

Report of My Project ,Grocery Store- System Design and Database Models

Author Part

Name- Sanjeev Kumar Yadav

Roll number – 21f2001185

Email - 21f2001185@ds.study.iitm.ac.in

Introduction

This report outlines the design of the database models and the overall system architecture for the application.

Database Models

User Model

The User model represents users of the application. It includes fields such as id, username, email, password, active, and roles. Users can have multiple roles, which are defined in the Role model.

Role Model

The Role model defines the roles that users can have in the application. It includes fields such as id, name, and description. Roles are associated with users through a many-to-many relationship defined by the RolesUsers association table.

Category Model

The Category model represents product categories in the application. It includes fields such as id, name, description, creator_id, is_approved, is_edited, and is_deleted. Categories can have multiple products associated with them.

SecondaryCategory Model

The SecondaryCategory model represents subcategories within a primary category. It includes fields such as id, name, description, and primaryctg, which is a foreign key referencing the primary category.

Product Model

The Product model represents products available in the application. It includes fields such as id, name, quantity, sold_quantity, rate, date, unit, category_id, creator_id, and is_approved. Products are associated with categories and users.

Cart Model

The Cart model represents user carts in the application. It includes fields such as id, name, quantity, rate, Total_price, Product_id, Category_id, and User_id. Carts contain products that users have added for purchase.

API Design

Products Resource (/api/product_data):

GET: Retrieves a list of products. If the user is a customer, all products are retrieved. If the user is not a customer, only approved products or products created by the current user are retrieved.

POST: Creates a new product. Only accessible to users with the "Manager" role. Requires parameters for the product's name, quantity, rate, unit, date, and category ID. Upon successful creation, the cache is cleared.

CategoryProducts Resource (/api/category_products/<int:category_id>):

GET: Retrieves a list of products belonging to a specific category. Customers can view all approved products in the category, while other users can view all products in the category.

EditProduct Resource (/api/EditProduct/<int:product_id>):

GET: Retrieves details of a specific product by its ID.

POST: Edits an existing product. Users with the "Manager" role can edit products within their categories. Requires parameters for the product's name, quantity, rate, unit, date, and category ID. If the category ID is changed, a new product is created, and the old product is deleted. Upon successful edit or deletion, the cache is cleared.

DELETE: Deletes a specific product by its ID. Also deletes any associated cart items. Upon successful deletion, the cache is cleared.

Categorys Resource (/api/section_data):

GET: Retrieves a list of categories. Admins can view all categories, including those pending approval. Managers can view categories they created and those created by admins. Customers can view approved categories.

POST: Creates a new category. Requires parameters for the category's name and description. Admins can create approved categories, while managers can create unapproved categories.

Video Link-

<https://drive.google.com/file/d/1cX-jltFT2gvyadDw2FKSzpAZzD5HsVOD/view?usp=sharing>

