Yizhe He, Yuning Gao

CSC 621·

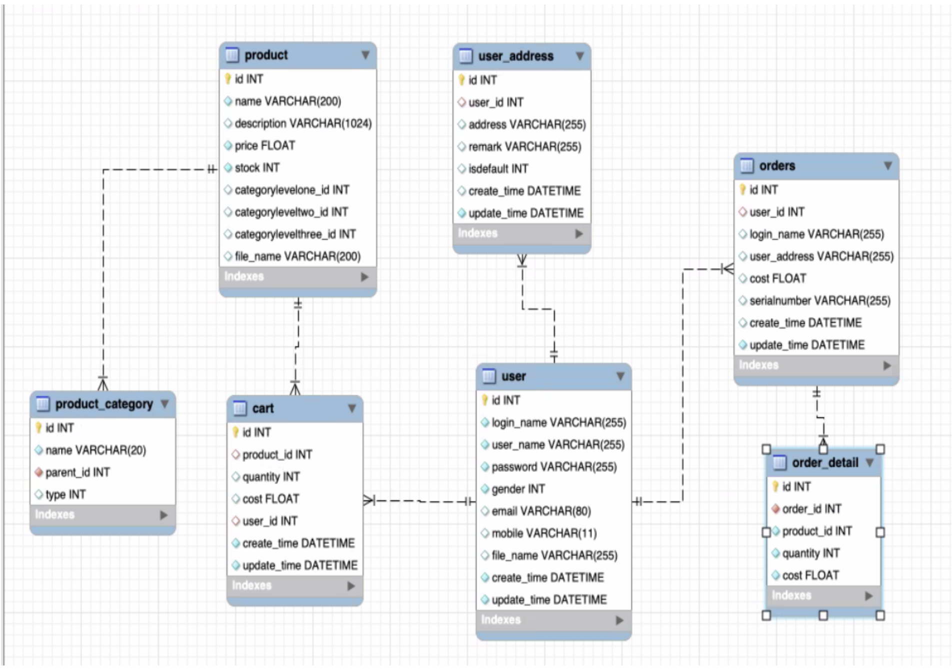
12/11/2023

Dr. Babak Forouraghi

Report of E-commerce web application

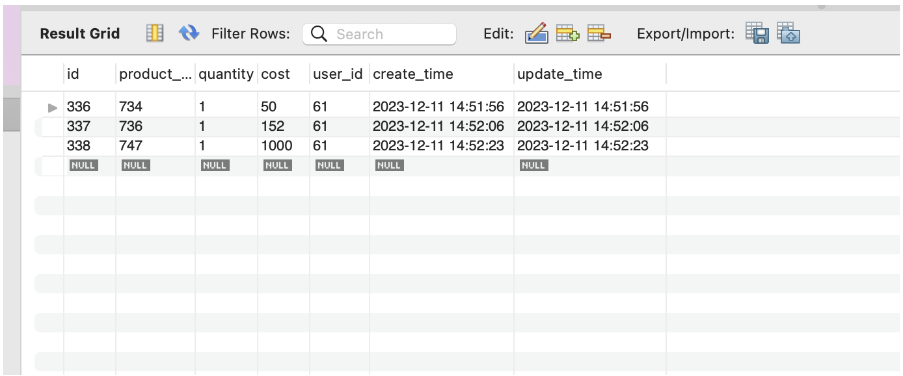
https://github.com/Yuninggao/final_project/tree/master

ER Model:



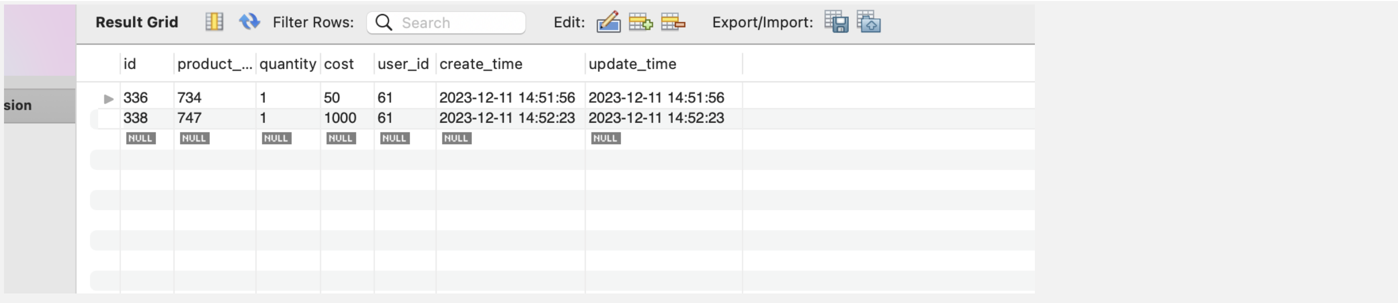
Select, Insert Into, Update, Delete:

