

CS316 Final Project Report

Team Name: Prime Team

Members: Isha Shah, Naveen Siva, Tanvi Poondota, Yasha Doddabele

Final Database Design

Users(id, email, password, firstname, lastname, account_balance, is_seller, balance)

- The Users table keeps a record of all users who have an account. It stores their personal information and balance information, as each User corresponds to a unique record of this.

ProductCategories(id, categoryname)

- The ProductCategories table corresponds to specific categories products can fall under. Each category has a name and a unique id.

Products(id, name, price, available, short_description, long_description, category_id)

- The Products table keeps a record of each individual product and their attributes, which can be accessed to display or aid in order fulfillment and final account balance. The Products table is mainly accessed to display search results depending on keywords or phrases searched for, and to display products that belong to specific categories. Each category_id references a unique id for a category in ProductCategories. The Products table updates each time a new product is created by a seller.

Orders(id, uid, date_fulfilled, fulfilled, processed)

- The Orders table keeps track of each order placed/attempted to be placed. New attributes include the processed attribute, which indicates if the order has successfully been placed (inventory/balance was sufficient). The fulfilled attribute indicates if the order has been successfully fulfilled, meaning all items in the order have been fulfilled by individual sellers. This helps differentiate between fulfilled and processed orders. uid is a foreign key reference to the Users table.

OrderItems(id, oid, pid, quantity, fulfilled)

- The OrderItems table keeps track of all order items within all orders, as well as the associated quantity for each. New attributes include the fulfilled attribute, which indicates if the seller individually fulfilled this item. It has a foreign key reference to Orders in oid and Products in pid.

ItemInCart(id, uid, pid, quantity)

- The ItemInCart table keeps track of the specific items a specific user has in the cart. This information is stored, so a User can access it any time they log in and want to check out. Technically its superkey is id, but the uid also functions as a superkey because each user has only one cart.

SellerInventory(id, seller_id, product_id, quantity)

- The SellerInventory table keeps track of the Seller's inventory and keeps a record of the updates, in case a seller adds or deletes to their inventory. The table stores which specific products the Seller has and their quantities.

SellerOrders(id, item_id, seller_id, order_date, total_amount, total_items, fulfilled)

- The SellerOrders table keeps track of the orders of specific sellers and additional details about the purchase, as well as fulfillment status.

ProductRatings(id, product_id, buyer_id, rating, review, created_at)

- The ProductRatings table corresponds to the specific ratings products have received. This is a new table that is needed to store which buyer gave a review and for which product.

Final List of Features

Account/Purchases features:

- Any person who goes to the website can register for a new account, if there is not one existing under that email, or log in with their existing email and password. There will be error checking on if the email and password is stored in the database. If there is a new user registering for an account, they will input their credentials, and the form will validate the email input, in case it is already stored or if it is a valid email. Once logged in, the user can navigate to their accounts page. This page displays their name, address, and id. They are also able to update their password here. Along with this information, they are able to add to their account balance, so they can use that money to buy products. It will then decrement the amount when a product is bought.
- A User's purchases are stored under the purchase history tab. The purchases will be sorted reverse chronologically by default. Each purchase order has a brief description and it also links to the order page.
- The public view of any user can be seen when you go to a specific product and click on the person's name who gave a rating. It will show their account number, name, and reviews they have given.
- The public view for a seller can be accessed by clicking on the name of the seller who is selling the product on the specific product page. This public view will display their email, address, account number, and name.
- In terms of advanced features, when a User goes to their purchase history, they are able to filter by total amounts, and also sort by price (high to low and low to high). This unique feature I believe should have a value of 1 point. Additionally, the public view of a User also displays the reviews they have given. I believe this should be worth **0.5 points**. This is because the reviews functionality was not a requirement for our project. Additionally, a user can register as a seller later on, and a seller can change their status to not being a seller anymore, on the update account page. According to this, the UI will change and some mechanisms for new sellers will be available. I believe this should be valued at 0.5 points.
- All of these features are fully implemented/functiona.
- The delivery location in the navigation bar is set to Durham as default but updates based on a users' address when they log in. This should be worth **0.5 points**.

