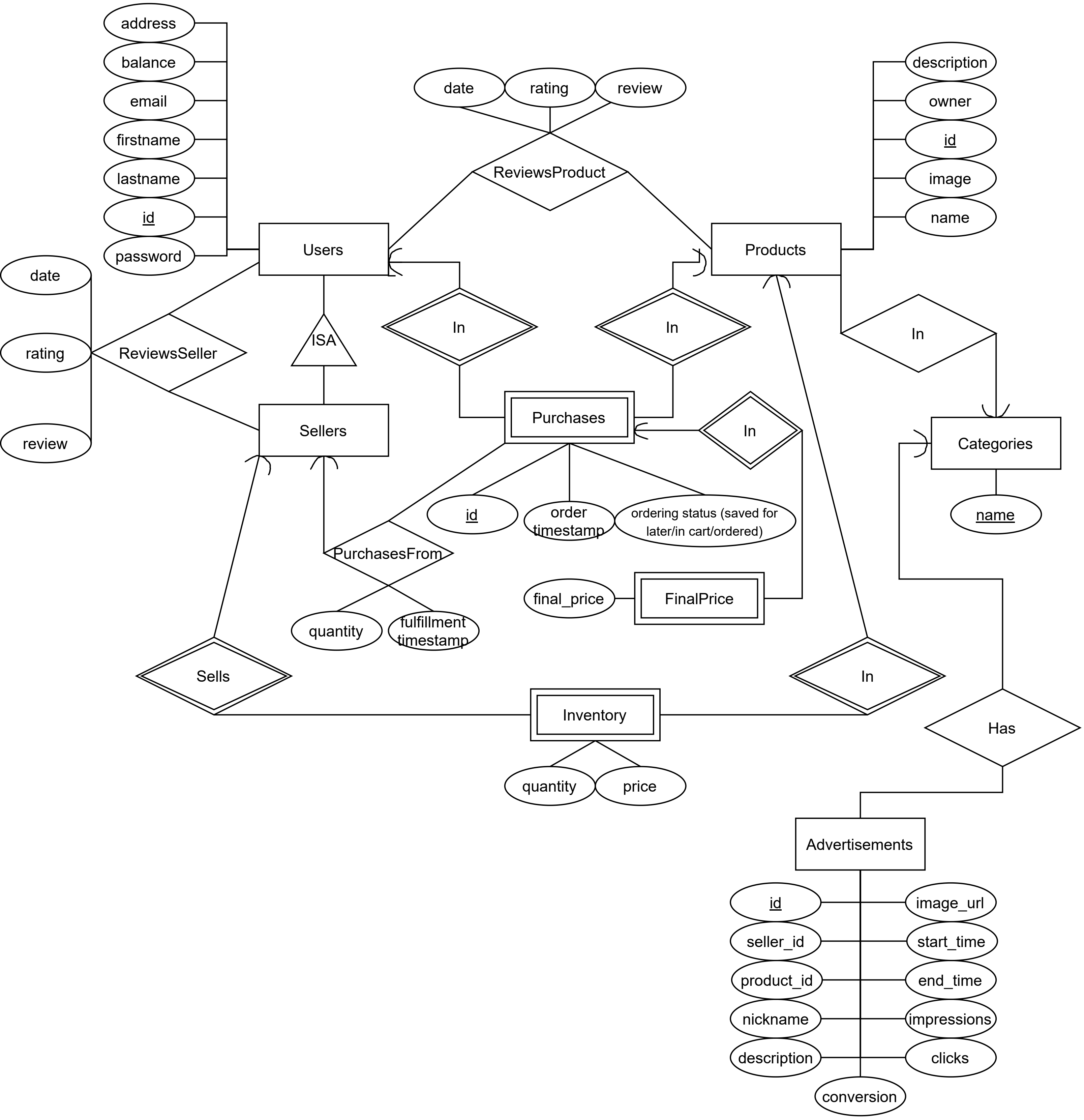
Refined Database Design [Highlighted Updates]

