E – Commerce Platform

Description: This code implements a simple e-commerce system with the following data structures and functionalities.

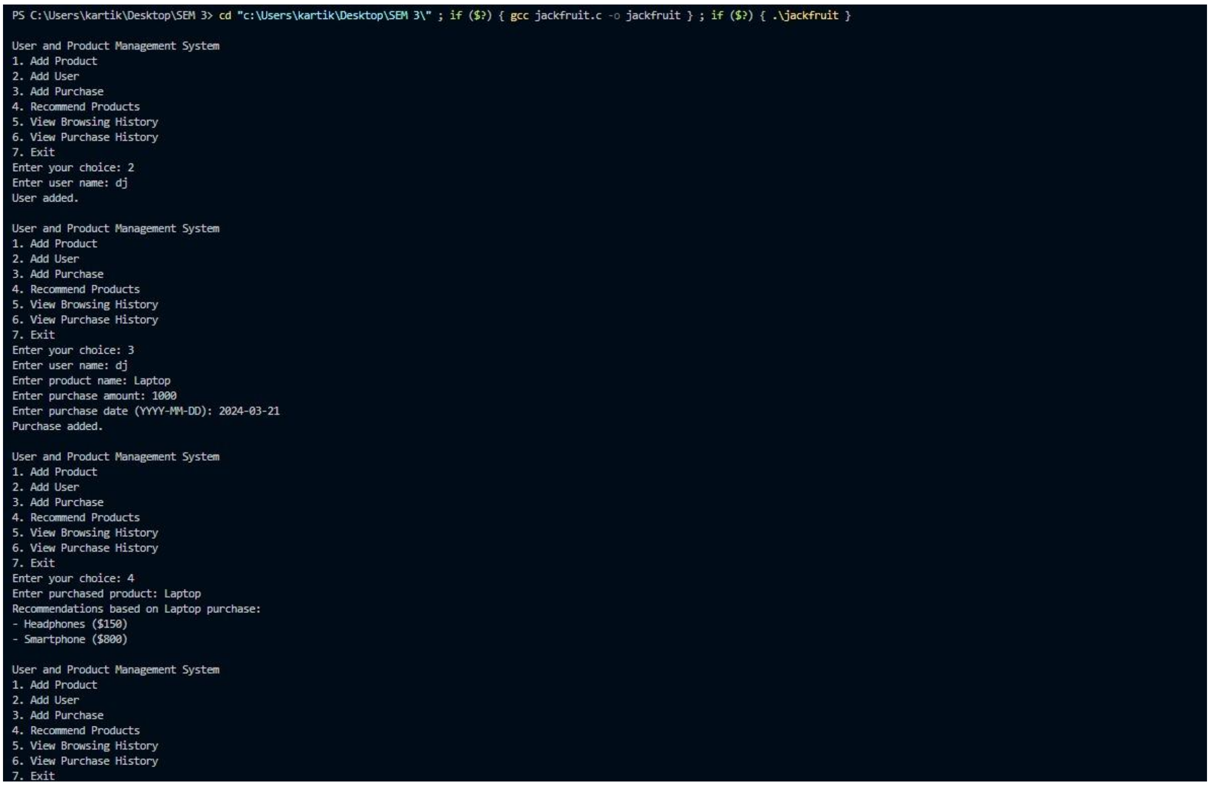
1. **Product Hash Table:**
    - The Product struct represents a product with a category, name, price, and a pointer to the next product in the same hash table bucket (next).
    - A hash table hash_table is used to store and retrieve product information, where the hash function is strlen(category) % HASH_TABLE_SIZE.
    - The add_product function adds a new product to the hash table.
2. **User Binary Search Tree:**
    - The UserNode struct represents a user with a name, a linked list of Purchase nodes, a linked list of Browse nodes, and pointers to the left and right child nodes (left and right).
    - A binary search tree root is used to store and retrieve user information, where the comparison is based on the user's name.
    - The create_user and add_user functions create and add a new user to the binary search tree.
    - The add_purchase function adds a new purchase to a user's purchase history, and the add_browse function adds a new browsing history entry to a user's browsing history.
    - The find_user function retrieves a user by their name.
3. **Purchase and Browsing History Linked Lists:**
    - The Purchase struct represents a purchase with a product, amount, date, and a pointer to the next purchase (next).
    - The Browse struct represents a browsing history entry with a category, date, and a pointer to the next browsing history entry (next).
    - These linked lists are used to store the purchase and browsing history for each user.

The justification for using these data structures is as follows:

1. **Hash Table for Products**: A hash table is used for the product data structure because it provides efficient constant-time (O(1)) access for lookup, insertion, and deletion operations. This is crucial for an e-commerce system that needs to handle a large number of products.
2. **Binary Search Tree for Users**: A binary search tree is used for the user data structure because it provides efficient (O(log n)) searching, insertion, and deletion operations, which are important for managing a potentially large number of users.

3. **Linked Lists for Purchase and Browsing History**: Linked lists are used to store the purchase and browsing history data because they allow for efficient insertion and traversal of the history data, which can be of varying length for each user.

Output Screenshots:



```
PS C:\Users\kartik\Desktop\SEM 3> cd "c:\Users\kartik\Desktop\SEM 3\" ; if ($?) { gcc jackfruit.c -o jackfruit } ; if ($?) { .\jackfruit }

User and Product Management System
1. Add Product
2. Add User
3. Add Purchase
4. Recommend Products
5. View Browsing History
6. View Purchase History
7. Exit
Enter your choice: 2
Enter user name: dj
User added.

User and Product Management System
1. Add Product
2. Add User
3. Add Purchase
4. Recommend Products
5. View Browsing History
6. View Purchase History
7. Exit
Enter your choice: 3
Enter user name: dj
Enter product name: Laptop
Enter purchase amount: 1000
Enter purchase date (YYYY-MM-DD): 2024-03-21
Purchase added.

User and Product Management System
1. Add Product
2. Add User
3. Add Purchase
4. Recommend Products
5. View Browsing History
6. View Purchase History
7. Exit
Enter your choice: 4
Enter purchased product: Laptop
Recommendations based on Laptop purchase:
- Headphones ($150)
- Smartphone ($800)

User and Product Management System
1. Add Product
2. Add User
3. Add Purchase
4. Recommend Products
5. View Browsing History
6. View Purchase History
7. Exit
```