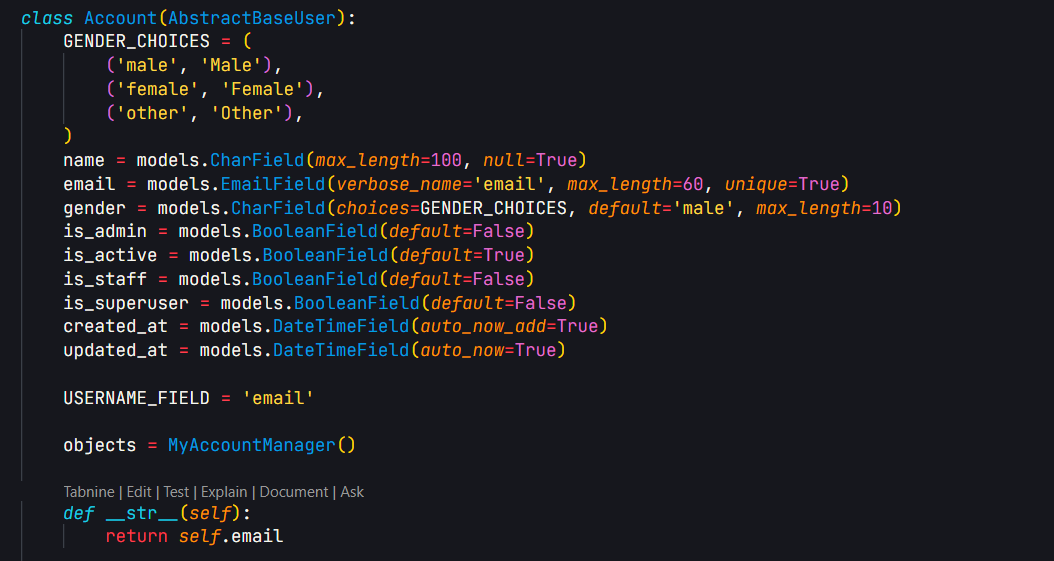
**Core Functionalities of Phase 1**

**Introduction**

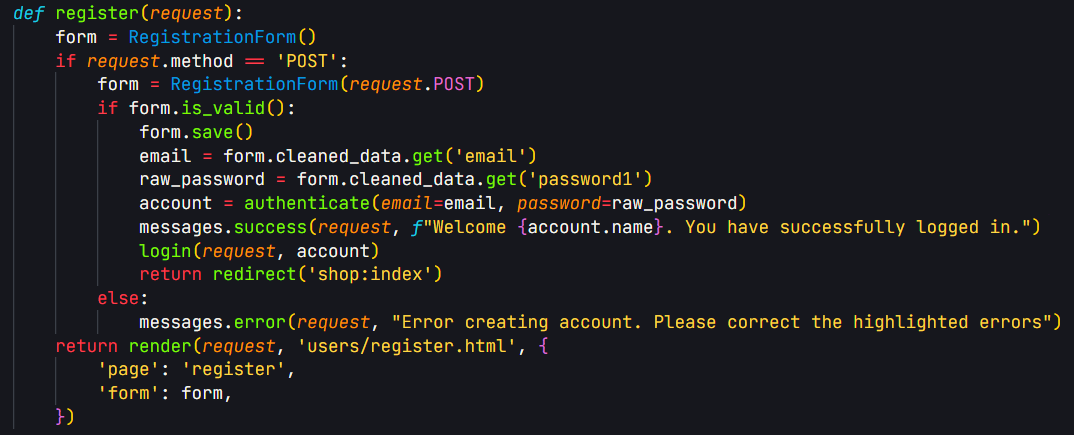
Phase 1 of the project is focused just on the critical features like—User Authentication, Product Listing, Search, Product Recommendation. The following steps include source code and explanations for each component but they’ll serve mainly as a reference for our development and stakeholders. Future enhancements and scalability of our e-commerce platform depend on this foundational work.

**1. User Authentication and Profile**

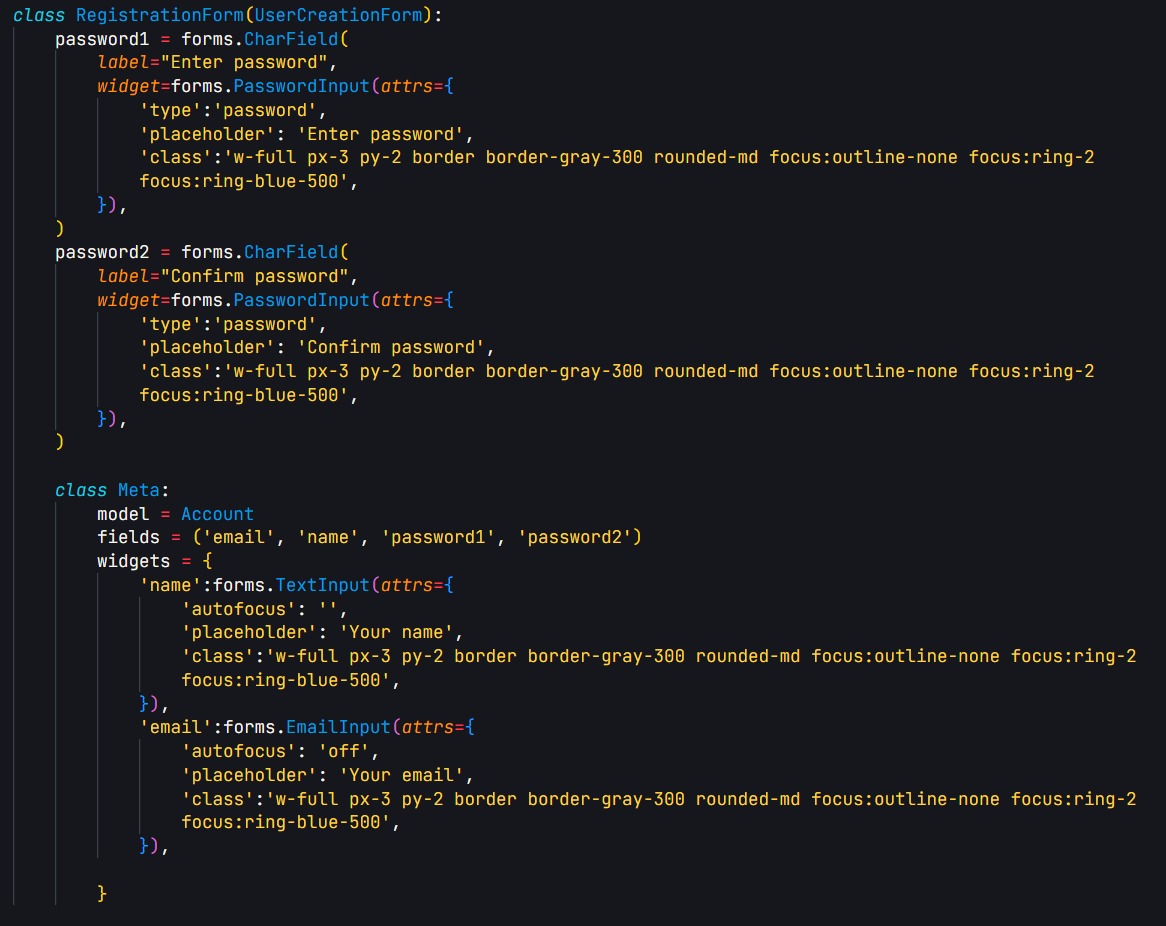
To begin with, we designed a custom user model that allows the implementations of most of the functionalities especially those designated for phase 1. Here is the Django model specifying the entity as Account (user’s table) and the various attributes that a user can have:



