GROCERY STORE REPORT

Author:

SHWETA JHA

21f2001336

21f2001336@ds.study.iitm.ac.in

A Third Year BCA Student, I like Poetry writing, listening music and Dancing and always tends to have a positive attitude for every situation encountered in life.

Description:

Basic Idea is to let Manager can create Categories and Products for customer and customer can buy any product anytime.

So, this could be done by having 4 Tables.

Storing information about User, Categories, Products and Booking and connected by relations (Backend).

Technologies used:

Packages Used: Flask, flask_login, flask_sqlalchemy

Flask used as a Web framework by Python Packages, used for developing Web Applications easily (It's a good place to start for beginners).

flask_login is used for remember the login information of our flask Application user and also validate the user.

flask_sqlalchemy is used for Database Linking and querying the database to retrieve information from the database and present to the Viewer.

DB Schema Design:

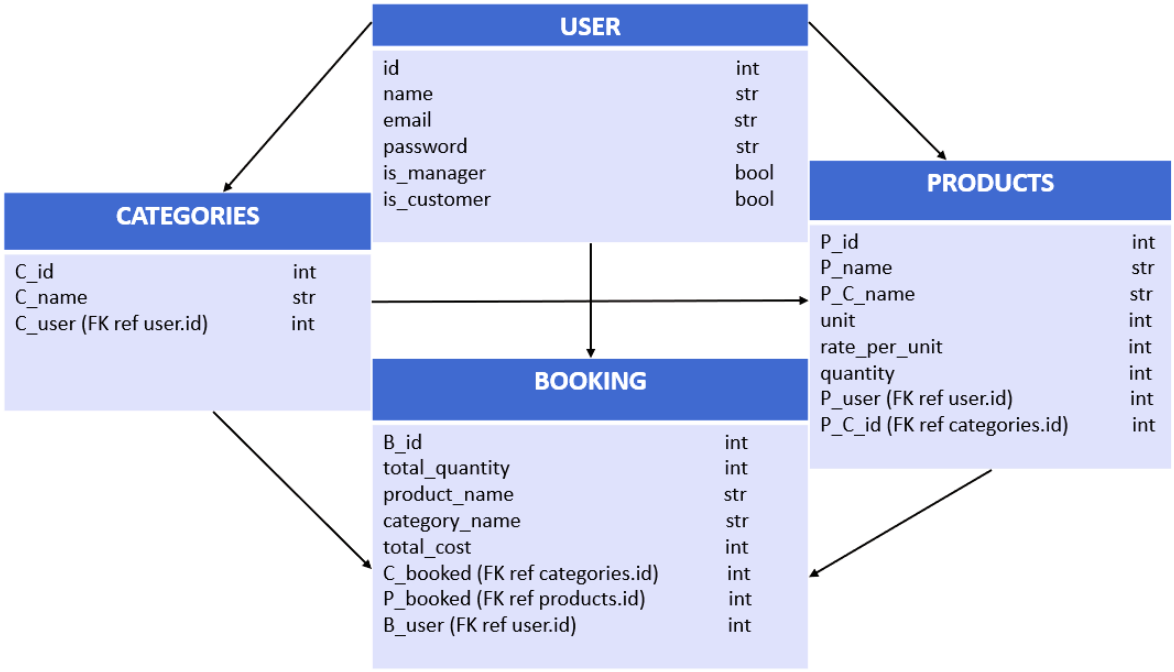
A one-to-many relationship is established between User and Categories, User and Products, User and Booking, Categories and Products, Categories and Booking, Products and Booking, connected by id so that each and every Categories and Products can be created by manager and, each and every Products with Category can be booked by customer.

All Categories information are stored in Categories Database and each Category is uniquely identified by C_id

and also each Category has C_user(reference for user.id). Which would let us identify each Category's parent are User and in turn we would also know which Category created by which manager.

All Products information are stored in Products Database and each Product is uniquely identified by P_id and also each Product has P_user(reference for user.id) and each Product has P_C_id(reference for categories.id). Which would let us identify each Product's parent are User and Category and in turn we would also know which Product created by which manager and are of which Category.

All Booking information are stored in Booking Database and each Booking is uniquely identified by B_id and also each Booking has B_user(reference for user.id), each booking has P_booking(reference for products.id) and also each Booking has C_booking(reference for categories.id). Which would let us identify each Product's parent are User and Category and in turn we would also know which Product created by which manager and are of which Category.



API Design:

GET -> 1. Logout 2. My booked product

POST -> 1. Sign UP 2. Login 3. Buy product 4. Add category 5. Add product

PUT -> 1. Update category 2. Update product

DELETE -> 1. Delete category 2. Delete product

Architecture and Features:

Each Manager can create their (Categories and Products) and could perform CRUD operations on all the Categories and Products, customer can by the Products.

All the Authenticated APIs as well as unauthenticated API's are in store.py', DB Models are in database.py' and all `jsonify API's` are in 'api.py'.

All the HTML documents are stored inside the templates folder. Around 9 html Webpages have been designed for carrying out various operations in the grocery store app with the help of 13 @store.route() controllers.

Features Implemented: Manager Login and SignUp, Customer Login and SignUp, Categories and Products can be created by manager and have also provided function of updation and deletion.

Customer can buy any product any time.

CRUD operations successfully working.

Review: APIs for Interaction to the Database and Input Validation also done.

Video: **https://drive.google.com/file/d/1iw-XqXDxncMIs4afwV6-EzhykMYf3CJZ/view?usp=sharing**

***Watch VIDEO at 1.5X