Title: Grocery Store Application

Hardik Sharma

21f1006555@ds.study.iitm.ac.in

Introduction:

Grocery Store App, a web-based platform designed to enhance the efficiency of grocery store management, inventory control, and user transactions. The primary objective is to empower store administrators by facilitating seamless product section management, efficient inventory control, and a user-friendly experience for customers to browse, order, and manage their shopping carts.

Overview:

The application comprises three key components: Section Management, Product Management, and User Transactions.

Data Models:

User Model: Represents user information with attributes such as id, username, email, password, active status, fs_uniquifier, and a many-to-many relationship with the Role model for defining user roles.

Role Model: Defines roles that users can have, featuring attributes like id, name, and description.

Login Form: This represents the user login form with fields including email, password, and remember_me.

Section Model: Represents product sections, including attributes such as section name and description.

Product Model: Defines individual grocery products, including product name, description, rate per unit, unit, stock, and associated section.

UserCart Model: Stores information about items in a user's shopping cart, including username, product details, quantity, and total amount.

Joint Model: Establishes a many-to-many relationship between sections and products.

UserTransaction Model: Tracks user transactions, recording details such as username, product details, quantity, amount, and transaction date.

System Architecture:

The system adopts a client-server architecture, with the client-side developed using Flask to provide interactive interfaces for administrators and users. Flask handles API requests and interacts with the database on the server side.

Functionality:

Section Management: Admins can add, edit, and remove product sections, each capable of containing multiple products.

Inventory Management: Admins can manage product details, organize products into sections, and control attributes such as name, description, rate, unit, and stock.

User Transactions: Customers can browse products, add items to their cart, and complete transactions. Transaction details are stored in the database.

User Interfaces:

Admin Dashboard: Provides administrators with access to section and inventory management through intuitive forms for adding/editing sections and products.

User Dashboard: Allows users to browse products, view details, add items to their cart, complete transactions, and review transaction history.

Additional Features:

Coupon Code: Users can apply coupon codes during transactions.

Celery-Redis: Utilizes Celery-Redis for asynchronous tasks, including sending notifications to users via webhooks.

Mailhog: Sends transaction and interaction reports to users via Mailhog and Celery tasks.

Scheduling Daily Reminder Jobs:

Implements daily reminder jobs to re-engage users, generating personalized alerts for users who haven't recently visited or made transactions.

Monthly Entertainment Report:

Generates and delivers a comprehensive Monthly Entertainment Report to users, providing insights into their interactions with the platform from the previous month.

User-Triggered Async Job:

Enables users to initiate asynchronous jobs for exporting grocery-related data in CSV format. Celery manages the export job, and users receive alerts upon successful completion.

Implementing Caching for Enhanced API Performance:

Optimizes API performance using caching to reduce server load, employing Redis for effective caching, and implementing cache expiry mechanisms to ensure data accuracy.

https://drive.google.com/file/d/1sm3ttALkzX3sSnRBOsHVHQZb5ccfZKq2/view?usp=sharing