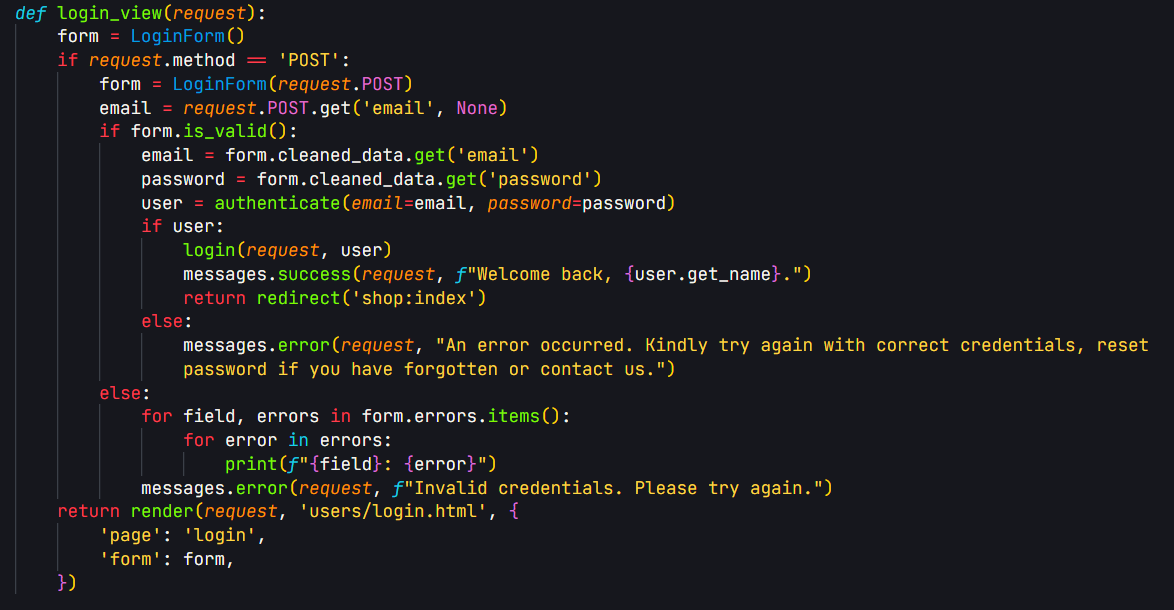
Registration function uses Django function-based view to register user using their credentials:



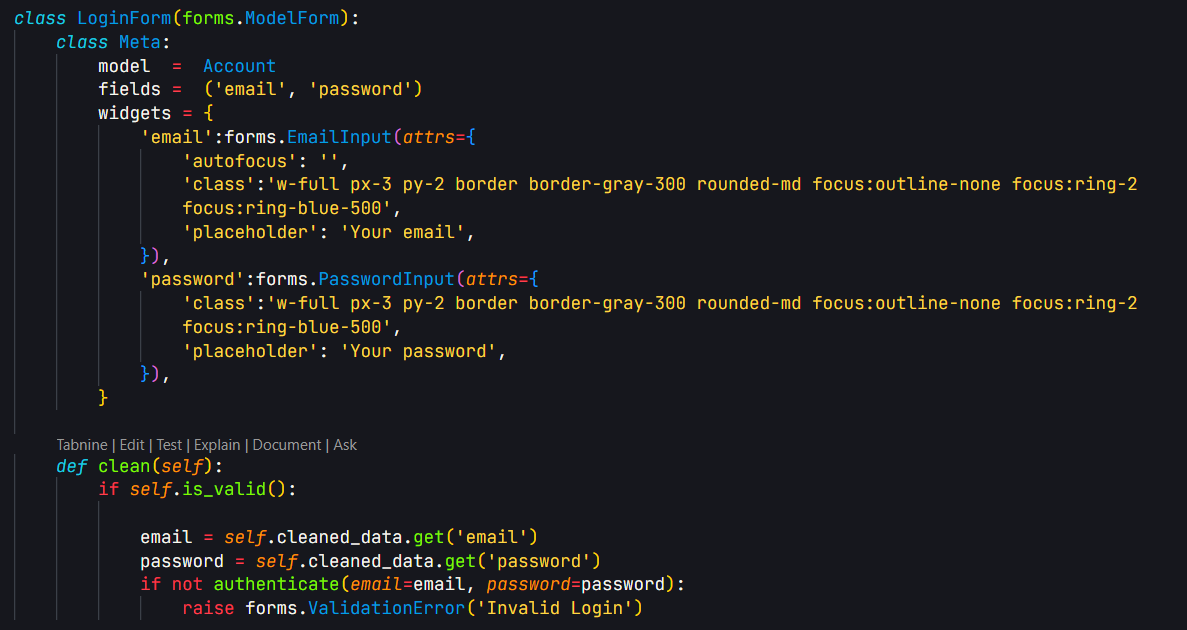
This makes use of the Django form to ensure form cleaning and input validation:



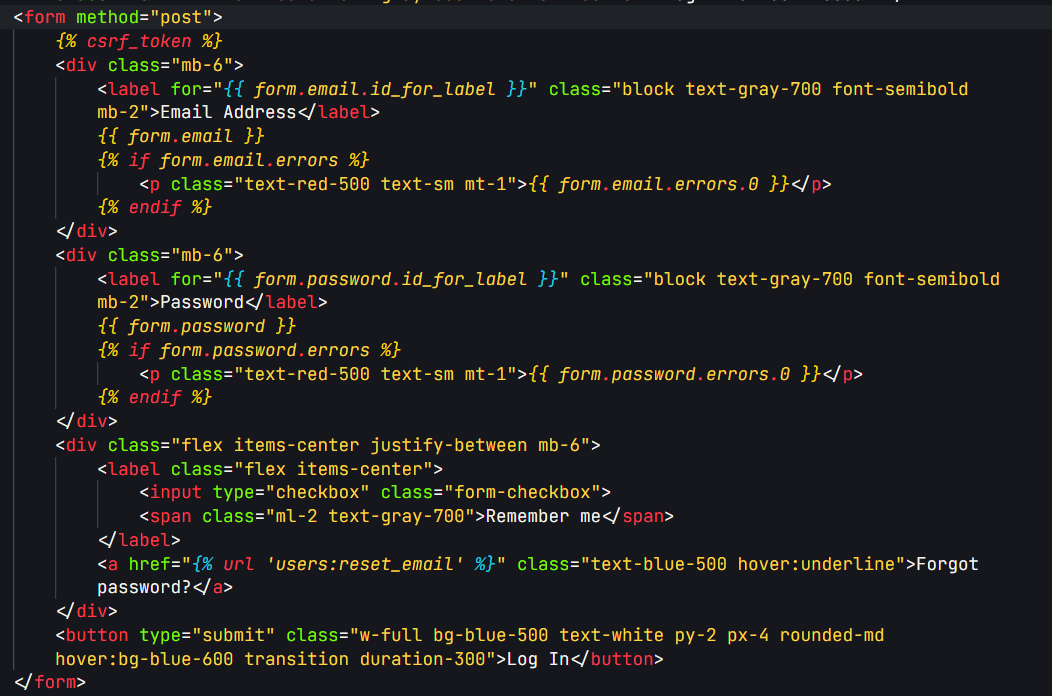
Similarly, the for user login, a Django function based view is constructed to handle user login as shown:



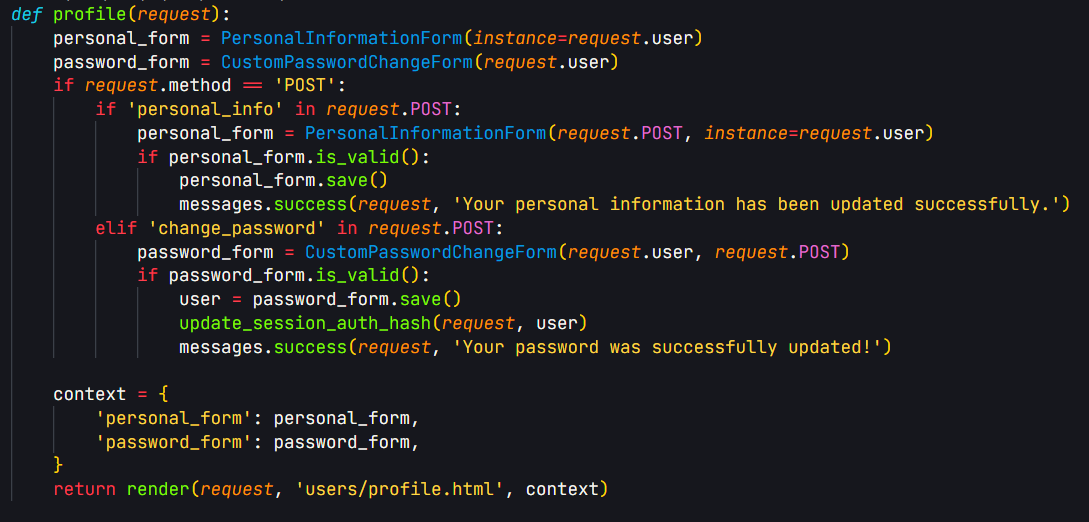
The following is its respective Login form:



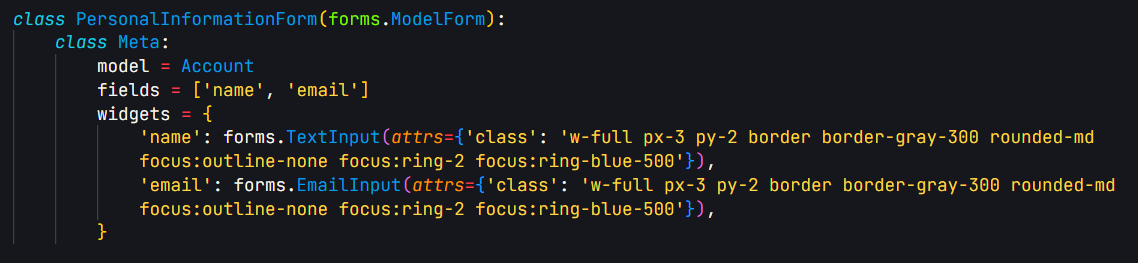
Finally, it is rendered in HTML page as shown below ensuring to also include CSRF tokens for enhanced form security:



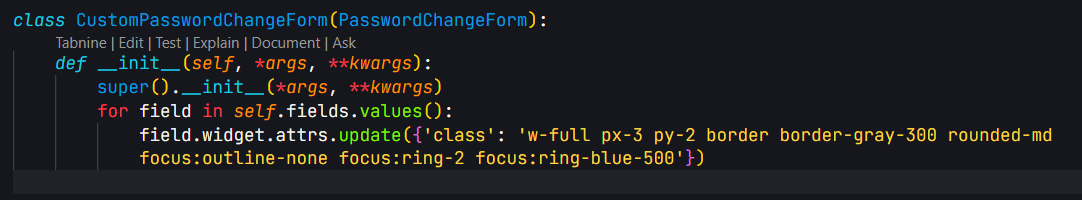
Once logged in, the user has access to their profile with ability to update their own personal data. Here is the Django view for displaying user profile and allowing personal details modification:



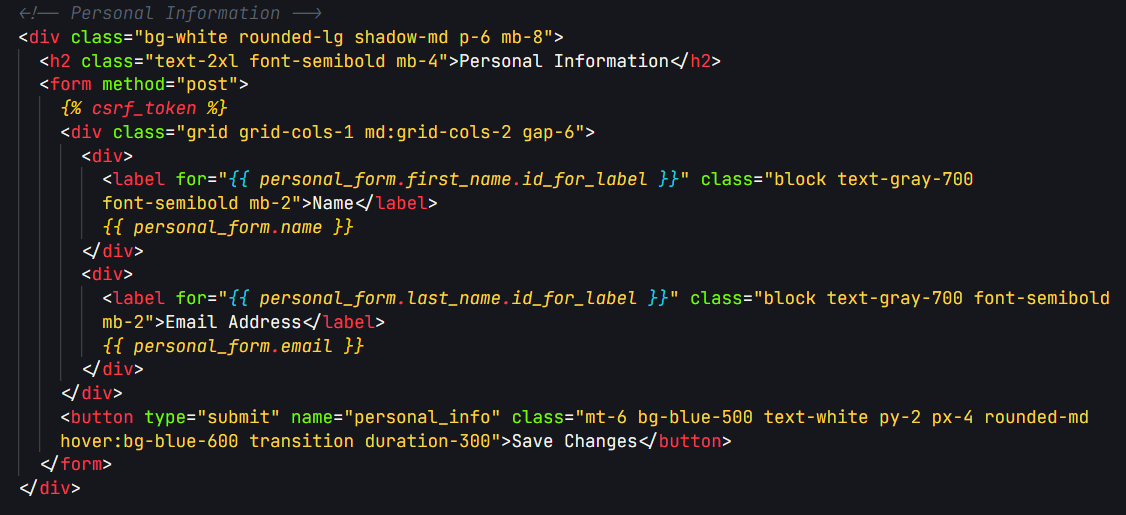
The following form is for allowing the user to change their personal details.

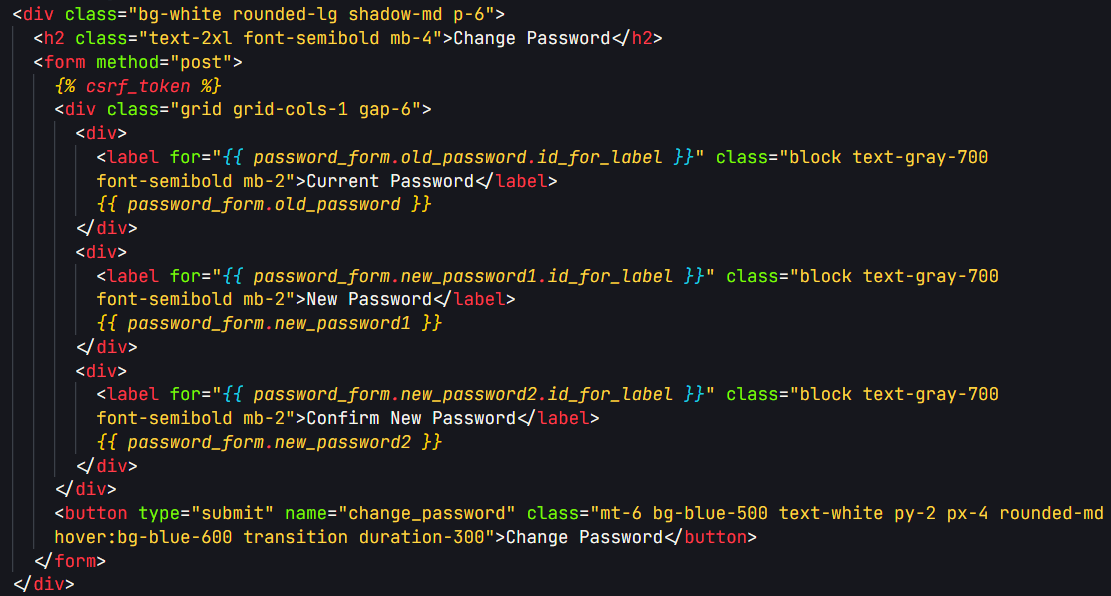


The next form allows the user to change their password:



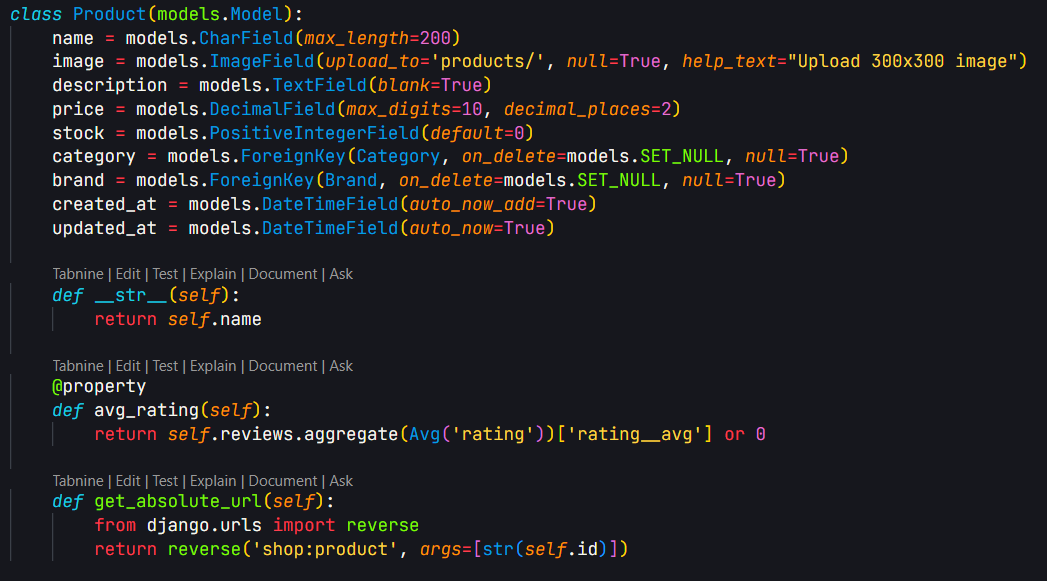
Here are the respective HTML pages containing the forms in the frontend:



