Author

Name: Yuvraj Sharma
Roll number: 21f3000180
Email: 21f3000180@ds.study.iitm.ac.in
About: I am a bachelor with half a decade of experience with web development, entrepreneurship, research and development and music.

Description

The application is a grocery shopping platform, which allows searching and purchasing of products from one or more categories. The users can sign up as a user or a manager. The users can purchase the products and view the invoices, the store managers can manage their products and request for new categories, the admin can create, update, delete and view the categories to be displayed to the users. The admin can also restrict / unrestrict a user's access on the app. Each user gets automated scheduled messages from the system regarding their account. The admin user gets the notification regarding any pending requests for new categories using GSpace, the users get a monthly report about their purchases in the given month, and the managers can download data about their products in a CSV format.

Technologies used

- ***Python*** (Used for creating the backend)
    - ***Flask*** (Used to create the web application)
    - ***Flask-SQLAlchemy*** (Used for interacting with the SQLite database)
    - ***Celery*** (Used for scheduling tasks)
    - ***Flask-Security*** (Used for authentication and RBAC)
    - …and related dependencies for the above packages to function
- ***Redis*** (Used for caching and storing async result)
- ***SQLite*** (Used for storing data)
- ***VueJS*** (Used for the front end)
- ***CSS3*** (Used for custom styling)
- ***Bootstrap*** (Used for basic styling)

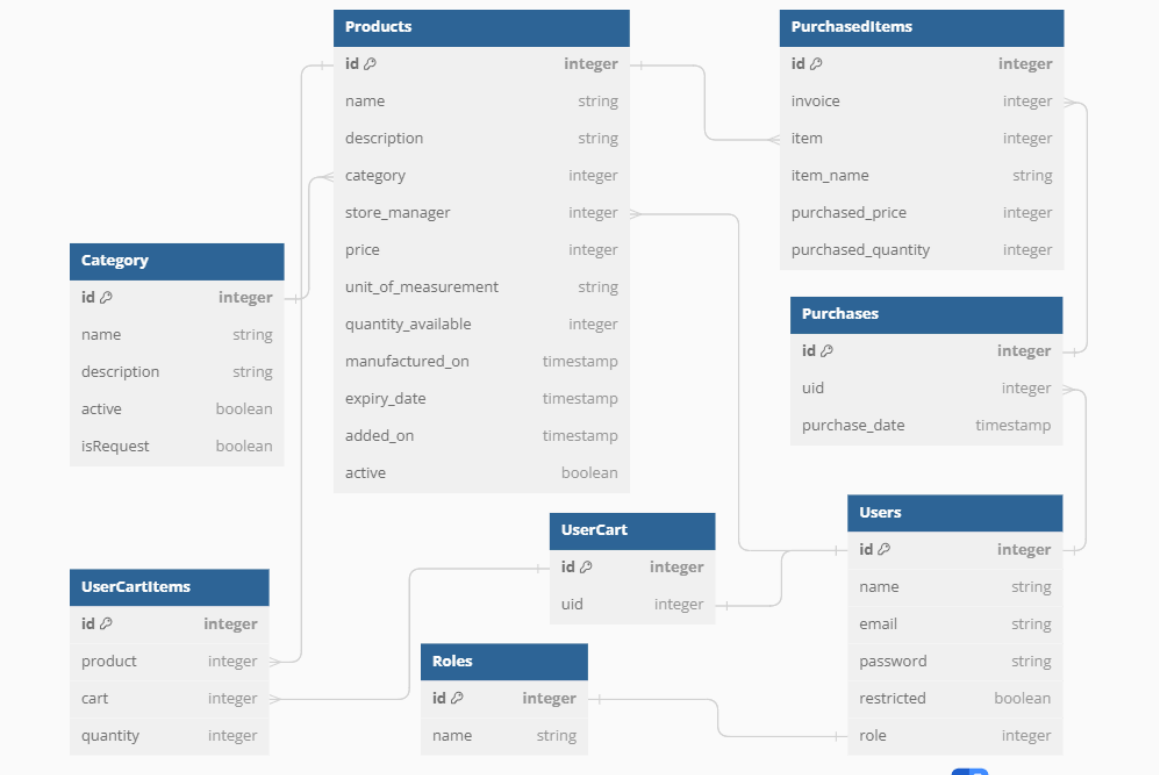
API Design

The API has been created for the following features:

- **Auth**: To perform authentication related functions.
- **Admin**: To perform CRUD on categories, approve requests, and manage users / managers by restricting / unrestricting their access to the app
- **Manager**: To perform CRUD operations for products, request categories and download data about their products
- **User**: To perform search across products and categories, and browse the marketplace
- **Cart**: To perform CRUD operations for cart items
- **Invoice**: To perform CRUD operations for purchased items / invoices

The API is secure, it allows restricted access using RBAC to perform these operations.

Database Schema Design



The database was designed to scale, taking a vague inspiration from various grocery apps. There are 2 main entities: category and product. These are related to each other via one-to-many relationships, as shown in the diagram above. To accommodate user retention and other business essential metrics, the user cart items and purchases are also stored in the database using necessary models and relationships

Architecture and Features

The project follows an iteration of the MVC model and follows the fundamental idea of separation of concerns. The models for each of the tables are defined in the `models` folder, the controllers which contain the functional methods with respect to the models are stored in the `controller` folder, the files which host the API endpoints and the usage of controllers with authentication validations are stored in the `views` folder. Additionally, there are some utility methods / classes for authentication, database connection, configuration and exceptions, which are stored in the `utils` folder. All these folders are collectively part of the `app` folder. The SQLite database is stored in the folder called `db`. The `static` and `templates` folder contain the static assets, like CSS files needed for styling the appearance, and the design templates respectively.
Further details have been explained in the README.md file in the root folder.

Features implemented in the application are:

- Proper login and signup page for users, managers and admin using RBAC
- Validation and authenticated access using token-based authentication provided by Flask Security package
- CRUD on products and categories
- Purchasing items across various categories and managing cart items
- Searching for products using various parameters like name and category
- Displaying all the products and their details to the user

Video

https://drive.google.com/file/d/1IZWxmrM8sd01VDAI908egwB4kszVnXPP/view?usp=sharing