DSA Hackathon

UE23CS252A

Team 7

1. Anupam G - PES2UG23CS904
2. Ayush Chakraborty - PES2UG23CS112
3. Archit Rode - PES2UG23CS088
4. Anuska Antara - PES2UG23CS084

The program files used:

- prod.c

This C program implements a hash table to store product information (ID and name). It uses separate chaining (linked lists) to handle collisions. The program allows inserting products, searching for a product by ID, and updating product names. Each product is stored at an index determined by hashing its ID. The hash table keeps track of products in linked lists, where each node represents a product with its ID, name, and count.

- userbase.c

This program implements a hash table to store user data (ID and name) with **linear probing for collision resolution**. Users can be added, searched, and updated by their ID. The hash table handles collisions by checking subsequent slots until an empty one is found. Each slot also has a flag to indicate if it is occupied. The program allows for inserting new users, searching for existing ones, and editing their names.

- graph_str.c

This program represents a graph using an adjacency list and supports Depth First Search (DFS) and Breadth First Search (BFS). It uses a queue for BFS and a recursive approach for DFS. The graph is constructed by adding edges between vertices, and traversal algorithms explore the graph either by going deep into branches (DFS) or level by level (BFS), marking nodes as visited to avoid revisiting.

- main.c

Building on the previous codes, this main.c ties together user and product management, allowing users to view, buy, and track products in a simple e-commerce system. It integrates hash tables for managing users and products, and uses graphs to track relationships between products—such as which products are viewed or purchased together. Users can be added, searched, or updated, and products can be added, found, or modified. Through graph traversal (BFS), the system visualizes the connections between products, helping users explore their purchase history or discover related products. The interactive menu guides users through managing accounts, browsing products, and tracking their shopping behavior, providing a dynamic and engaging experience.

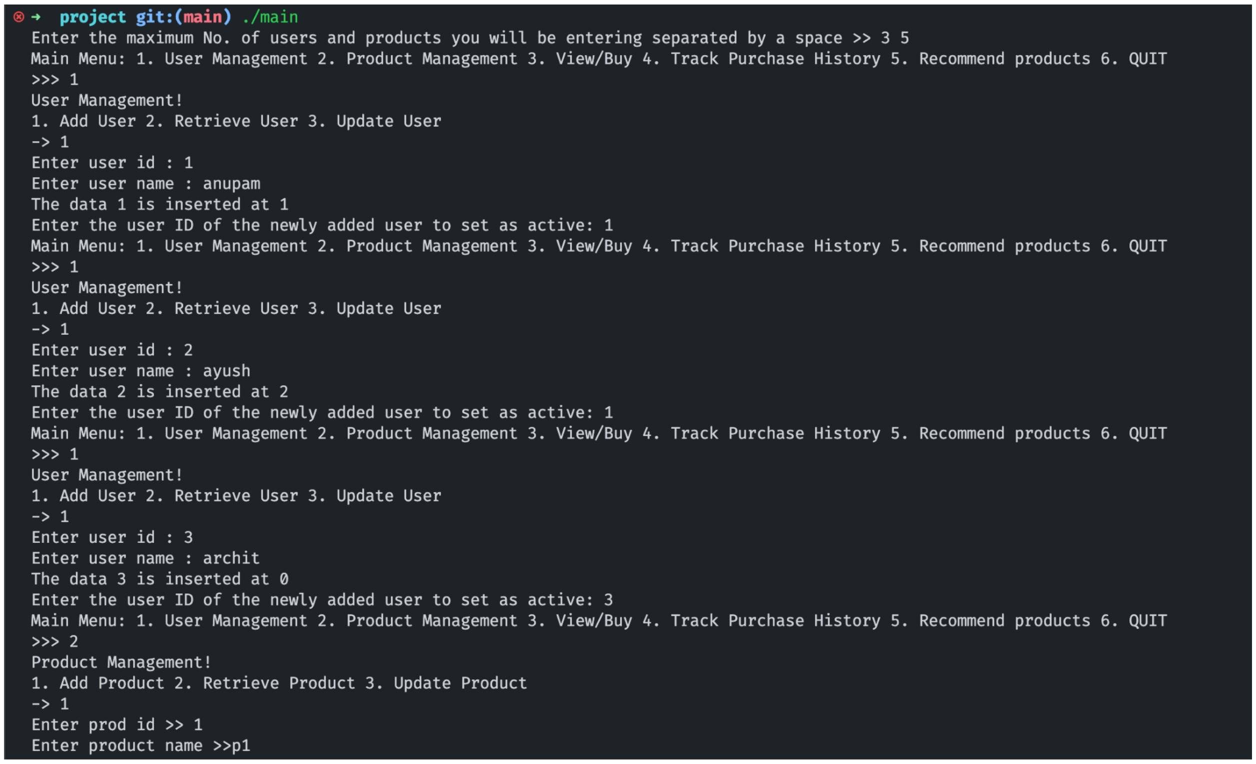
Data Structure Used :