- Our database looks very similar to the database in Milestone 3, though we have added an additional table that corresponds to one of our additional features. Because of its supplemental nature, the table has not yet been integrated with the other database tables; integration would be done with additional time.
- *Advertisements*(*id, seller_id, product_id, nickname, description, image_url, category, start_time, end_time, impressions, clicks, conversions)*
    - Note: In our E/R diagram, there is a many-to-one relationship set that links *Categories* with *Advertisements*, as every Advertisement must be associated with exactly one category (though categories themselves can be associated with multiple advertisement records).
        - For simplicity, we combine the relationship set with the *Advertisements* entity set in our database (i.e. our create.sql file) and use a NOT NULL constraint on category (which references the *Categories* table). This ensures that all ads are associated with a predefined category.
    - Each Advertisement has an id that identifies its characteristics.
    - The *seller_id* attribute refers to the seller that created the specific advertisement.
    - *Start_time* and *End_time* refer to the range of dates within which the seller wants the ad to be displayed for.
    - *Impressions, clicks,* and *conversions* keep track of how many people have seen the ad, clicked on the ad, and purchased the product after clicking on the ad respectively.
    - *Description* refers to the text that will be displayed with the ad, and *image_url* will be the image of the product displayed in the ad. By default, when a Seller creates an ad for a product, these attributes will be set to the product's description and image. Sellers are allowed to change the image or the description that is displayed in case they want to design a more enticing advertisement.
    - *Product_id* refers to the id of the product being displayed in the advertisement.
    - Sellers can create a nickname for the ad as well, which will be the name of the seller's "ad campaign".


address
balance
email
firstname
lastname
id
password
date
rating
review
ReviewsProduct
description
owner
id
image
name
Users
Products
date
rating
review
ReviewsSeller
ISA
In
In
In
Sellers
Purchases
In
Categories
id
order timestamp
ordering status (saved for later/in cart/ordered)
name
PurchasesFrom
final_price
FinalPrice
quantity
fulfillment timestamp
Sells
In
Inventory
Has
quantity
price
Advertisements
id
image_url
seller_id
start_time
product_id
end_time
nickname
impressions
description
clicks
conversion

List of Features

User Basic Features [All fully functional]

- Requirement 1: A new user can register for a new account; an existing user can log in using email and password.
    - Login button and link is displayed when a user is not currently logged in, and it directs the user to a login page. Users enter their unique email and password to log in, and they are redirected to the home page if login is successful.
- Requirement 2: Each user account has a system-assigned id. Other account information includes email, full name of user, address, and password. Users can update all information except the id. Ensure that email is unique among all users.
    - Register button displayed in login form and redirects to registration form. Form requires data for the attributes indicated above it (first name, last name, email, password, repeat password, and address). Submission of registration form redirects to login with a flashed message of successful registration.
- Requirement 3: Each account is associated with a balance. It starts out as $0 but can be topped up by the user. The user can also withdraw up to the full balance. In practice these actions would require some real payment mechanism, but it is not required for this project.
    - Balance is displayed in a user's Profile, and it is set to $0 by default. Withdraw and deposit functions are performed through the withdraw and deposit form. Users enter an amount that is rounded to two decimal places, and that amount is deposited or withdrawn. Any withdrawal amount greater than the balance is blocked by a database trigger, and exception handling flashes a message declaring insufficient funds. Successful processing will flash a success message.
- Requirement 4: Users can browse their history of purchases, sorted in reverse chronological order by default. For each purchase in this list, show a summary (e.g., total amount, number of items, and fulfillment status) and link to the detailed order page (see Cart / Order).
    - Purchase History page is accessible through the Profile page via a link. The purchases are sorted reverse chronologically and displayed with the aforementioned information. Because of our database design, each purchase is one line item; viewing the whole order is available through the View Order link.
- Requirement 5: Provide a public view for a user. It will show the account number and name as well as any other summary information you deem necessary. If the user also acts as a seller, show also email, address, and include a section with all reviews for this seller (see Feedback / Messaging).
    - Public view is accessed through the Public Profile link in the Profile page or through seller id links. Regular user public profiles display user id and name, while sellers show email, address, and reviews (list and summary). Seller reviews can be written through the review form on a seller's public profile.

