Author
JAMALLA KRISHNA CHARAN
21F2001078
21f2001078@ds.study.iitm.ac.in
I'm krishna charan jamalla an aspirated data scientist

Description

This is a grocery store web application with operations for different set of users where customers can select products by searching based on name,price,date of manufacture and buy the selected products from multiple catogories and admin can add, edit, delete specific product or category

Technologies used

- **sqlalchemy.ext.declarative:** The SQLAlchemy declarative base class is used to create object-relational mapper (ORM) models that map to database tables. This makes it easy to interact with the database using Python objects.
- **flask_sqlalchemy:** The Flask-SQLAlchemy extension provides a convenient way to use SQLAlchemy with Flask. It includes features such as automatic database connection management and session management.
- **sqlalchemy:** SQLAlchemy is a popular Python library for database interaction. It supports a wide range of database backends, including MySQL, PostgreSQL, and SQLite.
- **flask:** Flask is a lightweight Python web framework. It is used in the code to create a web application that can be used to interact with the database.
- **matplotlib.pyplot:** Matplotlib.pyplot is a Python library for data visualization. It is used in the code to generate plots of data retrieved from the database.
- **Database.database:** This is a custom Python module that provides functions for interacting with the database. It is used in the code to perform database operations such as queries and inserts.
- **HTML/CSS:** HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) are fundamental technologies for creating and styling web pages.
- **Bootstrap:** Bootstrap is a front-end framework for web development that provides pre-designed CSS styles and JavaScript components to build responsive and visually appealing web applications.
- **Jinja2:** Jinja2 is a templating engine for Python used with Flask to generate dynamic HTML content.

DB Schema Design

User Table:

- `id` (PK, INT, NOT NULL, UNIQUE, Autoincrement), `name` (STRING, NOT NULL, UNIQUE), `password` (STRING, NOT NULL), `telnum` (INT, NOT NULL), `mail` (STRING, NOT NULL), `dob` (STRING, NOT NULL), `gender` (STRING, NOT NULL), `admin` (BOOLEAN, NOT NULL, Default: False)

Product Table:

- `id` (PK, INT, NOT NULL, UNIQUE, Autoincrement), `dom` (STRING, NOT NULL), `name` (STRING, NOT NULL, UNIQUE), `stock` (INT, NOT NULL), `price` (INT, NOT NULL), `category` (STRING, NOT NULL, UNIQUE), `units` (STRING, NOT NULL)

Category Table:

- `id` (PK, INT, NOT NULL, UNIQUE, Autoincrement), `doa` (DATE, NOT NULL, Default: Current Date), `type` (STRING, Nullable), `category` (STRING, NOT NULL)

Cart Table:

- `id` (PK, INT, NOT NULL, UNIQUE, Autoincrement), `products` (INT, NOT NULL), `count` (INT, NOT NULL), `price` (INT, NOT NULL), `date_added` (DATE, NOT NULL, Default: Current Date)

Purchase Table:

- `id` (PK, INT, NOT NULL, UNIQUE, Autoincrement), `p_id` (INT, NOT NULL), `count` (INT, NOT NULL), `price` (INT, NOT NULL), `date_added` (DATE, NOT NULL, Default: Current Date), `user_id` (INT, NOT NULL), `category` (STRING, NOT NUL)

API Design

- sqlalchemy.ext.declarative.declarative_base() creates a base class for declarative SQLAlchemy models.

- flask_sqlalchemy.SQLAlchemy() initializes SQLAlchemy for use with Flask.
- sqlalchemy.create_engine() creates a SQLAlchemy engine for connecting to a database.
- sqlalchemy.or_() creates a SQLAlchemy logical OR expression.
- sqlalchemy.String() creates a SQLAlchemy string data type.
- sqlalchemy.distinct() creates a SQLAlchemy function to return the distinct values of a column.
- sqlalchemy.sql.expression.cast() casts a SQLAlchemy expression to a different data type.
- Database.database.query() performs a database query.
- Database.database.insert() inserts a new row into a database table.

Architecture and Features

Architecture:

The grocery store application is functionally divided into two folders code and instance. The code part consists controllers folder contains controller.py to store the controller code for every end point,database folder contains database.py for database models to access the database,static folder for all the images and templates folder for all the html files, finally app.py. The instance contains the sqllite database instance

Features:

- **User Authentication:** Users and admins can register and log in securely.
- **Product Search:** Users can search products by various criteria.
- **Shopping Cart:** Users can add products to their cart for easy checkout.
- **Admin Controls:** Admins can manage products and categories.
- **Data Visualization:** Matplotlib.pyplot is used for data visualization.
- **Responsive Design:** Bootstrap ensures a user-friendly interface on various devices.

User/Admin can login/register using login page by vadilidating credentials passed through form and details store in database Users can search the products based on the price ,date of manufature,name,category of the product where search term checked with the entries in the database and render the matched products User can select the product quantity and add to cart to buy the products that are added to cart at once,These purchased products are stored in database.Admin can add,edit,delete the product or category the changes are stored in database

Video

https://drive.google.com/file/d/165w285T1HZ4P2goGzudxBlaJ9qoq5RTG/view?usp=sharing