

Project Report: Grocery Store Application

Introduction

This project report presents a detailed overview of the Grocery Store Application developed for the course **Modern Application Development 1** at IIT Madras. The application is designed to be a multi-user platform, consisting of an admin/store manager and regular users. The primary purpose of the app is to facilitate the buying and selling of groceries. Users can purchase multiple products from one or multiple sections, while the store manager can manage sections, categories, and products.

Important Points

1. The application is a multi-user platform with an admin and regular users.
2. **The admin login details are as follows:**
 - a. **Email: sudeep@gmail.com**
 - b. **Password: sudeep**
3. A user account with the following details already exists:
 - a. **Email: user@gmail.com**
 - b. **Password: user1**
4. The database file 'db.sqlite3' contains the admin login credentials. **Do not delete this file as it is essential for the proper functioning of the application.**

Tech Stack

The Grocery Store Application is built using the Flask framework, a lightweight web application framework in Python. The database management is handled through SQLAlchemy, which provides an Object-Relational Mapping (ORM) system, simplifying interactions with the database. The front-end is designed using HTML templates with the help of Flask's built-in templating engine, and CSS for styling.

The tech stack includes:

- Python (programming language)
- Flask (web application framework)
- SQLAlchemy (ORM for database management)
- HTML (front-end markup language)
- CSS (front-end styling)

Tables in the Database

1. Product Table

The Product table represents individual grocery items available in the store. It contains the following columns:

id: The primary key for each product, which uniquely identifies it.

name: The name of the product, represented as a String with a maximum length of 100 characters.

quantity: The quantity of the product available in stock, represented as an Integer.

price: The price of the product, represented as a Float.

section_id: A foreign key that links the product to the corresponding section in the Section table.

2. Section Table

The Section table represents different sections or categories of products available in the store. Each product belongs to a specific section. It contains the following columns:

id: The primary key for each section, which uniquely identifies it.

name: The name of the section, represented as a String with a maximum length of 100 characters

.

3. Admin Table

The Admin table stores information about the store manager or admin. There is only one admin for the entire application. It contains the following columns:

id: The primary key for the admin, which uniquely identifies the admin.

email: The email address of the admin, represented as a String with a maximum length of 100 characters.

password: The password of the admin, represented as a String with a maximum length of 100 characters.

4. User Table

The User table represents registered users of the application. Users can create accounts and log in to the system. It contains the following columns:

id: The primary key for each user, which uniquely identifies the user.

email: The email address of the user, represented as a String with a maximum length of 100 characters.

password: The password of the user, represented as a String with a maximum length of 100 characters.

5. Purchase Table

The Purchase table records the purchase history of each user. It stores information about the products purchased by users. It contains the following columns:

id: The primary key for each purchase, which uniquely identifies it.

user_id: A foreign key that links the purchase to the corresponding user in the User table.

product_id: A foreign key that links the purchase to the corresponding product in the Product table.

quantity: The quantity of the product purchased by the user, represented as an Integer.

price: The total price of the purchased quantity of the product, represented as a Float.

section_id: A foreign key that links the purchase to the corresponding section in the Section table.

These tables are defined using SQLAlchemy's declarative syntax, where each table corresponds to a Python class. The relationships between tables are established using foreign key constraints, allowing efficient querying and retrieval of related data.

Features and Functionality

User Functionality

Home Page (Route: '/') : The home page provides an overview of the application.

User Registration (Route: '/user/new') : Users can create a new account by providing their email and password.

User Login (Route: '/user') : Registered users can log in using their email and password.

View Purchased Products (Route: '/user/purchased/<int:user_id>') : Users can view the list of products they have purchased.

View Available Products (Route: '/user/sections/<int:user_id>') : Users can view available products and sections.

Purchase Product (Route: '/user/purchase/<int:user_id>/<int:product_id>') : Users can purchase a specific product by specifying the quantity.

Admin Functionality

Admin Login (Route: '/admin') : The admin can log in using their email and password.

Manage Sections and Products (Route: '/admin/section') : The admin can view, add, edit, and delete sections and their respective products.

Add Section (Route: '/admin/add_section') : The admin can add a new section.

Edit Section (Route: '/admin/section/edit/<id>') : The admin can edit the name of an existing section.

Delete Section (Route: '/admin/section/delete/<id>') : The admin can delete a section after confirming the action.

Product Actions (Route: '/admin/actions/<id>') : The admin can perform actions (edit, delete) on specific products.

Add Product (Route: '/admin/section/product/<id>') : The admin can add a new product to a specific section.

Edit Product (Route: '/admin/product/edit/<id>') : The admin can edit the details of an existing product.

Delete Product (Route: '/admin/product/delete/<id>') : The admin can delete a product after confirming the action.

Working of Web App Video Link

<https://drive.google.com/file/d/1gIxMVefpsAddwy2xR4ulkrWWWiArUg0r/view?usp=sharing>

Conclusion

The Grocery Store Application developed for the "Modern Application Development 1" course at IIT Madras is a feature-rich platform that allows users to buy and sell groceries. It provides a user-friendly interface and essential functionalities for both users and the admin. The application is built using the Flask framework and SQLAlchemy for database management, making it efficient and easy to maintain.

The project aims to offer a practical learning experience in web application development, database management, and user authentication. It provides valuable insights into building multi-user applications and handling different user roles.

Thank you for reviewing this project report. If you have any questions or require further information, please feel free to reach out.

This report provides an overview of the Grocery Store Application and highlights important points. To view the actual code, please refer to the provided code snippet.