User Additional Features [All fully functional]

- Venmo feature allows for balance transfer from one logged in user to another user with the specified unique email. Venmo is found in the Profile page. Given the transfer amount, the Venmo feature checks user balance to prevent overdrawing before processing.
- Password verification in the edit profile form ensures the form is being submitted by the true user.

Product Basic Features [All fully functional]

- Requirement 1:There is a list of predefined product categories, and each product belongs to one category. At the minimum, a product should have a short name, a longer description, an image, and a price.
    - A predefined list of categories has been created by the database in the table. The *Products* table has attributes that store a product's name, longer description, image and image. Prices are created by sellers and stored in the inventory. Thus, a product can have multiple different prices if it is sold by many sellers.
- Requirement 2: Users can browse and search/filter all products. For each product in the result list, show a summary (e.g., image, name, average review rating, etc.) and link to the detailed product page. At the minimum, support browsing by category, searching by keywords in name/description, and sorting by price.
    - Through the use of a search bar that is displayed at the top of every page, users search for products. When a user inputs a search term, all products whose name, description, or category name contain that search term will be displayed to the user. I note that the results are paginated, as discussed below. I also note that the results are case sensitive in order to create more targeted search results. The page will display a table, such that users can see each product's id, name, image, price, category and average rating.
        - A series of buttons above the table of products allow for sorting. As it pertains to requirement 1, users can sort products by price in both ascending and descending order. By default, products are sorted by their ids until a user clicks on one of the buttons.
        - For each product in the table, it will be linked to its own detailed product page. Users can access the page by clicking on the product's name.
    - Through the browse by category button, the user is brought to a series of pages that display all product categories, which are paginated by letter. By default, the user will be taken to the A page (so the user will see all categories that begin with an A). Each category name is linked to a page that displays all of the products in that category. This page will look exactly like the page that results from the user inputting a term into the search bar.
        - As a note, I only show categories that have an existing product associated with them. There are quite a bit more categories without any products than categories with products.
- Requirement 3: A detailed product page will show all details for the product, together with a list of sellers and their current quantities in stock. For each seller, provide an interface for adding a specific quantity of the product from this seller to the user's cart. The page should also include a section showing all reviews for this product.
    - The detailed product page displays the name and image of a specific product at the top of the page, as well as a form for rating it.
        - The form has two fields: a rating field and a review field. The form will only update the database if a User has purchased the product before. If a user has already reviewed the product, then the review will be updated through the submission of this form. Otherwise, a new review will be created.
        - If a user is not logged in, the user will be sent to the login page and told "Log-in to review a product".
    - The detailed product page also has a table that shows information for all sellers of a product. Specifically, it displays the seller's id, the price the seller is offering the product for, the amount of quantity in stock that the seller has, and a series of buttons for adding the product from the seller to the user's cart.
        - Users can add the product to their carts in increments of 1, 5, 10, or 25. Additionally, users have the option to add all of the seller's stock to their carts. I

        also note that buttons will only be displayed to the extent that a seller has the given amount of quantity in stock (i.e. if a seller only has 2 units in stock, two buttons will be displayed: + 1 and + 2).
    - The detailed page also has a table that contains information regarding all reviews of the product. For each review, the page displays the reviewer's id, the rating the reviewer gave the product, the reason for the rating, and the time of the review. Above the table is a star graphic and text that indicates the average review of the product.
- Requirement 4: Users can create new products for sale. The user who created a product will be able to edit the product information.
    - From the seller portal, a seller can create a new product using a form that takes in 6 inputs (name, description, image url, category, quantity, and price). If a product already exists, then the new product will not be made. We define a product as already existing if there exists another product with the same name, description, and category.
        - This form checks the validity of an image's url by making sure it begins with "https://".
    - See "Seller Basic Features, Requirement 1" for information regarding the editing of products.

