Author: Pragyan Jyoti Dutta
Roll Number: 22f1000643
Email: 22f1000643@ds.study.iitm.ac.in

Description:

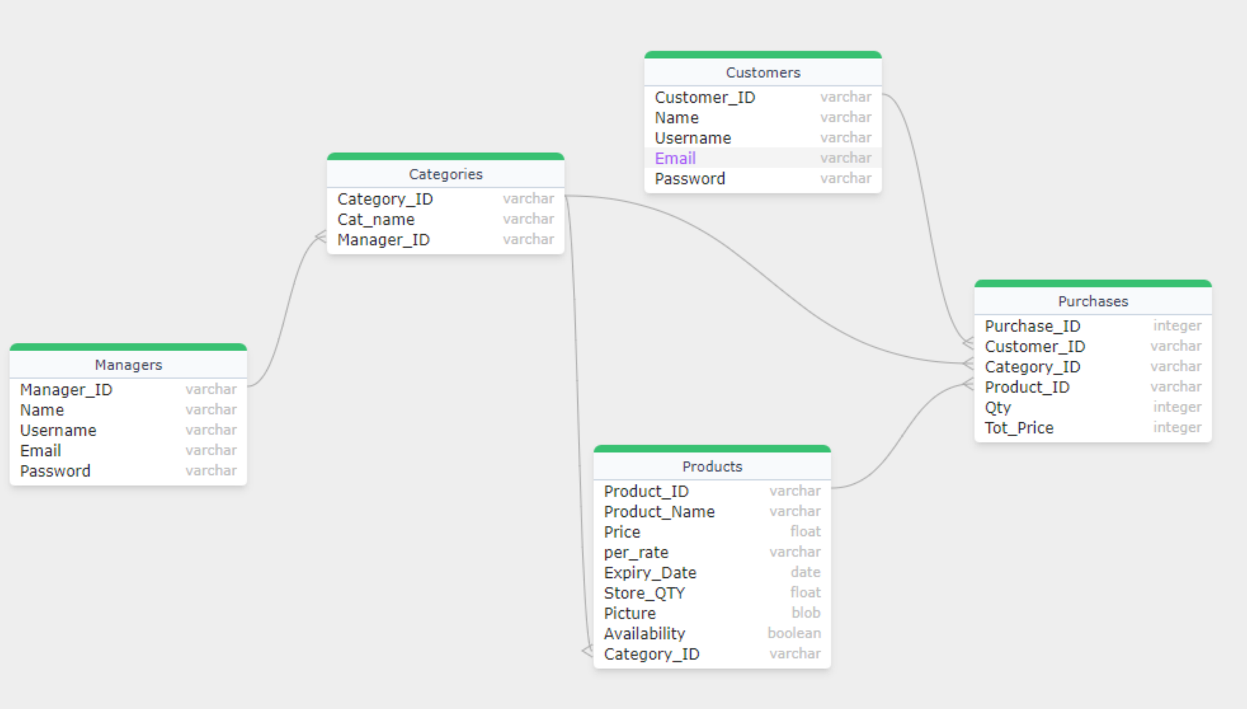
Grocify is a community-focused e-commerce grocery store designed to provide the freshest groceries directly to the doorsteps of local customers. The platform aims to bridge the gap between traditional grocery shopping and modern online conveniences, ensuring that customers receive quality products with the ease of a few clicks.

The innovative use of analytics by the managers enhances user experience, allowing for personalized shopping experiences and targeted promotions. By understanding customer behavior and preferences, Grocify can continually refine its offerings and service.

Technologies Used:

Flask: The backbone of the application code, handling routing and server-side logic.
Flask-Sqlalchemy: Extension used for ORM, making database interactions more intuitive.
Flask-login: Adds secure authentication, enabling sign-up and login functionalities.
Matplotlib: Utilized for generating interactive graphs, assisting managers in analytics and visualization.
Bootstrap: Responsible for styling, ensuring a responsive and visually appealing design.
Datetime: Essential for managing date and time datatypes, including product expiry dates.

DB Schema Design:



Customers	
Customer_ID	varchar
Name	varchar
Username	varchar
Email	varchar
Password	varchar

Categories	
Category_ID	varchar
Cat_name	varchar
Manager_ID	varchar

Managers	
Manager_ID	varchar
Name	varchar
Username	varchar
Email	varchar
Password	varchar

Purchases	
Purchase_ID	integer
Customer_ID	varchar
Category_ID	varchar
Product_ID	varchar
Qty	integer
Tot_Price	integer

Products	
Product_ID	varchar
Product_Name	varchar
Price	float
per_rate	varchar
Expiry_Date	date
Store_QTY	float
Picture	blob
Availability	boolean
Category_ID	varchar

Grocify's database consists of five interrelated tables:

Managers: Stores information about store managers, including unique Manager_ID, Name, Username, Email, and Password. Managers have the capability to add, edit, or delete categories and products.

Customers: Contains customer profiles with unique Customer_ID, Name, Username, Email, and Password.

Categories: Holds information on various product categories, with unique Category_ID and Cat_name. Manager_ID as a foreign key ensures that only managers can modify categories.

Products: Details the store's inventory with Product_ID, Product_Name, Price, per_rate, Expiry_Date, Store_QTY, Picture, Availability, and Category_ID.

Purchases: Tracks customer purchases, including Purchase_ID, Customer_ID, Category_ID, Product_ID, Quantity, and Total Price.

Architecture and Features:

Grocify's architecture is divided into well-organized folders and files:

Templates folder: Contains all HTML files for the view layer.
Static folder: Houses all CSS and image files.
models.py: Defines the database models.
Grocery_store.sqlite3: Stores all the tables.
app.py: Manages all the routes and functions to run the web application.

The application supports CRUD operations on both the manager/admin side and the customer side. Managers have control over categories and products, allowing them to add, edit, or delete items. Customers can seamlessly add items to their cart and manage their orders.

Furthermore, Grocify offers a personalized shopping experience by making use of analytics, potentially suggesting products based on previous purchases. The integration of visualizations offers managers insights into sales trends and customer behavior.

Video:

https://drive.google.com/file/d/1-ks0B8pGExJrOOPIoF5H3Ghe5eQ5rQTn/view?usp=drive_link