The entire project was constructed mainly using two data structures ; hash tables and graphs.
There are two types of Hash tables used in this project. For storing the users we stored all user data in a hash table using linear probing, so that each field in the array is utilized, allowing us to store individual's purchases using nodes of linked list. Whereas for the product item storage, we preferred a hash table with separate chaining, as it did not require any separate data holding for users as such

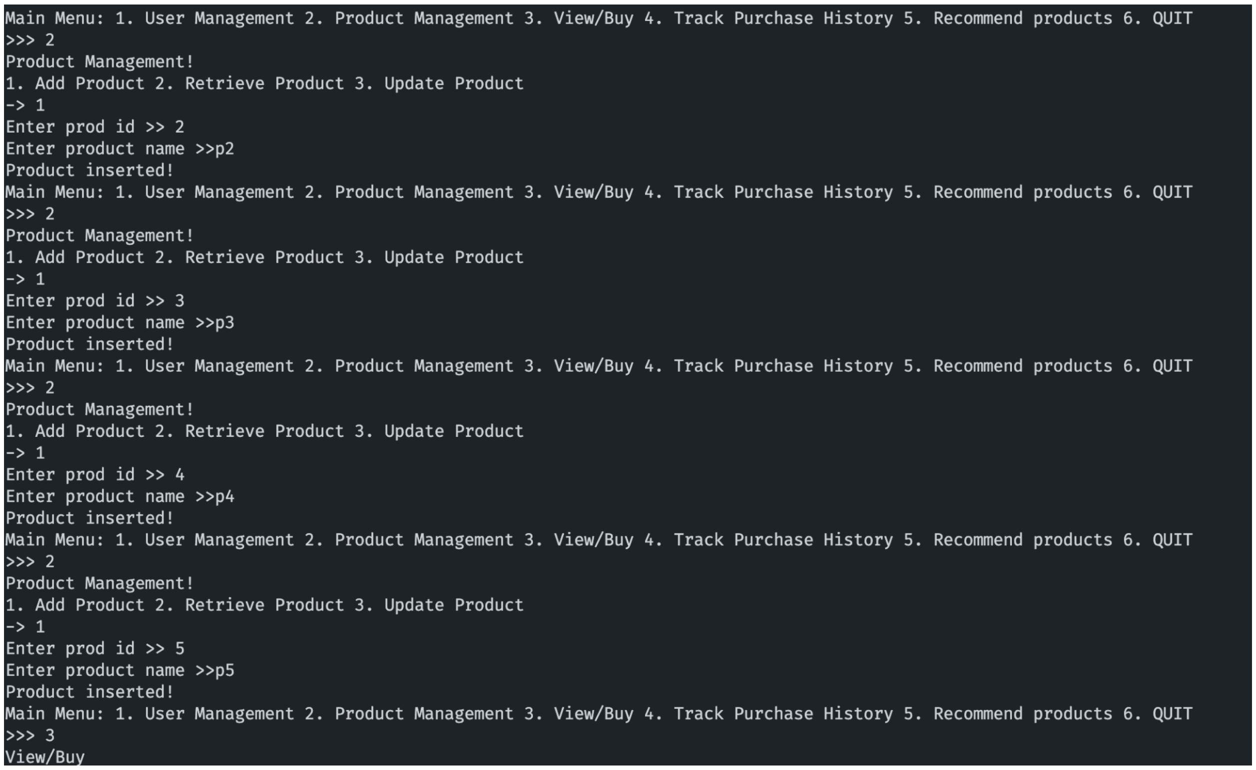
Once all these purchase and user details were stored, while viewing and purchasing, each instance was updated using graphs. This is done so because once all purchases, and browsing is over, we can create a graph of which products were viewed and which not for each individual user on demand. Graphs are the better choice for such a task.