Product Additional Features [All fully functional]

- In addition to the requirement that the website must support sorting products by price, it also supports sorting by Name (A to Z and Z to A) and Product ID.
- The pricing filter allows users to input a range of prices within which to filter the products shown while browsing. Users are allowed to input only low price filter, so that products with a price below this input are not displayed. Alternatively, Users can input only a high price filter, such that all products with a price above the filter are shown. Finally, of course, the user could input both a high and a low price filter, so that the website displays only products within the range of the inputs.
    - When a user inputs a pricing filter, the results will be sorted by price from low to high (as it seems likely that users would use a filter in an attempt to save money).
- On the detailed product page, I have added a basic 5-star graphic for indicating the average rating for the displayed product. The graphic will generate a specific number of gold and grey stars based on the product's average review. That is, if a product has an average rating of 2.0 / 5.0, it will display two gold stars followed by 3 grey stars. If the product has a rating that is not an integer, I use traditional rounding rules to determine the number of stars to fill in gold - for instance, if the average rating is 2.6, the graphic would generate 3 gold stars. If the product has no reviews, the graphic will not render. It will be replaced by text that says, "No Reviews".
- When a user chooses to browse products, the results will be paginated, with a maximum of 50 items appearing on a page. Results on subsequent pages will remain sorted.
    - When one scrolls to the list of pages of products, the button that corresponds to the current page will be highlighted yellow. This feature makes it easier to browse very large search results.
- **Smaller Features**
    - For each product that is displayed when a user is browsing products, there exists a link to a page that displays all products in its specific category.
    - For each seller and reviewer of a product on the detailed product page, there will be a link to that person's public profile.

Cart Basic Features [All fully Functional]

- Requirement 1: Each user has a cart. The detailed product page (see Products) will provide a way of adding to this cart. Each line item in the cart refers to one product from one seller with a specific quantity. A detailed cart page should list all line items (quantities and unit prices) and the total price. It should also provide ways to change quantities, remove line items, and submit the entire cart as an order.

  - User's cart accessible by adding a product to it (through the detailed product page) or using the button in the top right of the website's header. For each cart item, its product id, image, name, unit price, and quantity are shown. For each item there are buttons to increment or decrement that item's cart quantity or delete it altogether. At the bottom of the page, the cart's total price is shown with a button to place the order.

- Requirement 2: When submitting the order, make sure to check available inventories and balances. Beware that inventories and prices can be constantly changing. For simplicity of this project, we will update inventories and balances at the time of order submission. The buyer's balance will be decremented, and the sellers' balances will be incremented, and inventories decremented. Once the order is placed, the cart becomes empty.
  - When "place order" is clicked, it calls a function that checks if the user's balance is greater than the cart's total price and if there is sufficient stock of each item. If either of the checks don't pass, the order does not go through, an error message is shown to the user, and no updates to the database are made. If the checks pass, the order goes through, the user's balance and inventory quantities are updated, the cart is emptied, and the user is redirected to an order summary page.
- Requirement 3: The cart contents are persistent: i.e., they should remain after users leave the site and log back in.
  - Cart items are loaded by querying the Purchases table for rows that match the user's ID and have order status "In cart". This status is only updated to "Ordered" when the order goes through, so its contents remain between sessions (unless the user places the order).
- Requirement 4: A detailed order page (from the buyer's perspective) should contain all the information that would have been found on the cart page, but with prices "final." In addition, for each line item, it should show if and when that line item has been fulfilled by the seller (see Inventory / Order Fulfillment).
  - An order summary page is accessible through a few places in the website. First, it is displayed when a user places an order. It can also be accessed through a user's purchase history page, or a seller's sales history page.
  - For buyers, the page will show the following information for each item in the order: product ID, image, name, unit price (immutable and final following the time of the order), quantity, time purchased, and order status. Order status will be displayed as either "Ordered" or "Fulfilled" here (see Inventory / Order Fulfillment).
  - When sellers view a buyer's order, they can only view the items they are supplying, for which all the info listed above will be shown. Furthermore, sellers will see a buyer's address and email, and a button to update order status from "Ordered" to "Fulfilled" to ensure that an item has reached the buyer.
- Requirement 5: The entire order should be marked as fulfilled if all line items are fulfilled. Once submitted, an order cannot be changed by the buyer.
  - The top of the order summary page shows an overall order status. This depends on how many line items in the order have been fulfilled. If none of the products have been fulfilled, overall status is "Placed". If some have been fulfilled, overall status is "Partially Fulfilled". If all have been fulfilled, overall status is "Fulfilled".
  - Buyers have no way of altering order line items after placing the order.

