Author

Name: Thadeshwar Jay Jiteshbhai
Roll number: 22dp3000018
student email: **22dp3000018@ds.study.iitm.ac.in**
<< add a couple of lines about you>

Description

I have undertaken this project with the goal of develop web-based inventory management system for grocery store. Primary objective of this project is to create a user-friendly platform that enable people to buy product from grocery store online

My project focus on developing web application using flask by which user can buy products of grocery store online and admin can manage products and category on web application

Technologies used

Flask framework: used for building application's backend.
Flask-Security: For proper login system.
SQLite database: For data storage and retrieving
Bootstrap: For styling
Jinja 2: For Template
Flask-SQLAlchemy: For Database Integration
Flask RESTful: To design and implement Restful API endpoints.

DB Schema Design

roles_users:
user_id(Integer), ForeignKey(useruser_id)
role_id(Integer), ForeignKey(roleid))

user:
user_id (Integer), autoincrement, primary key)
full_name (String)
email (String, unique)
password (String(255))
active (Boolean)

role:
id (Integer, primary key)
name (String(80), unique)
description (String(255))

catagory:
catagory_id (Integer, primary key)
name (String(80), unique)
description (String)

product:
product_id (Integer, primary key)

name (String)
expiry_date (String)
price (Integer)
unit (Integer)
catagory_id (Integer,ForeignKey(catagory.catagory_id))
image_name (String)
description (String)
quantity (Integer)

cart:
cart_id (Integer, primary key)
user_id (Integer,ForeignKey(user.user_id))
product_id (Integer,ForeignKey(product.product_id)
quantity (Integer)
price (Integer)

order:
order_id (Integer, primary key)
user_id (Integer,ForeignKey (user.user_id))
total (Integer)
order_date = (DateTime)
delivery_date (Numeric)
status (String)
receiver (String(50))
address (String)
city (String)
state (String)
pin_code (Integer)

Order_Details:
order_details_id (Integer, primary key)
order_id (Integer, ForeignKey (order.order_id))
product_id (Integer,ForeignKey (product.product_id))
product_name (String, ForeignKey (product.name))
quantity (Integer)
per_unit_price (Integer)
price (Integer)

API Design

Elements Created:

- Category API:
    - Endpoint 1:/category/<string:name> (GET) - Get details of a specific category.
    - Endpoint 2: /api/category (POST)- Create a new category.
    - Endpoint 3: **/category/<string:name>** (PUT) - Update an existing category.
    - Endpoint 4: /category/<string:name> (DELETE) - Delete a category.

Implementation:

Categories API:
/category/<string:name> (GET) – Gives you information about provided category by provided name
/api/category (POST) – Creates category
/category/<string:name> (PUT) – Updates existing category

/category/<string:name> (DELETE) – Deletes category

YAML File:
Yaml file is present inside project folder. I have created YAML file for CRUD operation on Category as well as Product.

Architecture and Features

My project follows model-view-controller architecture. For this project I have created one file “app.py”. All the controllers are reside in app.py. All the templates are present inside “Template” folder. And YAML file and database file is present inside project folder.

Features Implemented:

I have implemented following features in my application:

- User registration and Authentication: I have use flask security to make proper login system
- Role-based Access Control: I have implemented role-based access control, allowing admin to manage categories and products while restricting regular users from these actions
- Category Management: Use can view list of category and Admin can add and delete Category.
- Shopping cart: Users can add and remove products from their shopping cart
- Search Functionality: Users and admin can search for products or category using keywords, and the application displays relevant results
- Api For CRUD operation on category:
- Styling:

Video

https://drive.google.com/file/d/1nwLzuTaKaSchxjQJ15guRUtfdE88zcJC/view?usp=drive_link