Finally the recommendation is done based on the most bought item for each person. So if a person has bought product "A" multiple times, he will be recommended that product. In the future it is possible to predict the top 3 or 5 products.

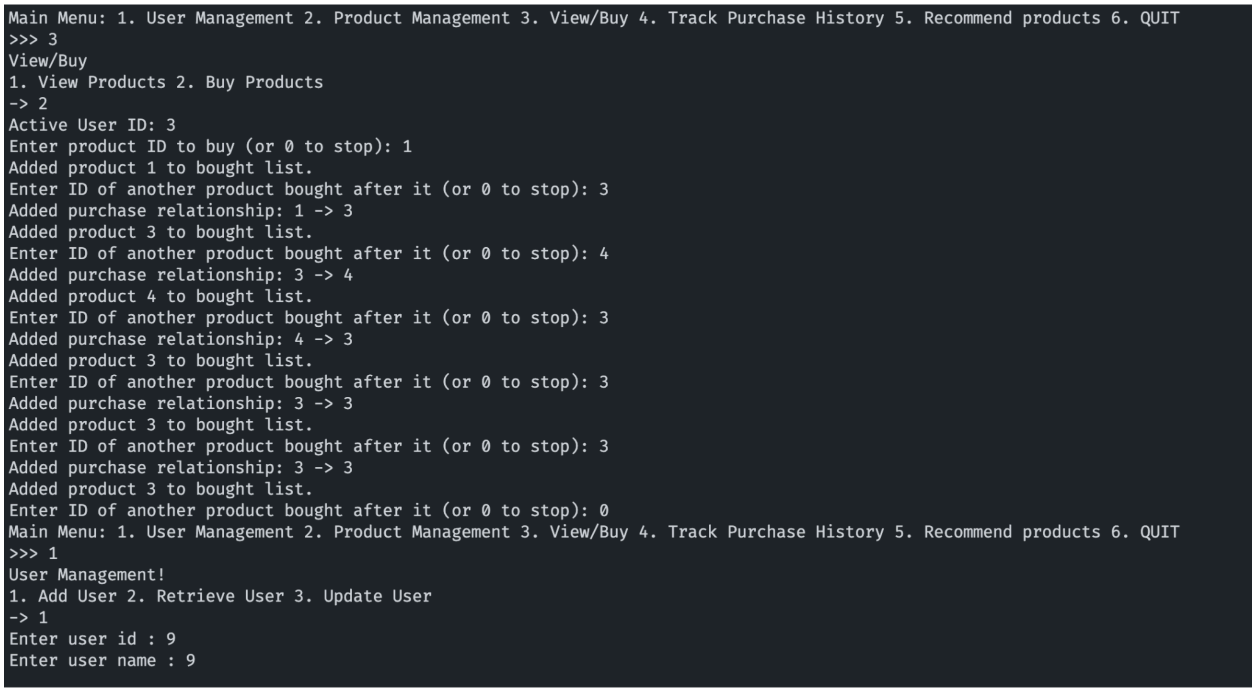
OUTPUT :



