Name = Manas Verma
Roll. No. = **21f1000467**
Email - 21f1000467@ds.study.iitm.ac.in
Video Link -

Description -
The FreshCart app, created with Flask, enables users to browse products, add to cart, view and remove items, and make purchases. Admins can manage products. It offers a user-friendly experience.

- **How To Use**
  - To access as ADMIN, the email is **'manasverma012345@gmail.com'** and password is **'1'**.
  - To access as USER, the email is **'123@manas'** and password is **'1'**.

- **DataBase Structure**

  1. "User" model: Represents user sign-up information, containing attributes id, email, password, first_name, and Place.
  2. "Categories" model: Represents product categories, containing attributes id and cat_name, as well as a relationship with the "Products" model.

3. "Products" model: Represents individual products, including attributes id, product_name, manufacture, expiry, p_per_u, stock, and c_id (foreign key referencing the "categories" table).

4. "Cart" model: Represents user's shopping cart, holding attributes cart_id, cart_item_id (foreign key referencing the "products" table), cart_user_id (foreign key referencing the "user" table), cart_cat_id (foreign key referencing the "categories" table), cart_quantity, and cart_amount. It establishes relationships with the "Products" and "Categories" models using foreign keys.

Technologies Used

The FreshCart web application is built using the following technologies:
- Flask: A micro web framework used for creating routes, handling requests, and managing the backend logic.
- Jinja2 Templating: Used to render dynamic content in HTML templates.
- Bootstrap: Provides CSS styling and responsive design for improved user experience.
- SQLite: A lightweight, embedded relational database management system used to store product information and user data.

Important Routes:

- Users: '/': First page with login options. '/user/login': User login. '/user/register': User registration. '/userhome/int:userid': User's home page with categories and products. '/add_to_cart/string:user_name': Add items to cart. '/see_cart/string:user_name': View cart items. '/make_purchase/string:username': Proceed to payment. '/payment/string:username': Handle payment.

- Admin: '/admin/login': Admin login. '/adminhome<bool1>': Admin's home page for inventory and categories. '/adminhomecategories': Display all categories. '/category/adding': Add categories. '/category/int:category_id/delete': Delete a category and associated products. '/products/int:category_id': Show products within a category. '/add_product/int:category_id': Add products to a category. '/product/int:product_id/edit': Edit product details. '/category/int:category_id/edit': Edit category details.

- Common: '/searching/string:user_name': Search for an item. '/addtocart/string:product_id/int:userid': Add items to cart. '/showcart/string:product_id/int:userid': Show cart items. '/delcart/int:cartid/int:userid': Delete item from cart. '/purchase/int:userid': Complete purchase and clear cart. '/searchcat/int:userid': Search for a category. '/search_categ/int:userid': Search based on category name. '/searchitem/int:userid': Search for items. '/search_item/int:userid': Search based on item details.

- Authentication: '/user_login': User login. '/admin_login': Admin login. '/user_signup': User sign-up. '/logout': Logout.

Final summary

Making this application using Flask has been a great experience, I have learned a lot in this process and this kind of application can be extremely useful for various kinds of grocery stores and also can be scaled to bigger ones.