Grocery Shopping Application Report

Introduction

The Grocery Shopping Application is a web-based system designed to facilitate the buying of groceries by users. The application is built using Flask for the API, VueJS for the user interface, Jinja2 templates for rendering, and Bootstrap for styling. The database management is handled by SQLite, and Redis is utilized for both caching and background batch jobs, including Celery for asynchronous task processing.

System Architecture

The system follows a multi-user approach with three main roles: Admin, Store Manager, and User. It features a robust Role-Based Access Control (RBAC) system to manage user permissions.

Core Functionalities

User Signup and Login

- Users can sign up and log in using a username and password.
- Authentication is based on JWT (JSON Web Tokens) for secure and stateless communication.
- A suitable user model is implemented to store user-related information.

Admin Login

- Admins have a dedicated login form with similar credentials as other roles.
- Flask Security or JWT-based Token Authentication is used for secure admin access.
- Only one admin account is allowed in the application.

Store Manager Signup and Login (Using RBAC)

- Store managers can sign up and log in with specific credentials.

- Admin approval is required for new store manager sign-ups.
- Requests for approval automatically appear on the admin dashboard.

Section/Category Management (Only for Admin)

- Admins can create, edit, and remove sections/categories.
- Support for multiple languages is implemented using UTF-8 encoding.
- Approval process for store managers' requests to add, edit, or delete categories.

Product Management (Only for Store Manager)

- Store managers can create, edit, and remove products.
- Support for multiple languages for product details.
- Allocation of products to specific sections/categories.
- Request system for admin to add, edit, or remove sections/categories.

Search for Products

- Users can search for products based on section/category or specific criteria like price or manufacture date.

Shopping Cart for User

- Users can add multiple products to their shopping cart.
- The cart can contain products from various categories.

Buy Products

- Display of all available products in a category.
- Users can buy multiple products from one or multiple sections.
- Out-of-stock indication for unavailable products.

- Display of the total amount to be paid for the transaction.

Daily Reminder Jobs

- Scheduled daily reminders sent via Google Chat, SMS, or Email.
- Reminder checks if the user has not visited or bought anything and encourages them to do so.

Monthly Activity Report

- Monthly progress report generated in HTML and sent via email.
- Includes user-specific details such as orders made, total expenditure, etc.

User-Triggered Async Job - Export as CSV (Only for Store Managers)

- Store managers can trigger the export of product details in CSV format.
- Dashboard for store managers to initiate the export.
- Batch job completion alert sent once the export is done.

Performance and Caching

- Caching implemented strategically to enhance performance.
- Cache expiry mechanism in place.
- Focus on optimizing API performance for better user experience.

Deployment and Compatibility

The application is designed to run on Linux-based systems, and compatibility is extended to Windows OS using Windows Subsystem for Linux (WSL).

Conclusion

The Grocery Shopping Application provides a comprehensive solution for users, store managers, and administrators. It leverages modern web technologies and follows best practices for security, performance, and user experience. The application can be easily set up on a local machine for demonstration purposes, creating a seamless grocery shopping experience.