

Dawki user Interface guide

Part 1: THE CLI

```
Welcome to Dawki Store!  
  
1. Enter as Admin  
2. Enter as Customer  
3. EXIT
```

Our Welcome/ Login page

This is the homepage of our Command line interface , here the user can choose to login as an Admin , customer or exit.

```
Enter your choice: 1  
Enter the admin id: 1  
Enter the username: admin  
Enter the password: admin  
Login successful!
```

Our admin login page

This is the login page for admins , here admins can login using their id , username and password.

```
Enter the customer id: 1
Enter the username: steve
Enter the password: strive714
```

Our customer login page

This is the login page for customers, here customers can login using their id , username and password.

```
1. Create a new customer account
2. Enter a new product
3. Display all customers
4. Display all products
5. Test the triggers
6. Display all orders made through cash on delivery
7. Show the number of customers having a particular membership status
8. Print all the customer details along with delivery agent assigned for their order
9. Run OLAP queries
10. Change the price of a product
11. Change the order assigned to a delivery agent
12. Execute trigger 1
13. Execute trigger 2
14. EXIT
```

Our Admin page

Admins are directed to this main page , where they get various options which they can use to use the database.

1. Change delivery address
2. Change mobile number
3. Change password
4. Change email address
5. Change username
6. Change membership status
7. Display my cart
8. Add product to my cart
9. Add money to my wallet
10. Show the delivery agent details
11. Checkout
12. Remove product from my cart
13. View order history
14. Check wallet balance
15. Search a product
16. Empty cart
17. EXIT

Our customer page

Customers are directed to this main page , where they get various options which they can use to use the database.

PART 2: FUNCTIONS

Our CLI is written in Python and contains various functions related to managing customers and products in a database.

1.The **create_customer_account** function creates a new customer account by taking input from the user for various attributes such as customer_id, username, delivery_address, password, email, and mobile_number, and then inserts the values into the customers table in the database.

2.The **create_product** function creates a new product by taking input from the user for various attributes such as product_id, product_name, product_price, product_stock, and product_rating, and then inserts the values into the product table in the database.

3.The **display_customers** function retrieves all the records from the customers table and displays them in a formatted way.

4.The **display_products** function retrieves all the records from the product table and displays them in a formatted way.

5.The **change_address** function allows the user to change their delivery address by taking input for the new address and updating the delivery_address field in the customers table for the specified customer_id.

6.The **change_mobile** function allows the user to change their mobile number by taking input for the new mobile number and updating the mobile_number field in the customers table for the specified customer_id.

7.The **change_password** function allows the user to change their password by first verifying their old password and then updating the pass field in the customers table for the specified customer_id.

8.The **display_cart** function retrieves all the records from the cart table for the specified customer_id and displays them in a formatted way.

9.The **add_money** function allows the user to add money to their wallet by taking input for the amount and updating the balance field in the wallet table for the specified customer_id.

10.The **show_delivery_agent_details** function retrieves all the records from the delivery_agent table for the specified customer_id and displays them in a formatted way.

11.The **change_email** function allows the user to change their email by taking input for the new email and updating the email_id field in the customers table for the specified customer_id.

12.The **list_products_by_category** function takes user input for a product category and returns a list of all products in that category from a database. It uses a SQL query to retrieve the products and prints them in a readable format.

13.The **change_rating** function takes user input for a new rating and a product ID and updates the product's rating in the database. It uses an SQL query to update the rating.

14.The **change_username** function takes user input for a new username and a customer ID and updates the customer's username in the database. It uses an SQL query to update the username.

15.The **change_membership_status** function takes user input for a new membership status and a customer ID and updates the customer's membership status in the database. It uses an SQL query to update the membership status.

16.The **checkout** function updates the customer's wallet balance based on the transaction amount and empties the customer's cart if they agree to proceed with the checkout process. If the wallet balance is insufficient, the function prints a

message and returns from the function. If the customer cancels the checkout, the function prints a message stating that the transaction was cancelled.

17.The **execute_trigger1** function creates a trigger in the database that updates the coupon value in the cart table based on changes in a customer's membership status. It uses an SQL query to create the trigger.

18.The **execute_trigger2** function creates a trigger in the database that records price changes of products in the price_change table. It uses an SQL query to create the trigger.

19.The **add_product_to_cart** Prompts the user to enter a product ID and quantity, checks if the requested quantity of the product is available in stock, calculates the total price of the requested quantity, determines the discount coupon amount based on the total price, and inserts the product details along with the calculated values into a 'cart' table in the database. If the requested quantity is not available in stock, the function returns a message indicating that the product cannot be added to the cart.

20.The **remove_product_from_cart** Prompts the user to enter a product ID, removes the product from the customer's cart in the database.

21.The **view_order_history** Retrieves the order history for a given customer from the database and displays it to the user.

22.The **Check_balance** Retrieves the balance from the customer's wallet in the database and displays it to the user.

23.The **change_order** Prompts the user to enter an agent ID and an order ID, and updates the delivery_agent table in the database with the new order ID for the given agent ID.

24.The **search_product** Prompts the user to enter the name of a product, searches the product table in the database for products that match the search query, and displays the results to the user.

25.The **change_price** function allows the admin to change the price of a product by taking input for the product_id and the new price and updating the product_price field in the product table.