Amazon like e-commerce Web Application

Team :

Bagathi Shiv Kiran (Roll number: 200050019)

Deekonda Venkatesh Prasad (Roll number: 200050030)

Bale Teja Ramachandramurty (Roll number: 200050020)

Moganti Harshadeep (Roll number: 200050075)

Overview of application :

Our project aims to develop an E-commerce web application similar to Amazon, which will provide users with features such as product listings, search functionality, and user account management. Users will be able to create accounts and manage their personal information, track their orders, and leave product reviews. Similarly vendors can add products and view customer reviews. The application will have multiple user roles, including buyers, vendors.

Users :

The main classes of users of our application will be buyers, sellers, and administrators. Buyers will be able to browse products, add them to their cart, and make purchases. Sellers will be able to manage their inventory, add new products, and process orders.

Data to be stored :

The main entities that we will need to keep information about are products, users, and orders. For products, we will need to store information such as product name, description, price, and quantity in stock. This info will be updated when Buyers make purchases or when Sellers add to inventory. For users, we will need to store personal information such as name, address, and payment information. For orders, we will need to store information such as the items purchased, the total cost, and the shipping information.

Application Structure :

Backend :

s name, address, and payment information. For orders, we will need to store information such as the item

UI :

Based on React. We will create responsive UI and stylize using material UI.
Navbar: Shopping cart, Profile, Searchbox
Home Page: Where there are some random products/ based on previous purchases
Dashboard/profile Page: Admins and sellers will have access to a dashboard page where they can manage their products, orders, and other account details.
Product Views: Users will be able to view product details, including descriptions, images, prices, and reviews. with ui for order placement.

Other Aspects :

Test Data :

We plan on scraping few of amazon product pages and use a script to insert them into database.

Security :

We plan on using session management and cookies for security and faster login. We will look into prevention of XSRF attacks and ensure our website is free from them

DB Internals Proposal: Functional Dependencies in Postgres

Team :

Bagathi Shiv Kiran (Roll number: 200050019)

Deekonda Venkatesh Prasad (Roll number: 200050030)

Bale Teja Ramachandramurty (Roll number: 200050020)

Moganti Harshadeep (Roll number: 200050075)

Problem Specification :

The goal of this project is to add Functional Dependencies support to PostgreSQL. This feature will help remove redundancy in the database by normalizing it. We plan to provide the ability to create Functional Dependencies in the database schema and also a way to access them later. Additionally, we plan to implement error handling by checking for flags violated by these dependencies. This will ensure that the database maintains consistency and integrity over time, avoiding data inconsistencies and errors.

Solution Approach :

The approach we are planning is to create an index on (X,Y) for every functional dependency X -> Y, which stores the count of the number of tuples with that (X,Y) value and handles errors when there is a conflict of the co-domain. To implement this, we can modify the existing Gist indices to keep track of the count of tuples. To provide the feature of Functional Dependencies, we need to modify the query processing code to include syntax for specifying them.

Additionally, we need to modify the code for insertion, deletion, and creation methods in the access layer to keep track of counts and check for conflicts. To access Functional Dependencies as a query, we can modify the code to enable querying the index created for each dependency.

We are not entirely sure if this the correct implementation, but we plan on improving and learning through our experience as we explore more around this topic.

P.S. We are not entirely sure how well we could do database internals that is the reason we did not propose this earlier. We are still looking for resources and looking through official postgres implementation.