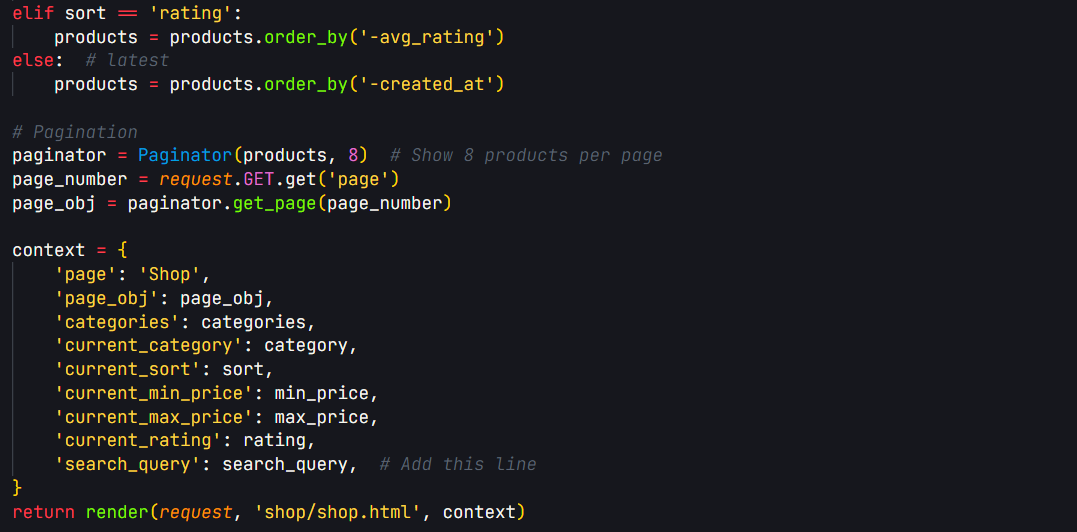
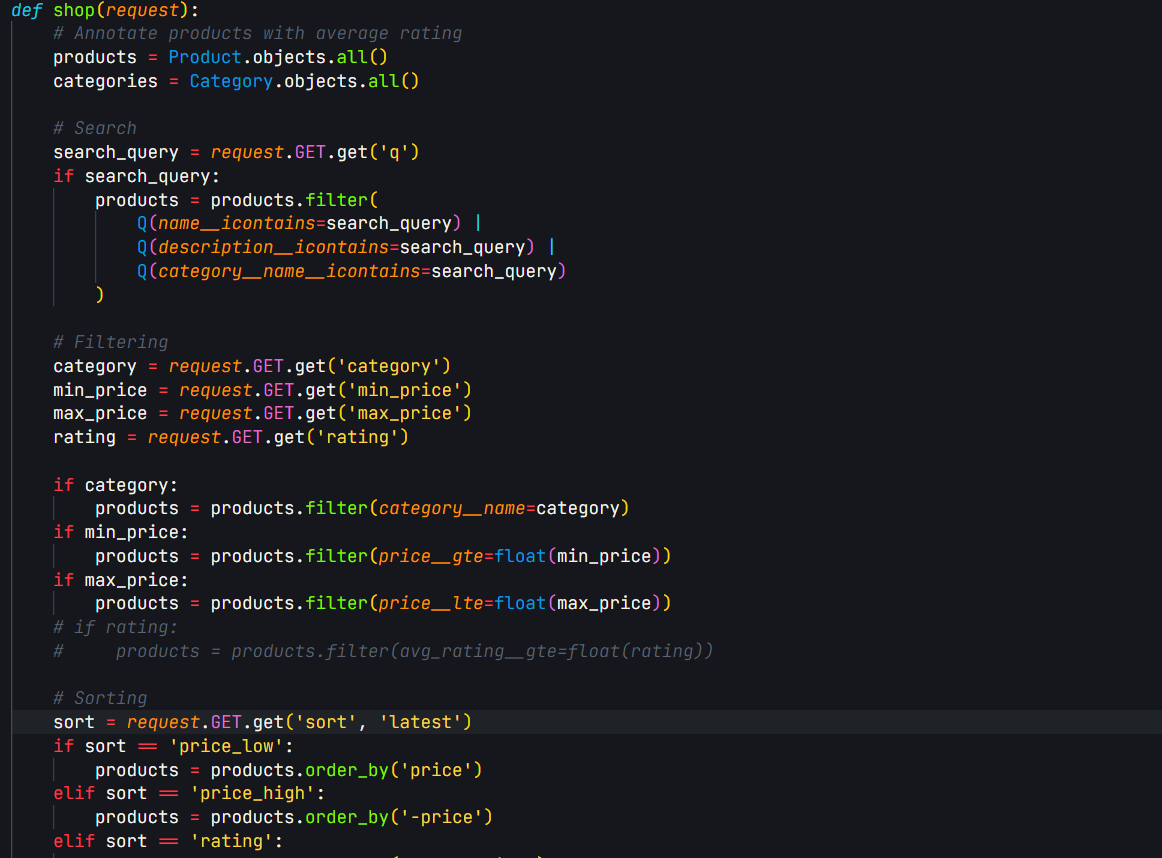


