Report: MAD-1 Project

Author
Maanav A
21f1004840
21f1004840@ds.study.iitm.ac.in

Description:

The grocery store app project aims to create a web-based application that allows users to browse and purchase various grocery items online. It provides functionalities for user registration, login, searching for items, adding items to the cart, and placing orders. The app also includes an admin interface to manage categories and items, view insights, and perform CRUD operations on the database.

Technologies Used:

Flask: A Python web framework used for building the backend of the application.
SQLAlchemy: A Python ORM (Object-Relational Mapping) library for working with the database.
SQLite: A lightweight and serverless database management system used for storing data.
Jinja2: A templating engine for rendering dynamic content in HTML templates.
Bootstrap: A frontend framework for creating responsive and visually appealing user interfaces.
Chart.js: A JavaScript library for generating interactive charts and graphs.
Matplotlib: A Python library for plotting and visualizing data.

DB Schema Design:

The database schema comprises four main tables - User, Items, Categories, and Order. The User table stores user-related information such as user_id, username, email, mobile number, password, and isAdmin status. The Items table holds information about grocery items, including item_id, name, quantity, manufacture date, expiry date, price, and unit. The Categories table stores category_id and category name, while the Order table keeps track of order details, such as order_id, user_id, total price, and order date. The relationship between Categories and Items is maintained through the categoryOfItem table, using foreign keys to establish a many-to-many relationship between them.

API Design:

The APIs are designed to support CRUD (Create, Read, Update, Delete) operations for managing categories and items. Each API endpoint corresponds to a specific action, such as adding a new category, updating an item's details, deleting a category, etc. The API documentation is provided in YAML format.

Architecture and Features:

The project follows the Model-View-Controller (MVC) architecture, where the controllers handle user requests, the models represent the data structure and database interactions, and the views display the user interface. The controllers are organized in separate modules to handle user authentication, product search, order processing, and admin functions. The templates contain HTML files rendered using the Jinja2 templating engine, with Bootstrap for styling.

Default features include user registration and login, product search by name and category, adding items to the cart, and placing orders. Admin features include managing categories and items, viewing insights through interactive charts, and performing CRUD operations on the database.

Overall, the grocery store app provides a user-friendly and efficient way for users to shop for groceries online, while also offering admin tools for managing the store's inventory and gaining valuable insights into sales trends and performance. The technology stack, including Flask, SQLAlchemy, and Bootstrap, ensures a robust and responsive web application that caters to the needs of both users and administrators.

Video Link:
https://drive.google.com/file/d/1_IN0uBvdY-KT1dZCgBq0j2N19sleDBE9/view?usp=sharing