Development Process Report: Creating a Django Online Store

Project Overview:

This project's goal was to use Django to create an online store and integrate the fashion.csv file's data to fill the product database. Users of the online store can browse and buy fashion items that are divided into different categories. It would describe the development process, libraries used and a cloud URL that shows a preview of the site.

Development Stages:

1. Project Setup: Initially, the Django project structure was set up with essential components such as models, views, templates, and static files. The project was initialized with version control using Git, and the repository was hosted on GitHub.

2. Data Import: A custom management command was implemented to import data from the fashion.csv file into the Django database. This involved parsing the CSV file, mapping its fields to the Product model attributes, and saving the products into the database.

3. Model and Admin Setup: The Product model was defined to represent the products available in the store. Admin functionality was implemented to manage products via the Django admin interface, allowing for easy CRUD operations on products.

4. Frontend Development: HTML templates were created to render the user interface for various pages such as the home page, product list page, and purchase confirmation page. CSS styles were applied internally for easy edits and to enhance the visual appeal and usability of the online store.

5. Testing: Both behavioural and unit tests were conducted to ensure the functionality and reliability of the online store. Behavioural tests were performed to simulate user interactions and verify that the expected behaviour was achieved. Unit tests were written to test individual components and functions, ensuring that they perform as expected in isolation.

6. Deployment: The project was deployed to the pythonanywhere platform to make the online store accessible over the internet. Continuous integration and deployment (CI/CD) pipelines were set up to automate the deployment process, ensuring that any changes pushed to the repository were automatically deployed to the production environment.

Challenges Faced:

- Circular import issue: During the development process, a circular import issue occurred when importing the Product model in certain modules. This was resolved by restructuring the code to eliminate circular dependencies.
- CSS Styling: Achieving a visually appealing and responsive design posed a challenge, especially when dealing with different screen sizes and devices. Iterative refinement of CSS styles was performed to address this challenge and ensure a consistent user experience across devices.
- Add to Cart Functionality Implementation: Carefully crafting the backend code in views.py was necessary to develop the add to cart feature. This included maintaining session data, making sure functionality worked properly in a variety of settings, and precisely updating the user's cart when things were added. It took testing and improvement to confirm the accuracy of the feature and improve user experience.

Conclusion:

In conclusion, the development of this Django online store involved various stages including project setup, data import, model and admin setup, frontend development, testing, and deployment. By leveraging Django's powerful features and adhering to best practices, we were able to create a functional and reliable online store that provides users with a seamless shopping experience.

Git Repository:

The source code for the project is available on GitHub at: https://github.com/casidiswags/shoppingwebsite.git

Cloud Deployment URL:

The online store is deployed and accessible at: Home - My Website (casidiswags.pythonanywhere.com)

Student Name:

Cassidy Jeremiah Imhanlahimi 52318847