**2. Product Listing and Filtering**

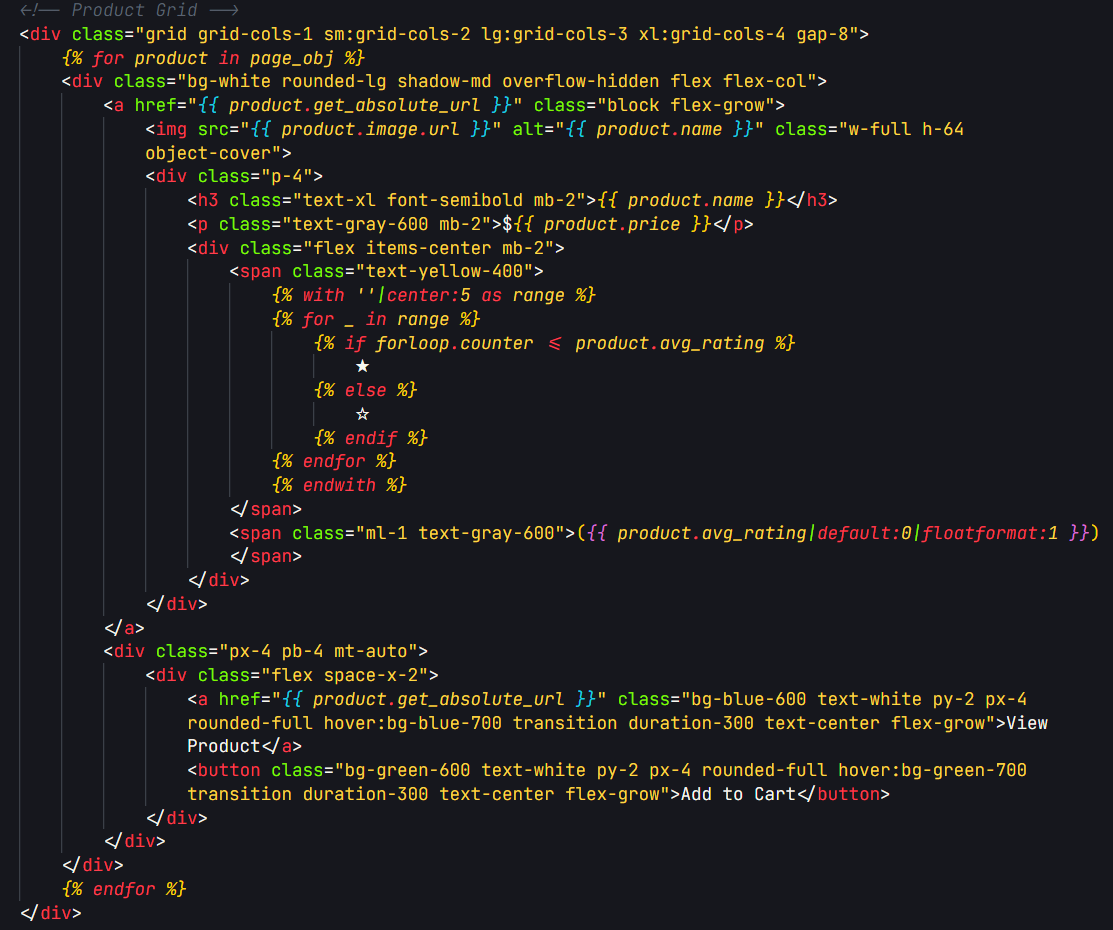
The following is the model (“database table”) that was created to store product details:



Using this model, products can be added and stored to the database. The following view is for product listing and filtering based on various filters such as price, rating among others:



For displaying the products in the frontend, the following is the respective HTML code:



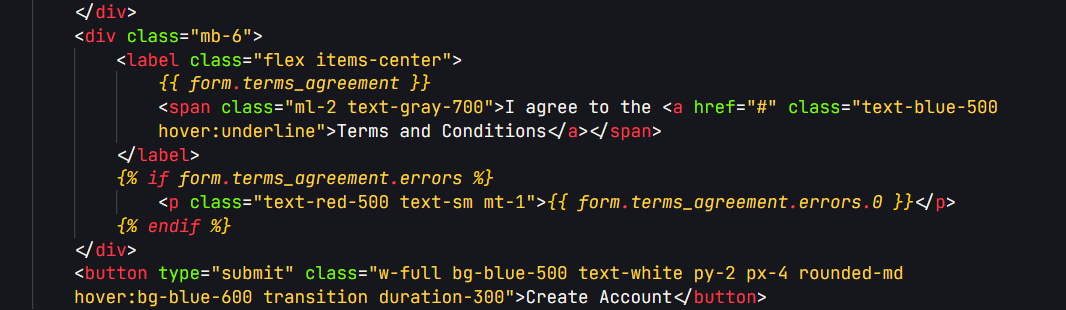
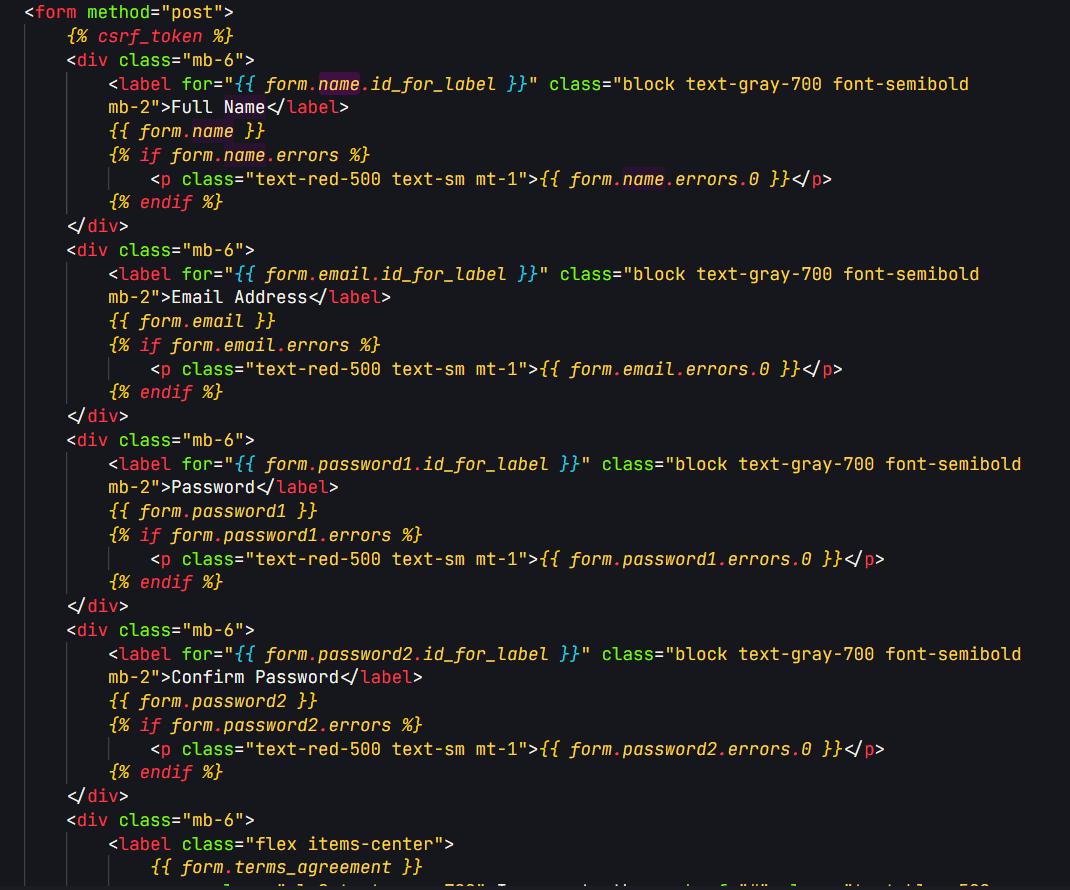
**3. User Interface for login, signup and products**