add 3 products in cart



id	product_...	quantity	cost	user_id	create_time	update_time
336	734	1	50	61	2023-12-11 14:51:56	2023-12-11 14:51:56
337	736	1	152	61	2023-12-11 14:52:06	2023-12-11 14:52:06
338	747	1	1000	61	2023-12-11 14:52:23	2023-12-11 14:52:23
NULL	NULL	NULL	NULL	NULL	NULL	NULL

after delete 1 product:



id	product_...	quantity	cost	user_id	create_time	update_time
336	734	1	50	61	2023-12-11 14:51:56	2023-12-11 14:51:56
338	747	1	1000	61	2023-12-11 14:52:23	2023-12-11 14:52:23
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Member 1: Yizhe He

My teammate and I agree that I handled 45% of the overall project. My specific tasks included:

Task 1: I took charge of designing and implementing a crucial program module within the E-commerce application. This module focuses on managing user authentication and authorization using SQL for database operations and JavaScript for client-side interactivity. To achieve this, I created a SQL database schema to store user information securely. The HTML and CSS components were developed to provide a responsive user interface for the authentication and registration processes. JavaScript was employed to enhance the user experience by implementing dynamic client-side validations and asynchronous communication with the server for real-time feedback.

Task 2: My responsibility was to write dynamic front-end components for the E-commerce application, enhancing the user interface and overall user experience. Leveraging HTML, CSS, and JavaScript, I crafted interactive and visually appealing components.I implemented responsive design principles using CSS to ensure optimal display across various devices and screen sizes. JavaScript was extensively used to enable dynamic content updates without the need for page reloads. This significantly improved the overall performance and responsiveness of the application.

Task 3: In JavaScript, I have created two functions named "addQuantity" and "subQuantity." When users interact with the "Add" button, this function engages in several processes. First, it validates the current quantity based on inventory levels and retrieves basic information such as product price and ID. Subsequently, it efficiently handles quantity updates by increasing or decreasing the quantity of the selected item, calculating the new cost, and dynamically refreshing the information displayed on the page. Additionally, the function utilizes Ajax to interact with the server, seamlessly updating the shopping cart with the new quantity information. It is worth noting that it dynamically recalculates and updates the total cost of all items in the shopping cart.

Task 4: In JavaScript, I have created a function called "removeCart." This function is invoked when users interact with the delete button associated with products in the shopping cart. During this process, the function first determines the index of the clicked "delete" button relative to other elements with the "delete" class. Subsequently, it extracts the unique identifier (ID) of the corresponding item from the associated hidden input field. To enhance the user experience, a prompt notifies users when attempting to delete all items from the cart, ensuring that users are aware the cart cannot be empty. Upon user confirmation of the deletion, the removeCart function redirects users to the server endpoint responsible for removing items from the shopping cart.

Member 2: Yuning Gao

My teammates and I agree that I handled 55% of the entire project. My specific tasks include:

Task 1: designed the product category for better performance of CRUD operation. For instance, I design and implement an e-commerce product list divided into three layers. I build a hierarchical relationship using a field that points to the ID of its parent category. To achieve three-layer product classification, it is necessary to reasonably design the database structure and establish the association between classifications.

Task 2: Encryption for sensitive data. For example, in order to ensure user security, I created a password encryption class

Task 3: Designed filter and use session for data integrity and API security. When the user logs in, I use session to store user information to identify the user and follow the singleton design pattern that ensure each user has only one shopping cart.

Task 4: Designed Exception handling. For example, when a user adds an item to the shopping cart, the get function extracts user information to ensure that the user is logged in. If the user is not logged in, an alert will be sent to inform them that they are not logged in. In addition, we also implemented a lot of error checking logging. For example, if the inventory does not meet the quantity required by the user, the backend will receive log reminders such as inventory errors. , if the cart quantity is reduced to less than one, the user will also receive a hint of at least one.

Task 5: Implementation for CRUD operation. I use the post method to add user information to the database. When a user registers, we send the information to the database and retrieve the information. If the username already exists. We will create log message username exists.

Task 6: Ensure the transaction won't cause data corruption. For instance, I implement the functions of adding and deleting shopping carts in the cart class. When implementing this function, I will also check the database and adjust the inventory quantity in the database. If the user adds items to the shopping cart, I also need to subtract the inventory in the database from the number of items in the shopping cart. If the user deletes items from the shopping cart, I also need to Add the inventory in the database to the number of items in the shopping cart. In order to implement this function, I also created an xml class and wrote a query for inventory upgrade. In the modify shopping cart function, I used the post api to create an update method to update the shopping cart through id, variable, etc.