Cart Additional Features [All Features are Fully Functional]

- Save for Later
  - Besides editing the quantity of an item in one's cart (or deleting it altogether), a user can save it for later, which removes it from the cart, and adds it to a list of saved items. These

items can be viewed by clicking “View your Saved Items” in the cart. In the saved items page, a user can either re-add an item to the cart or delete it. Once there are no saved items, a user is redirected back to their cart.

Seller Basic Features [All fully functional]

- Requirement 1: A user who wishes to act as a seller will have an inventory page that lists all products for sale by this user.
  - From the Seller Portal (on which reviews don’t appear, because that information is “public”), a user can see their inventory, along with relevant details of each product that the person is selling. From the “Seller Portal” page, users can create a new product, add a product to the inventory, edit a product they created (designated by “owner” status), and remove it.
- Requirement 2: There should be a way to add a product to the inventory.
  - From the inventory page, sellers can choose a button to “catalogue add to inventory”. This button allows sellers to search and filter as if they were browsing normally, except there is a button next to every product to “add to inventory”. The “add to inventory” button ends the process by adding the product to the seller’s inventory. Required heavy integration with the products guru, who created the search results pages that this feature takes place in.
- Requirement 3: For each product in the user’s inventory, the user can view and change the available quantity for sale by this user, or simply remove it altogether from the inventory.
  - There is a “remove product” button for every product row on a seller’s inventory page. For each product on the page, quantity is also displayed in an input field. Additionally, the seller can quickly change their price OR quantity by just typing in a number in the row they want to update and pressing enter.
- Requirement 4: A seller can browse/search the history of orders fulfilled or to be fulfilled, sorted by in reverse chronological order by default.
  - Reverse chronological order is the default method of sorting order history, but (as mentioned below) sellers can sort the page any way they want.
- Requirement 5: 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. (Recall from Cart / Order that order submission automatically decrements the available quantity in the seller’s inventory; so, fulfillment should not further update the inventory.)
  - The necessary columns are in place and integration with the cart guru makes sure information from other sellers doesn’t appear on this page. Information from other sellers also will not appear when the seller clicks to view an order, for which there is a seller view and a buyer view. From viewing a single order or from the fulfillment status column, the seller can fulfill a line item. Sellers can see the exact time they fulfilled a line item by clicking fulfill again at any point afterwards, which will let them know they already fulfilled the order and when.

Seller Additional Features [All Features are Fully Functional]

- Advertisements API: Sellers can create an ad for any product. Ads consist of an image and description appearing side-by-side in a banner (sellers can change the image and description, as well as give a nickname to their ad for easy finding later on, but these default to the product image, description and name in case the seller is incredibly lazy). Ads appear on product pages for products in whatever category the seller chooses to advertise to. Each ad can only appear in one category at a time, in order to encourage sellers to create multiple ads for a single product if they wish to advertise to multiple categories. This feature is intended to improve the tracking of advertising performance. From the ads manager page, sellers can see their current ads, with their inactive (not currently running) ads below. Sellers can sort/jump to either table. They can also edit ads, see what the ad will look like to a user, and create a new ad from this page. Each ad in the "inactive" table shows its most recent schedule. Impressions, clicks, and conversions (defined as the quantity of this item that has been added to users' carts directly after they clicked the ad) are all tracked and viewable here. Conversions are encouraged through highlighting the seller whose ad a user clicked to get to the product page. Methods exist in their own class, and steps to integrate ads into any page are clearly explained.