Products features:

- I implemented the home page which displays all of the product categories that a product can belong to. This is an additional feature which I believe should be worth **1 point** because the page is aesthetic compared to other pages we implemented throughout the site and adds a theme to the user view.
- The user can click on each category box from the home page to access a Category Results page that displays all of the products that belong to the category selected. The user can also access this page using the drop down menu of categories that is placed to the left of the search bar on the navigation bar.
- Once a user is directed to the Category Results page, they can view all the products that belong to the category selected. These results are paginated so if the results for each product include more than 24 products, the user has the option to view the products on a separate page. I also implemented the functionality where users can select whether they would like to view 24, 48, or 64 pages at a time. This is an additional feature which I believe should be worth **2 points** because the results number option that the user selects actively updates the results list that is displayed, and the option selected remains consistent as the user navigates through the pagination links. There is a message that displays how many products are displayed across all pages, and this message updates the user on which section of the results the user is viewing on a given page. The message updates to broaden the range, depending on whether the user wants to view 24, 64, or 48 pages at a time.
- I implemented the search results page. A user is taken to this page when they search for a keyword or phrase using the search bar on the navigation bar. The search results display similarly to the category results page. These results are paginated in the same way and users have the option to select to view 24, 48 or 64 pages at a time.
- On both the category results page and the search results page, viewers can view the product name, image, short description, price and average rating of each product in the results. They can also choose to filter through the category results and search results by price, either high to low or low to high. This is an additional feature that I believe should be worth **2 points** because the filtration option that the user clicks is remembered and remains consistent even when the user clicks through different pages of the pagination links. Also once a user clicks on one of the results in the list, and decides to go back to the results page using the “Back To Results” button, the results are still filtered by price (and page number view).
- Once a user clicks on a product, from either the category results page or the search results page, they are directed to a product details page which lists more in depth information about the specific product. Here the user can view the product name, price, long description, image and average rating for the product. They can also see whether there are sellers for the product, if not the product is labeled unavailable. They can fill out a form to add the product to their cart which fails if they are not logged in. These features are all viewable if the user is not logged in.

- I also implemented a function that calculated the average rating for a product, which I believe should be worth **1 point**. I displayed the average rating for each product in the results views and when the user views a product details page. If a product does not have any reviews/ratings, I made sure to list that no reviews were available for the product so that the user was aware of this.
- On the Product Details page, the user can also view any reviews that have been left for the product. This is an additional feature that I believe is worth **2 points** because we had to redesign the schema to accommodate for the functionality. We added a ProductRatings table. We also had to add new methods to fetch all of the reviews given a specific product_id. The user can view reviews both if they are logged in and if they are not logged in.
- On the Product Details page, I also implemented a scrolling bar view of products towards the bottom of the page which displays 10 products that are similar to the product. I believe that this additional feature is worth **1 point** because I had to implement a new method to find similar products to the product displayed on the details page based on keywords that the specific product had with other products in the database. Each of the products is clickable, and leads you to the selected product's product details page. This is viewable even if the user is not logged in.
- I also implemented a scrolling bar view of products beneath this which displays 10 other products that belong to the same category as the product being displayed. I believe this additional feature is also worth **1 point** because I had to implement a new method to find 10 products of the same category. Each product displayed is also clickable, and leads the user to the selected product's product details page. This is viewable even if the user is not logged in.
- Now if the user logs in, the user can access all of the same pages I mentioned earlier. From the product details page, a user can add a product to their cart only if there are sellers available for the product and if the quantity being added is less than or equal to that in stock.
- A logged in user that has purchased a product can access their purchase history and view products similar to the most recent product they purchased. This appears as a scrolling bar view of products on the purchase history page. The user can click on each product and is directed to the product's product details page. This additional feature should be worth **1 point** for the additional methods I implemented to find the most recent product purchased and similar products to this.
- Lastly, I implemented a logic functionality so that a user can only review a product, using a review form that is displayed on a product details page, that they have purchased. I believe this additional feature should be worth **2 points** because I implemented new methods to check whether the logged in user has purchased the product that they are viewing, and only display the review form and allow a user to review a product if they have purchased it, otherwise the user does not have any access to the review form.

- If you click on the amazon logo, you are redirected to the home page.
- All of these features are fully implemented/functional.