The user interface adopts a modern structure with the help of modern CSS frameworks such as Tailwind. The following code imports the required stylesheets responsible for the user interface:

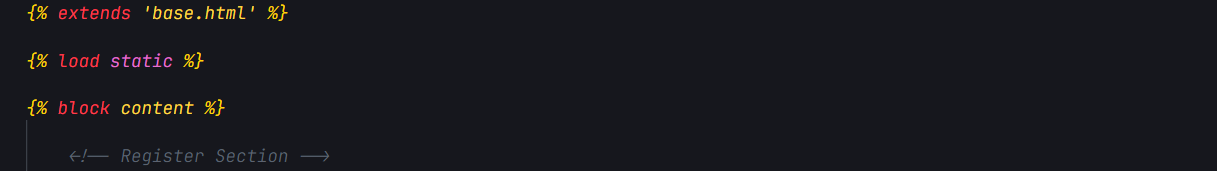


The styles are applied uniformly and universally throughout the entire website to ensure style consistency.

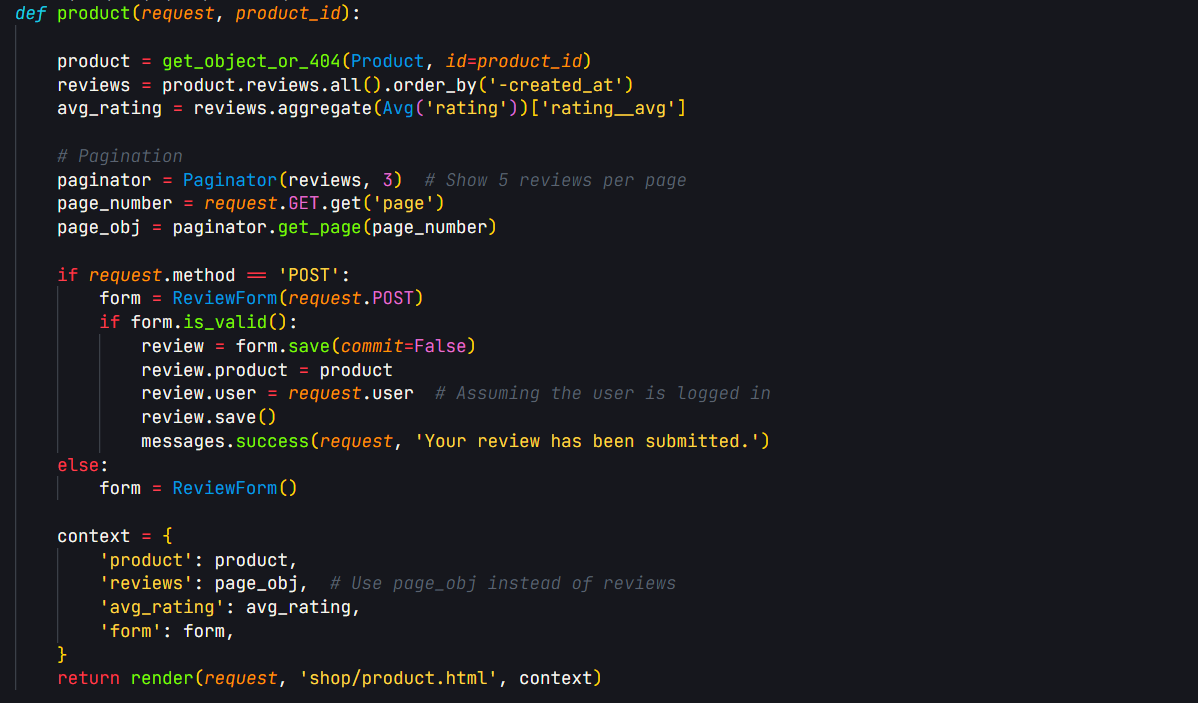
Using Tailwind CSS framework, modern styling is applied and the various forms being the registration and login among others. Below is registration HTML page form:



Using Django’s template structure, each template inherits the base template which contains the consistent styles. For instance, consider the following registration page that extends the base.html file:

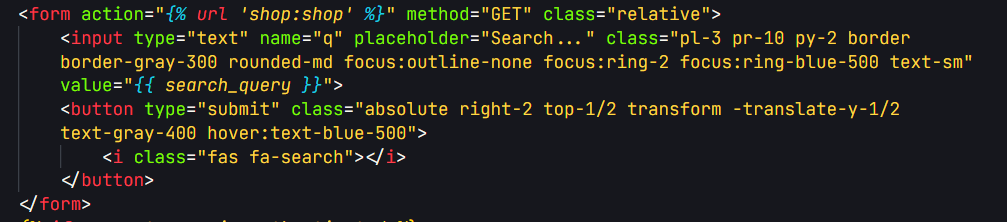


The following is the Django view code for displaying a product’s details:

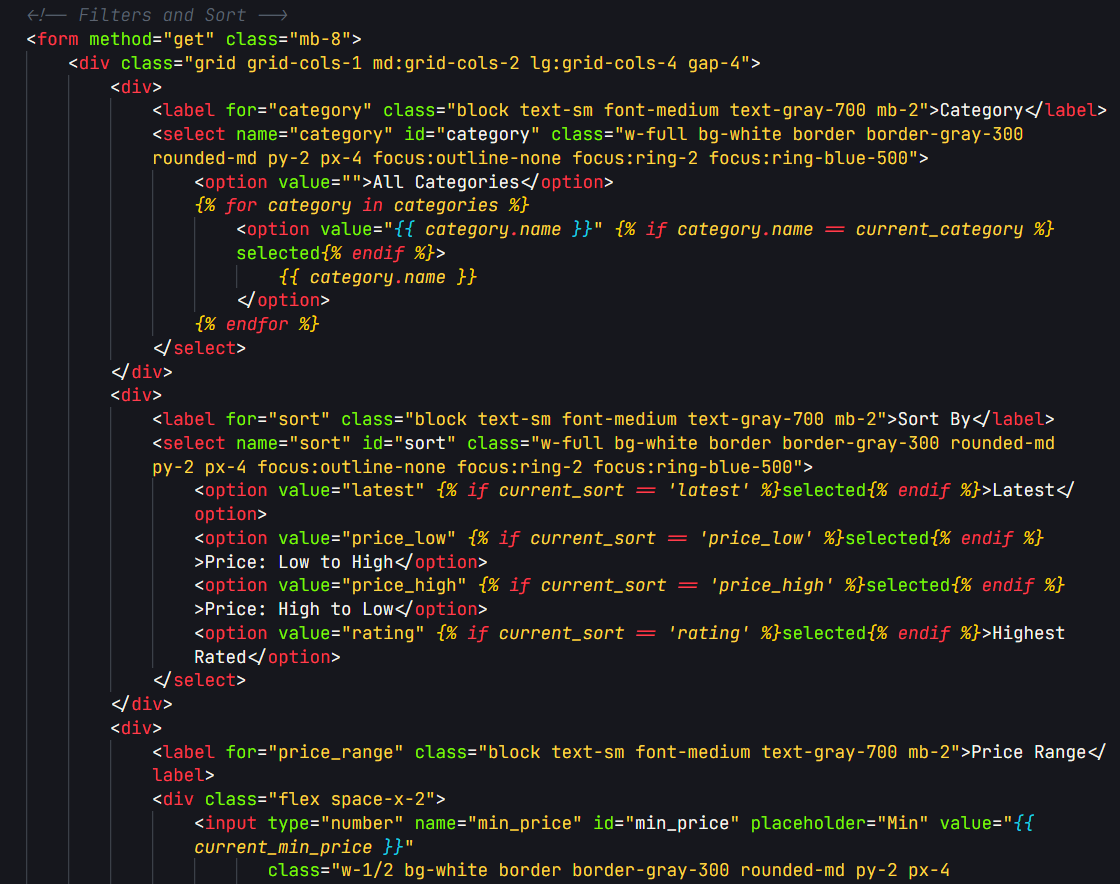


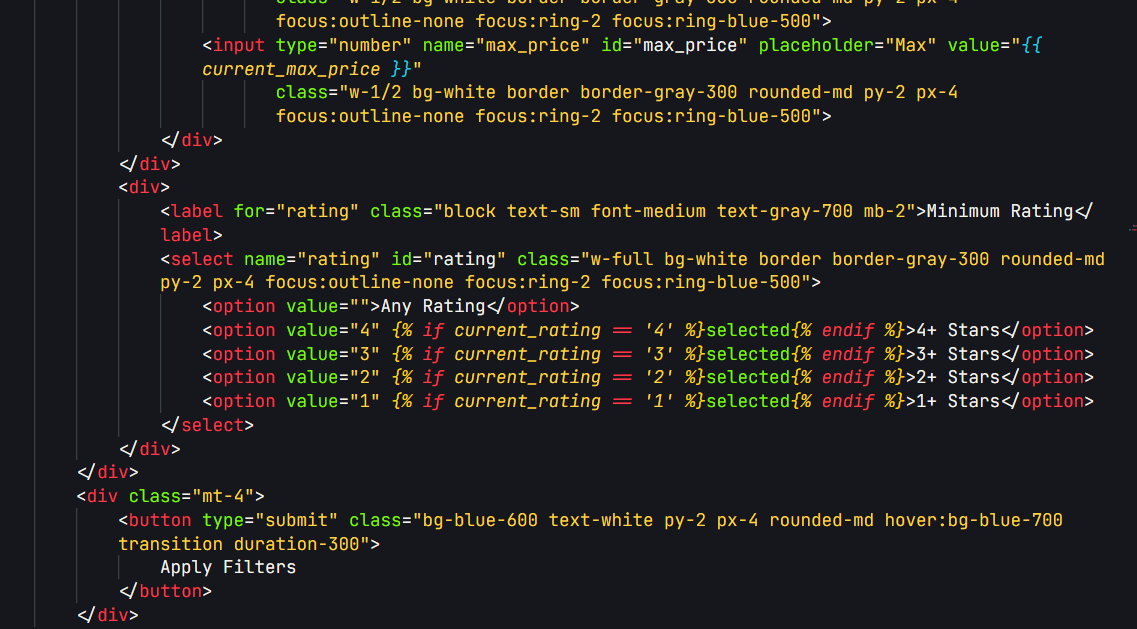
**4. Implementing search and filter functionality**

The search option is displayed in the navigation menu and by using a form, the provided search query is sent to the backend for appropriate query handling and fetching of the results. The following is the HTML code for the search input:



The following is the frontend code for performing product filtering:





**5. Recommendation Engine (Initial Development)**

Next, we are advancing our User and Product models to include more fields that will enable us perform advanced product recommendations to our users providing more custom and personalized recommendations.