- Sorting Module: (not explicitly defined, but there are multiple consistent backend and frontend implementations). This module displays cool-looking up/down arrows directly in the headers of every column and allows a user to intuitively choose to order by any column ascending or descending, and then the second sort is always id ascending. This is also always the default sort when the user first gets to the page. If we onboarded a new team member, we might have a .models.sort which contains easily duplicated functions along with an explanation (a quick one is already given in our comments, in ad.py) of how to reproduce a sorting module for the page you want to add sorting to.
- Update Module: Module, which keeps track of what row a user has just updated using the "update" variable. For every case in which a seller would interact with a product and end up in a page contained in the seller portal, a message will flash with a button that jumps to the updated row. This updated row is highlighted, and there will be a "jump back" button in this row to get back to the top - since it might be annoying to scroll all the way back up. Commented explanations are sufficient for replication.

Reviews Basic Features [All fully functional]

- Requirement 1: A user can submit a single rating/review for a product. The submission link will be incorporated in the detailed product page. The user cannot submit multiple ratings/reviews for the same product, but can edit/remove any existing ratings/reviews by this user.
  - On the detailed product page, users can access a form with fields for a numerical rating from 0 to 5, and a text review. From the same page, they can click the submit button, which updates the database and reloads the product page with the newly created review. Furthermore, if a user has already submitted a review for a product, this form can be used to update their review. Finally, if a user has not purchased this product before or attempts to submit an invalid review, the page will warn them of the problem with a message. In this case, the form will not interact with the database on the backend.
  - Reviews can be deleted in the proper table on a user's private profile page.
- Requirement 2: A user can submit a single rating/review for a seller, provided that the user has ordered something from the seller. Incorporate the submission link in appropriate places in the website, e.g., the detailed order page (see Cart / Order) and the public view of a seller (see Account / Purchases). Again, the user cannot submit multiple ratings/reviews for the same seller, but an existing rating/review can be edited or removed.
  - On a seller's public profile page, users can also access a form with entries for numerical ratings and text reviews. Similar to how product reviews work, the submission of a seller review checks if a user has purchased a product from this seller, and it prevents users from reviewing sellers they have not purchased from. Users can also access the seller review form by clicking the seller ids from their purchase histories or cart/order views. Finally, users can edit existing seller ratings/reviews by resubmitting the form. This will appropriately change the review in the database rather than creating a new review entry.
  - Reviews can be deleted in the proper table on a user's private profile page.

- Requirement 3: Each user should be able to list all ratings/reviews authored by this user, sorted in reverse chronological order by default. From this interface the user should be able then select ratings/reviews to update. Incorporate the link to this interface in user account view.
    - A user can access a list of all the ratings/reviews they authored in their detailed user account view. The reviews are listed in tables that show the products and sellers to which the review refer to, as well as the review's numerical ratings and text contents. All tables are sorted in reverse chronological order. The tables also contain links through which users can select specific reviews to update. The links take users to the appropriate product/seller pages where the form for updating reviews is located.
- Requirement 4: Produce summary ratings for products and sellers; pages or sections showing lists of reviews for products and sellers. At the very least, the summary needs include the average and number of ratings; the reviews lists should be sorted by rating or date.
    - Finally, the detailed pages for products and sellers contain a section with summary ratings and lists of reviews. The summary shows the average rating and number of ratings, while the review lists are sorted in reverse chronological order.

Other Additional Features [Fully functional]

- Last Chance Offers on the Home page displays information of 20 products with Inventory listings having low quantity in stock. Future work could improve the feature with pagination and more products.