shows how delivery staff manage assigned orders and update delivery status.

Dhiya Cake
About Us
Contact Us

Admin Home
Add Category
Add Product
All User
All Orders
Assign Order Delivery
Logout

Welcome, Administrator!

All Categories
Cup Cakes
Donut
Birthday Cake
Beverages



cup cake
vanilla flavor

Cup Cakes

Price : ¥15

Remove Edit Stock : 10



Birthday Cake
birthday

Birthday Cake

Price : ¥15

Remove Edit Stock : 2



Donut
Doughnut with delicious flavor

Donut

Price : ¥5

Remove Edit Stock : 10



Water
Mineral Water

Beverages

Price : ¥2

Remove Edit Stock : 10

Figure 10. Administrator Main Dashboard
Provides the central interface for admin-level users to manage users, products, and orders.

Dhiya Cake
About Us
Contact Us

Admin Home
Add Category
Add Product
All User
All Orders
Assign Order Delivery
Logout

All Users						
User Id	First Name	Last Name	Email	Phone No	Role	Action
2	david	pierre	david.pierre@gmail.com	12345567789	Customer	<button>Remove User</button>
3	ryan	achmad	ryan.achmad@gmail.com	13232534553	Delivery	<button>Remove User</button>
4	Dhiya	Abdurrahman	Admin@gmail.com	13179536298	Admin	<button>Remove User</button>

Figure 11. User Management Interface
Allows the administrator to view, edit, or remove registered users.

Add Category

Category Title

enter title..

Category Description

enter description..

Add Category

Add Product

Product Title

Product Description

Category

Select Category

Product Quantity

Product Price

Select Product Image

选择文件

未选择文件

Add Product

Add Product

Product Title

Product Description

Category

Select Category

Select Category

Cup Cakes

Donut

Birthday Cake

Baverages

Select Product Image

选择文件

未选择文件

Add Product

Dhiya Cake About Us Contact Us Admin Home Add Category Add Product All User All Orders Assign Order Delivery Logout						
Product Id	Image	Title	Category	Description	Price	Quantity
9		cup cake	Cup Cakes	vanilla flavor	¥ 15	10
Update Product Title Enter new title... Description Enter new description... Price Enter new price... Quantity Enter new quantity... Update Product						

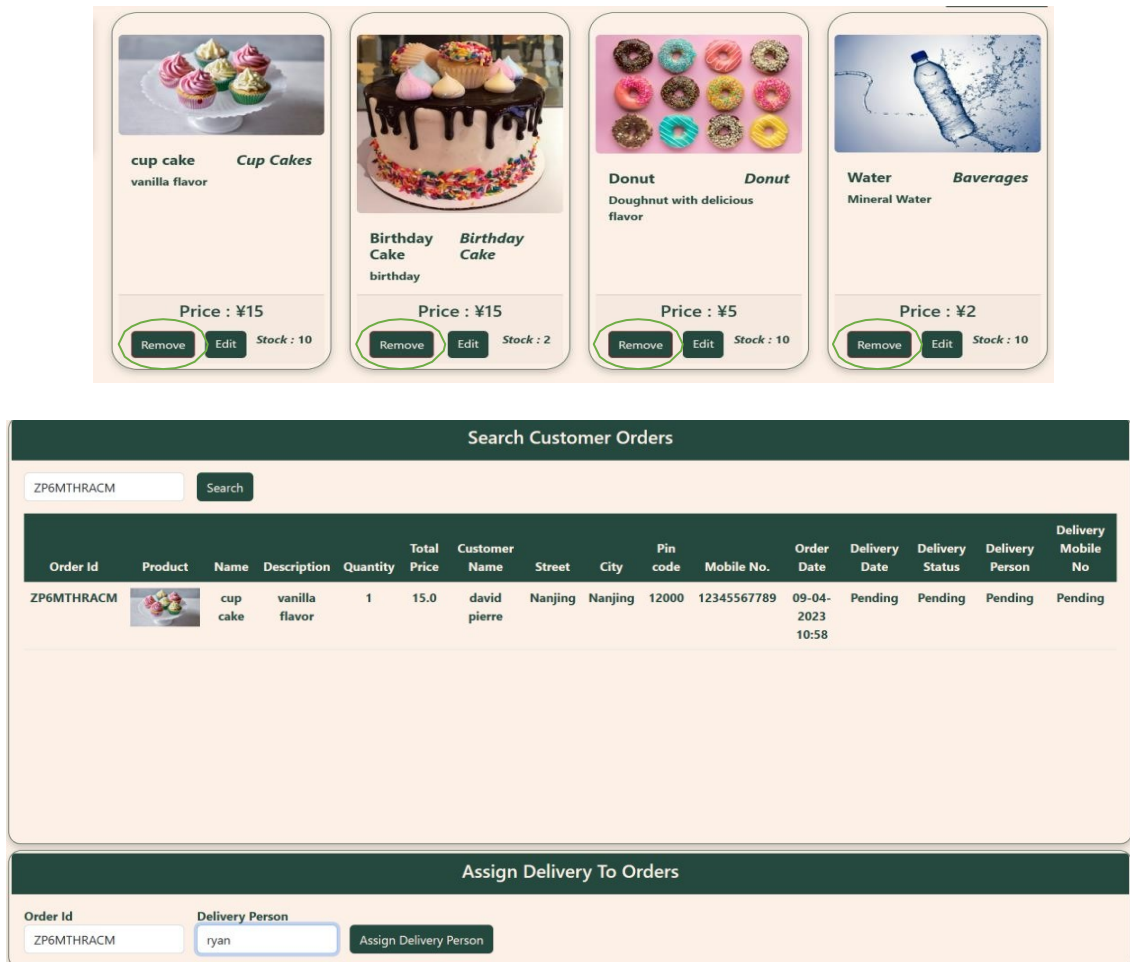


Figure 12. Product and Order Management Interfaces

Displays how the administrator adds, edits, or deletes products and assigns delivery personnel to orders.