Modern Application Development -I
Grocery Store Application

Author

BhavyaSingh
21F3001937
21f3001937@ds.study.iitm.ac.in

Link to presentation:
https://drive.google.com/file/d/1q3VdCgkwggkrIPRp-8IU-Y_zDwZor8ks/view?usp=sharing

Architecture, Features and Technologies used:

A brief description about the general structure of the application:

- **Application**: This directory has all the necessary python files such as controllers for the flask application and is responsible for the overall functionality of the application
- **Static**: This directory contains the static components of any webpage such as images, HTML and CSS files.
- **Templates**: This subfolder of the static folder has all the necessary html files which are responsible for the frontend of the application
- **app.py**: The relevant function, class, or variable is imported into this Python file from the module application. In a virtual environment, it is also in charge of importing Flask and executing the minimal server needed for the application.
  This application uses basic HTML and CSS for frontend while using flask and flask-SQLAlchemy for the backend.

The Application directory:

The application module holds the initializing empty python file "__init__.py", config.py (holds the basic configuration details of our application), admin_controllers.py and customer_controllers.py (describes all the endpoints), database.py and models.py.

- Admin_controllers.py and customer_controllers.py: these files include the endpoints of the application while using inbuilt and Flask libraries such as SQLAlchemy, Flask, matplotlib, etc.

Database Design:

- **models.py**: the tables defined in this file are: Product, Categories, Cart and User.

The Tables and Relationship models defined are as follow:

- **Product**: This table has the columns:
  1. Product_id: Integer, auto increment, primary key. This column is used to filter out unique products
  2. Name: String, indicates the name of the product. Users are shown product names and their choice is handled on the backend using the product id of that particular product

    3. Category: String, indicates the category of the product. Products are displayed and managed in a sorted manner according to their categories.
    4. Price: Integer, admin, as well as users can view prices and search according to price.
    5. Manufacturing date: String
    6. Expiry date: String

- users: This table has the following columns with their properties assigned as follows:
    1. User_id: Integer, primary key. This column if related to the carts table and is unique to each user.
    2. Name: String, chosen by user.
    3. Password: String, chosen by user, used for authentication

- Categories: This table has the following columns with their properties assigned:
    1. Id: Integer, primary key, id of the particular category
    2. Name: String, name of a category, unique
    3. Sold: Integer, indicates how many products of that category are sold
    4. Total: Integer, total number of products of that category previously or currently in the inventory

- Cart: This table has the following columns with their properties assigned:
    1. User_id: related to user id in the users table
    2. Product_id: related to product id in the products table
    3. Quantity: number of items of a particular user for that product