Cart/Order features:

- Each user has one shopping cart, and it is fully functional. You can add available items to your cart through the detailed Products pages, and within the cart page you can update each cart item's quantity and delete the item from the cart. The total price of the cart is listed and changes dynamically depending on a user's updates and deletions within the cart. If there are no items in the cart, the page does not display a table and tells you to go shop for items. Finally, a user cannot see their cart if they are not logged in. This includes simply going to the /cart route as an unauthorized user, but also if you are on the carts page, you log out, and you press the back button. It redirects you to the login page instead.
- User orders are stored and tracked, and they are fully functional as well. First, a user cannot place an order with no items in their cart. Once there are items, upon placing an order seller inventories and the user's balance are checked. If the item quantities for any items in the cart are more than the available seller inventory, the order is not processed, and the page tells you that the order could not be placed. The same happens if your account balance is not enough to pay for the total price of the cart. For successfully processed orders, the page tells you the order went through, lists the date of the processing, and sends the order to the respective sellers of each product in the order, waiting for fulfillment. The buyer's balance is decreased by the total price, the seller's balance is increased by the item price*quantity of what they are selling, and seller inventories are decremented. Once all order items are fulfilled, the entire order is listed as fulfilled, and the date of fulfillment is changed on the order page. All orders (processed/fulfilled or not) are linked to see in the purchase history page of a user.
- Advanced features: cart and order pages have an advanced user-interface and design.
- All of these features are fully implemented/functional.

Seller Order/Inventory features:

- A user who wishes to act as a seller will have an inventory page that lists all products for sale by this user. Each seller has the option to add inventory that already exists in the database, or can add an entirely new product. For each product in the seller's inventory, they can view and change the available quantity for sale to users. Additionally, they can also delete it if they do not want to sell the product anymore. If a user wants to add a completely new product, they have an option that collects all the necessary information about the product such as name, price, availability, and category of the product. After adding the product to inventory, sellers can then update the quantity of the newly added product. The new product is added to the SellersInventory table and the Products table. And then becomes this product becomes available for users to buy and sell.

- Additionally, sellers have the option to view all the orders placed by users in the platform for their products. In this View Orders page, sellers see two tables displaying information - Pending Orders and Fulfilled Orders table. The pending orders table displays all the current orders that users have placed, and then sellers have the option to either fulfill the order or not. If the sellers chooses to fulfill the order, the order will be removed from the pending orders part of the table and moved to the Fulfilled orders. Sellers can see the product details of the product such as name, price and ID of the product, and can also see buyer information such as name, address, and if you are a seller, you can see all the current orders placed by buyers for your products by clicking on the View Seller's Inventory Button will appear in the dashboard. Once a seller clicks on the button, it will take them to the Sellers's Orders page. Here, a seller can browse/search the history of orders fulfilled or to be fulfilled, sorted by in reverse chronological order. For each order in this list, show a summary (buyer information including address, date order placed, total amount/number of items, and overall fulfillment status), but do not show information concerning other sellers (recall that an order may involve multiple sellers), and provide a mechanism for marking a line item as fulfilled.
- Upon order fulfillment, the order submission automatically decrements the available quantity in the seller's inventory; so fulfillment should not further update the inventory. It also lets the user note that the order was fulfilled. This ensures sellers have an accurate and up-to-date inventory reflection post-transaction.
- The 'View Seller's Inventory' button, easily accessible from the dashboard, directs sellers to the Sellers's Orders page, where they can peruse the history of orders, organized in reverse chronological order. Each entry on this list offers a summary that includes buyer information and the order's fulfillment status while preserving the confidentiality of other sellers' details.
- Advanced Features:
 - We also have visualization/analytics for both the inventory and the order fulfillment pages to provide the seller with data to help them make better decisions. Sellers have access to bar and pie charts showing their current inventory breakdown by quantity and product, which provide a quick and clear visual representation of stock levels. Additionally, sellers can view all the current orders in the pending orders that need their attention as well as a bar graph. Additionally, there is another bar chart that shows all the fulfilled orders that sellers have fulfilled in the past, broken down by product name and quantity.
 - We also implemented the sellers review form where sellers can leave feedback for products and for the sellers themselves. We implemented a form in the product details page where a buyer, after purchasing a product can leave feedback for the sellers
- POINTS:

- I think the additional features(graphs and analytics) are worth **2 points** they provide additional information backed by data to sellers to help them keep track of inventory and manage orders
- I think the seller feedback feature is worth **2 points** because it provides valuable data to sellers to help better the experience. Additionally, because we are a 4 person team, we were told not to implement this feature, but we still went ahead and implemented this feature.
- All features are fully implemented/functional.

We each added comments to our code as well.