

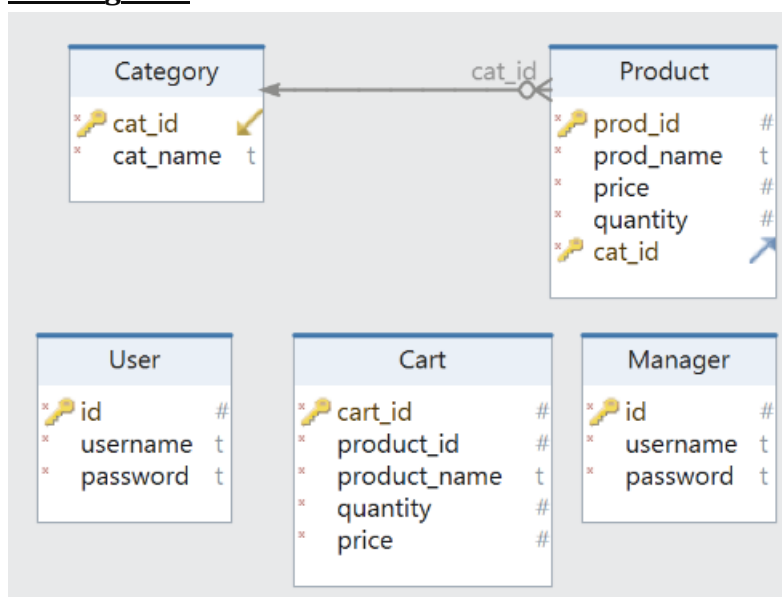
Description:

Grocery Shopping Application is a project under MAD-1 for the May-2023 Term of Online BS Degree. For this project we had to build a grocery shopping web application with Managers and Users. The manager has full control over the inventory can view the inventory and can also view a brief summary, whereas the user can add multiple products from multiple categories their cart and view the final bill.

Technologies used:

- Flask (for backend)
- Flask-Login (for user authentication)
- Flask-SQLAlchemy (for database establishment)
- SQLite3 (for database storage)
- Pyecharts (for inventory summary charts)
- Bcrypt (for password encryption)
- wtforms (for login input and validation)

ER Diagram:



- Every product belongs to a category.
- Every relation has an integer primary key 'id'.
- All attributes are 'NOT NULL'.
- 't' denotes varchar datatype.
- '#' denotes integer for all ids and float for price.

Features:

- Separate Login for Manager and User : On the Home page itself the one using the application has an option to identify themselves as Admin or User. The Database is initially empty and no user can login if a manager does not exist.
- Registration for both Manager and User : Since the database is initially empty, if a manager or a user tries to login without registering, a pop up message is displayed, alerting them that a registration is required in order to login and to reach respective dashboards.
- Authentication : When a manager or a user registers, their id, username and password is stored in their respective tables. This has been achieved using flaskforms and the information from the html page's input fields is sent as csrf tokens. Doing so improves the overall security of the login process. Certain validators are set for smooth functioning. The passwords are first salted and hashed using the Bcrypt model and this encrypted version of the password is stored in the database and further used for authentication when the manager or user tries to login. Upon successful login, they are redirected to their respective dashboards.

- Manager Dashboard : When the manager successfully logs in, he/she is redirected to manager_dashboard where 4 buttons are present, Create Category, Show Categories, View Products and Summary. Adequate routing and methods allow the manager to create categories, and when the Show Categories button is clicked, every created category can be seen along with 3 buttons per category, Add Product, Edit and Delete. Again the manager can add product(s) into each category and using flaskforms the information is stored in Category and Product tables respectively. The manager also has the ability to edit the category name. When the manager clicks on the View Products button, a web page is loaded with a table, which displays every product's name, the price per unit, the quantity in stock and the category details. Each product in this table has two more buttons, Edit and Delete which can help the admin to edit or delete products in or from the database. If the manager clicks on the Delete category button, then all the products within that category shall be deleted as well and the changes can be seen in the products webpage as well. Now coming to the Summary button, when that is clicked, we get two charts, one displaying the number of products in each category and the other one displaying the cost of the most expensive product from each category. This is achieved with the help of pyecharts module. The summary webpage also has a Go Back to Dashboard button for ease of navigation. This button is present in the categories and products webpage as well.

- User Dashboard : Upon successful login, the user lands on the user_dashboard webpage where the user can see all the available products from different categories. Each product has a counter with which the user can select the amount of a particular product that the user wishes to add to their cart. If the chosen amount exceeds the quantity in stock for that particular product, then a pop up message informs the user of the same. Otherwise, the Cart table is populated based on the user's actions on the dashboard. There is a search bar too with which the user can search for the desired products, when there is a match between existing product(s) and the user's search keyword, all other products are hidden and only the searched product is visible for the user to interact with. There is a View Cart button which when clicked, redirects the user to the view_cart webpage where a table is shown, displaying all the selected products by the user, their respective sub-total and a grand total at the end of the web page. There is a Go Back to Shopping button which enables the user to add more products to their cart.

Quick Demo video link:

https://drive.google.com/file/d/1UEdSlVMozTW0eSPdnuGmj9Uy7xFQ6mH_/view?usp=sharing