Iteration 2 Report

Group 6: X00195342, X00204831, X00217101

A screenshot of a computer

Description automatically generated

Lamar Fadil (X00195342) – Search App, Wishlist App

Daryna Tertychna (X00204831) – Order App, Payments(Stripe)

Sofia Kabanets (X00217101) – Account App, Cart App

**Search App:**

I decided to start off this iteration with the search app. It was planned to create an advanced search using methods such as “Q Objects” to give us the ability of filtering search using price range, category, etc. This search bar is to be located on the navigation bar, where it would be easily accessible by the user, aiding in the creation of a more user friendly interface.

During the implementation stage, I encountered a flaw where after using the search bar on the main page, the user would get redirected to a different page with a different search bar dedicated for filtering items. I assumed that this was an issue with the urls, however, after playing around with the code and testing different methods of implementation, the problem still remained.

It was decided that a simpler search bar was to be implemented temporarily for this iteration. Ultimately, we would like to have the option of a filtered search by the end of this project, and I plan on testing more methods as well as bringing the issue up with our project advisor in attempt to put the desired search bar into effect.

**Wishlist App:**

In order to make the wishlist app, I thought it would be best to implement it in a way that is similar to a cart app. This method worked very well as the wishlist and cart app hold the same function for different purposes. Previous labs where a cart app was applied were very helpful as I used them as a guideline. Of course, there were some differences such as leaving out the subtotal and checkout sections that were unnecessary for a wish list. I also didn’t make the product availability effect whether the user can or can’t add a product to the list, as it’s not typically a characteristic for wishlists.

During implementation, I was reminded that this feature was to be accessed by registered users only. To make this work I imported “login\_required” to my app’s views.py.

**Order App and Stripe:**

The order app is created to show the customers their orders. First as list of orders, but if clicked on specific order the customer will see all information about this order. I created models and views for this. Before that, payment through Stripe was introduced. There were no problems with this, but after authorization (only a registered users can see their orders), the entire list of orders did not appear. It was redone, but the problem was not found. There is no problem with DB, so in the next step we have to ask the supervisor for advice and fix this bug. It will also be necessary to implement an additional form of order notification. For example, after placing an order, the client will see the details of the order on the main screen, and will also receive a letter to the e-mail.

**Accounts App:**

During the development of the "accounts" app for our Django project, I encountered several challenges that required careful consideration and problem-solving. One of the key aspects I focused on was user authentication and profile management. Setting up user registration, login, and logout functionalities was relatively straightforward, but ensuring seamless integration with user profile creation proved to be more complex. I faced issues such as foreign key constraint violations and integrity errors when attempting to create user profiles upon registration. Configuring login and logout functionalities, including specifying redirect URLs and implementing custom views, also posed challenges. Resolving conflicts between migrations, cache files, and Git branches was another hurdle I had to overcome.

**Cart App:**

Setting up the basic functionality of the shopping cart, including adding and removing items, was relatively straightforward. However, I faced challenges when implementing more advanced features, such as updating item quantities and handling out-of-stock items. I had to carefully design the data models and logic to ensure consistency and accuracy in the cart's behavior. I spent a lot of time trying to make session tokens work to create a cart for any user.