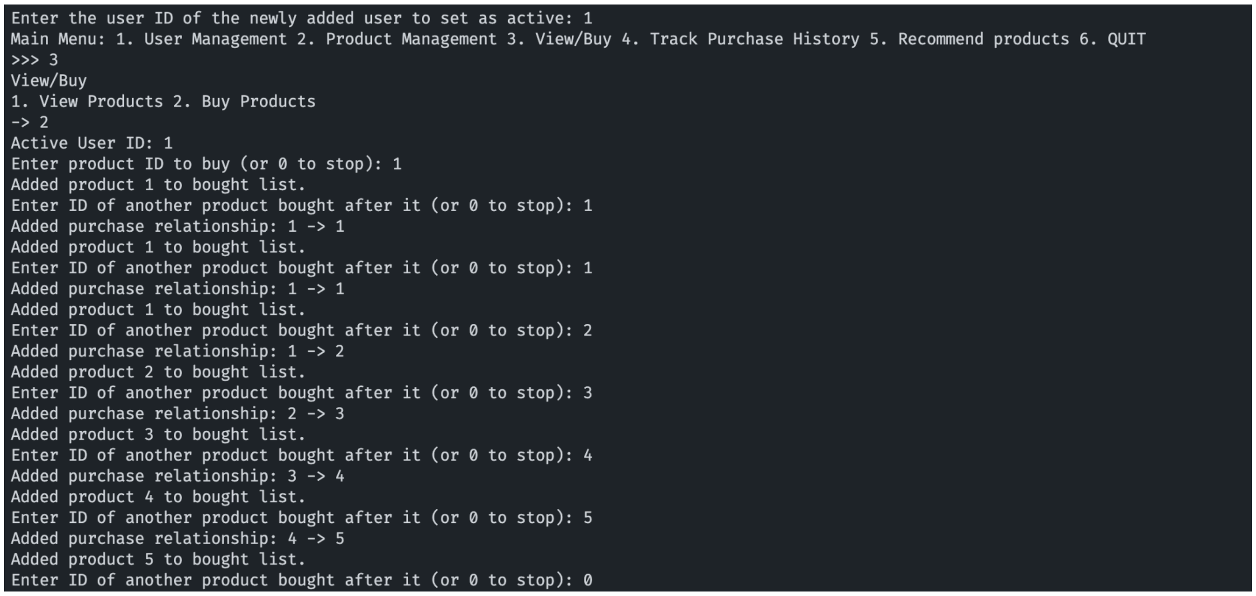
```
⊗ → project git:(main) ./main
  Enter the maximum No. of users and products you will be entering separated by a space >> 3 5
  Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
  >>> 1
  User Management!
  1. Add User 2. Retrieve User 3. Update User
  -> 1
  Enter user id : 1
  Enter user name : anupam
  The data 1 is inserted at 1
  Enter the user ID of the newly added user to set as active: 1
  Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
  >>> 1
  User Management!
  1. Add User 2. Retrieve User 3. Update User
  -> 1
  Enter user id : 2
  Enter user name : ayush
  The data 2 is inserted at 2
  Enter the user ID of the newly added user to set as active: 1
  Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
  >>> 1
  User Management!
  1. Add User 2. Retrieve User 3. Update User
  -> 1
  Enter user id : 3
  Enter user name : archit
  The data 3 is inserted at 0
  Enter the user ID of the newly added user to set as active: 3
  Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
  >>> 2
  Product Management!
  1. Add Product 2. Retrieve Product 3. Update Product
  -> 1
  Enter prod id >> 1
  Enter product name >>p1
```



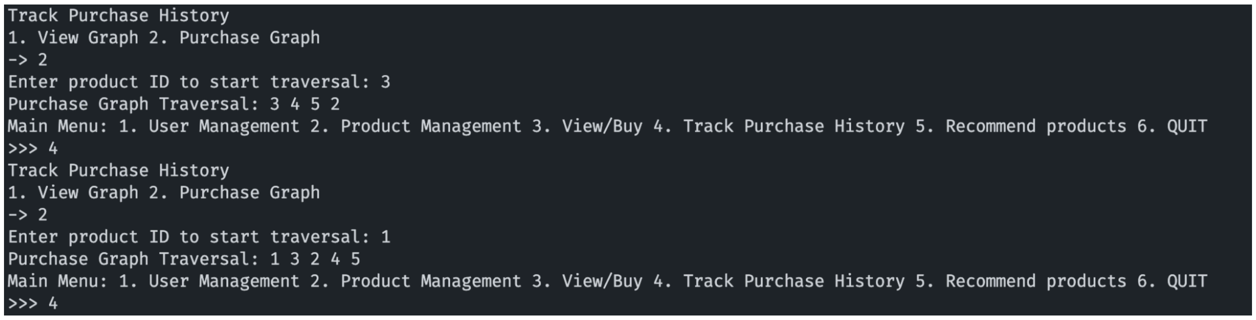
```
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 2
Product Management!
1. Add Product 2. Retrieve Product 3. Update Product
-> 1
Enter prod id >> 2
Enter product name >>p2
Product inserted!
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 2
Product Management!
1. Add Product 2. Retrieve Product 3. Update Product
-> 1
Enter prod id >> 3
Enter product name >>p3
Product inserted!
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 2
Product Management!
1. Add Product 2. Retrieve Product 3. Update Product
-> 1
Enter prod id >> 4
Enter product name >>p4
Product inserted!
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 2
Product Management!
1. Add Product 2. Retrieve Product 3. Update Product
-> 1
Enter prod id >> 5
Enter product name >>p5
Product inserted!
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 3
View/Buy
```



