

Author

Shashank Kumar

22f1000698

22f1000698@ds.study.iitm.ac.in

A creative and Curious individual, passionate about continuous learning and exploring new horizons. With a flair for drawing and photography, I aspire to innovate and make a mark in the world of AI/ML.

Description

A Flask-based grocery web app with separate customer and store manager login with CRUD operations for items and item categories, allowing customers to add items to their cart and place orders.

Technologies used

Flask: Micro web framework for building the web application. **Flask-SQLAlchemy:** Efficient database management using SQLAlchemy with Flask. **Flask-RESTful:** API development. **SQLite3:** for database creation and storage. **Matplotlib:** For creating bar graphs and line graphs on Summary page. **base64:** to encode graph data. **Requests:** To make HTTP requests to interact with resources. **datetime:** To save order's date and time. **os:** To handle file paths and interact with the operating system. **Bootstrap:** Front-end styling. **HTML:** Template structuring. **jinja3 :** To render dynamic content on HTML templates **Google Material Icons:** Used through CSS links for cart, order, profile etc icons.

DB Schema Design

❖ Table: customers

Columns:

- id (Integer, Primary Key): Unique identifier for each customer.
- name (String(50), Not Null): Customer's name.
- username (String(50), Unique, Not Null): Unique username for login.
- email (String(100), Unique, Not Null): Customer's email address.
- password (String(100), Not Null): Encrypted password for authentication.
- phone (String(20), Not Null): Customer's phone number.
- address (String(200), Not Null): Customer's address details.

Purpose: Stores customer details for registration and login.

❖ Table: store_managers

Columns:

- id (Integer, Primary Key): Unique identifier for each store manager.
- name (String(50), Not Null): Store manager's name.
- username (String(50), Unique, Not Null): Unique username for login.
- email (String(100), Not Null): Store manager's email address.

- password (String(100), Not Null): Encrypted password for authentication.
- store_name (String(100), Unique, Not Null): Name of the store managed by the store manager.
- address (String(200), Not Null): Store manager's address details.
- phone (String(20), Not Null): Store manager's phone number.

Purpose: Stores store manager details for registration and login.

❖ **Table: items**

Columns:

- id (Integer, Primary Key): Unique identifier for each item.
- name (String(100), Not Null): Name of the item.
- price (Float, Not Null): Price of the item.
- unit (String, Not Null): Unit of the item (e.g., kg, piece, etc.).
- seller (String(100), Not Null): Name of the seller for the item.
- manufacture_date (String(100), Not Null): Date of item manufacture.
- quantity (Integer, Not Null): Quantity of the item available.
- category_id (Integer, Foreign Key): References the category that this item belongs to.

Purpose: Stores information about grocery items available for purchase.

❖ **Table: item_categories**

Columns:

- id (Integer, Primary Key): Unique identifier for each item category.
- name (String(255), Not Null): Name of the item category.
- manager_id (Integer, Foreign Key): References the store manager responsible for this category.

Purpose: Stores item categories, each managed by a specific store manager.

❖ **Table: cart**

Columns:

- id (Integer, Primary Key): Unique identifier for each cart item.
- item_id (Integer, Unique, Not Null): Unique identifier of the item in the cart.
- customer_id (Integer, Foreign Key, Not Null): References the customer who owns the cart.
- quantity (Float, Not Null): Quantity of the item in the cart.
- price (Float, Not Null): Price of the item in the cart.

Purpose: Stores items added to the cart by customers for future checkout.

❖ **Table: orders**

Columns:

- id (Integer, Primary Key): Unique identifier for each order.
- item_id (Integer, Not Null): Identifier of the ordered item.
- item_name (String(100), Not Null): Name of the ordered item.
- order_date (String(100), Not Null): Date of the order.
- quantity (Float, Not Null): Quantity of the ordered item.
- price (Float, Not Null): Price of the ordered item.
- customer_id (Integer, Foreign Key, Not Null): References the customer who placed the order.

- `manager_id` (Integer, Foreign Key, Not Null): References the store manager handling the order.

Purpose: Stores information about orders placed by customers and aids store managers in managing and analysing transactions and inventory.

API Design

- The API, built with Flask and Flask-RESTful, manages items and item categories in the grocery web app. It enables users to view, add, update, and remove items and categories. Additionally, the APIs support CRUD operations for sections and products, along with fetching data for displaying sections and products.

Architecture and Features

- The project is organised with a main "app.py" file and separate modules for API, controllers, configuration, and database in the "applications" directory. The "dbDir" folder holds the "grocery.sqlite3" database. The "static" folder contains the "images" directory for static files, and the "templates" folder holds HTML templates. This structured approach ensures modularity and easy maintenance.

- **app.py**

- **applications/**

- **api.py**

- **controllers.py**

- **config.py**

- **database.py**

- **models.py**

- **dbDir/**

- **grocery.sqlite3**

- **static/**

- **images/**

- **templates/**

- **Default Features:** View Available Items, Add to Cart, View Item Details, view final amounts, place orders, Search Items based on date, price, item name etc.
- **Additional Features:** View Past Orders, Increase Cart Quantity by multiple clicking, Store Manager's Sold Items List.

Video

Link: <https://drive.google.com/file/d/1FCEji-6CNI7MouiUAMNkyVS3DYPdOHgs/view?usp=sharing>