Author

Gomati Manikandan Iyer
21f3002227
21f3002227@study.iitm.ac.in
I am Gomati Iyer , this is my project submission for MAD-1 Project course for the May 2023 term:
Title : **Grocery Store Application**

Description

The application is an online grocery store management system built using Flask, HTML, and SQLite. It features customer and admin login functionalities with registration, dashboard views for both customers and admins, category and product management, shopping cart, and order processing. Admins can manage categories, products, and view orders, while customers can explore categories, add products to their cart, and place orders.

Technologies used

The technologies and libraries used in the app include:
1. **Flask**: A Python web framework used to build the backend of the application.
2. **HTML**: HyperText Markup Language for creating the structure and content of web pages.
3. **CSS**: Cascading Style Sheets for styling the web pages.
4. **Bootstrap**: A popular CSS framework for responsive design and pre-styled components.
5. **SQLite**: A lightweight relational database management system for storing application data.
6. **Jinja2**: A template engine used to render dynamic content in HTML templates.
7. **Flask-Session**: An extension for managing user sessions in Flask applications.
8. **Flask-Flash**: An extension for displaying flash messages to users.

DB Schema Design

Table: customers
- Columns: customer_id (Primary Key), name, email, password
- Constraints: email must be unique

Table: categories
- Columns: category_id (Primary Key), name
- Constraints: All categories should have distinct names.

Table: products
- Columns: product_id (Primary Key), product_name, category_id (Foreign Key), available_quantity, unit, brand_name, price
- Constraints: Foreign Key constraint on category_id referencing categories

Table: cart
- Columns: id (Primary Key), user_id (Foreign Key), product_id (Foreign Key), quantity
- Constraints: Foreign Key constraint on user_id referencing customers, Foreign Key constraint on product_id referencing products

Table: orders
- Columns: order_id (Primary Key), customer_id (Foreign Key), order_date, total_price
- Constraints: Foreign Key constraint on customer_id referencing customers

Each table's schema is designed to store specific types of data. Customers and admin users are stored in their respective tables. Categories and products are related, with products having a foreign key reference to their category. Cart stores the items added by users for purchase. Orders and order_details are used to store information about orders and the products within them. Foreign key constraints maintain data integrity by ensuring that references to other tables are valid.

API Design

1. **User Authentication and Registration**: An API for user registration and login was implemented using Flask's session management and SQLite for storing user data securely.

2. **Admin Dashboard**: The admin has access to features like adding categories, adding products, and viewing orders. These functionalities are provided through different routes, each handling the corresponding tasks.

3. **Customer Dashboard:** Customers have the ability to browse categories, explore products, add products to their cart, and place orders. Their interactions are managed through session-based cart management.

4. **Cart Management**: An API was built for adding products to the cart, displaying cart contents, and placing orders. Cart data is stored in the SQLite database for each user.

5. **View Orders**: Admins can view all orders placed by customers, including details such as customer names, order dates, and total prices.

6. **Search Functionality**: Implemented a search feature to search for categories and products using SQL queries to filter and retrieve relevant data.

Architecture and Features

The app's main operations and settings are in the app.py file. This setup keeps things organized .The templates, written in HTML, are kept in a folder called "templates" and help in showing information to users. Static files like styles and images are in the "static" folder.
User roles differentiate between customers and administrators, each with their respective dashboards.Product categories and their associated products are managed dynamically through the admin dashboard, allowing addition, deletion, and modification. Customers can explore and add products to their cart, and administrators can view orders placed by customers.

Video

Link : Video demo