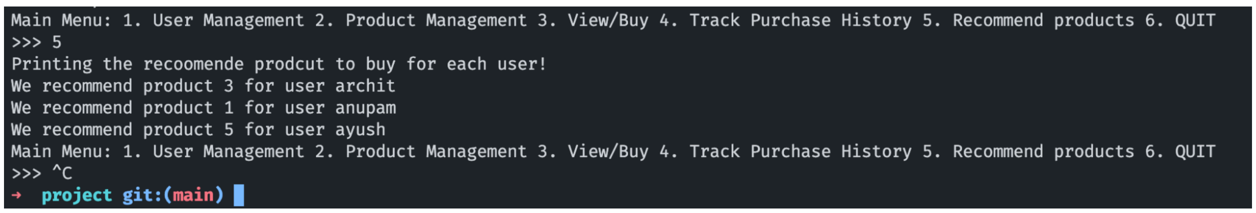
```
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 3
View/Buy
1. View Products 2. Buy Products
-> 2
Active User ID: 3
Enter product ID to buy (or 0 to stop): 1
Added product 1 to bought list.
Enter ID of another product bought after it (or 0 to stop): 3
Added purchase relationship: 1 -> 3
Added product 3 to bought list.
Enter ID of another product bought after it (or 0 to stop): 4
Added purchase relationship: 3 -> 4
Added product 4 to bought list.
Enter ID of another product bought after it (or 0 to stop): 3
Added purchase relationship: 4 -> 3
Added product 3 to bought list.
Enter ID of another product bought after it (or 0 to stop): 3
Added purchase relationship: 3 -> 3
Added product 3 to bought list.
Enter ID of another product bought after it (or 0 to stop): 3
Added purchase relationship: 3 -> 3
Added product 3 to bought list.
Enter ID of another product bought after it (or 0 to stop): 0
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 1
User Management!
1. Add User 2. Retrieve User 3. Update User
-> 1
Enter user id : 9
Enter user name : 9
```



```
Enter the user ID of the newly added user to set as active: 1
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 3
View/Buy
1. View Products 2. Buy Products
-> 2
Active User ID: 1
Enter product ID to buy (or 0 to stop): 1
Added product 1 to bought list.
Enter ID of another product bought after it (or 0 to stop): 1
Added purchase relationship: 1 -> 1
Added product 1 to bought list.
Enter ID of another product bought after it (or 0 to stop): 1
Added purchase relationship: 1 -> 1
Added product 1 to bought list.
Enter ID of another product bought after it (or 0 to stop): 2
Added purchase relationship: 1 -> 2
Added product 2 to bought list.
Enter ID of another product bought after it (or 0 to stop): 3
Added purchase relationship: 2 -> 3
Added product 3 to bought list.
Enter ID of another product bought after it (or 0 to stop): 4
Added purchase relationship: 3 -> 4
Added product 4 to bought list.
Enter ID of another product bought after it (or 0 to stop): 5
Added purchase relationship: 4 -> 5
Added product 5 to bought list.
Enter ID of another product bought after it (or 0 to stop): 0
```



```
Track Purchase History
1. View Graph 2. Purchase Graph
-> 2
Enter product ID to start traversal: 3
Purchase Graph Traversal: 3 4 5 2
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 4
Track Purchase History
1. View Graph 2. Purchase Graph
-> 2
Enter product ID to start traversal: 1
Purchase Graph Traversal: 1 3 2 4 5
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 4
```



```
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> 5
Printing the recoomende prodcut to buy for each user!
We recommend product 3 for user archit
We recommend product 1 for user anupam
We recommend product 5 for user ayush
Main Menu: 1. User Management 2. Product Management 3. View/Buy 4. Track Purchase History 5. Recommend products 6. QUIT
>>> ^C
➜  project git:(main) 
```