

Rapid Ration - User and Admin Management System

1. Introduction The code represents a user and admin management system for a ration distribution application called "Rapid Ration". It provides a graphical user interface (GUI) using the Tkinter library in Python and interacts with a MySQL database to store and retrieve data.
2. Dependencies
 - Python 3.x
 - Tkinter library
 - PyMySQL library
3. Database Configuration The code establishes a connection to a MySQL database using the following configuration:
 - Host: 'localhost'
 - User: 'root'
 - Password: 'abcd@1234'
 - Database: 'aditya' Make sure to have a MySQL server running and update the configuration according to your setup.
4. User Interface The main window of the application allows the user to select their user type: Admin or User. Clicking on the respective button opens a login window for authentication.
5. Login Window
 - The login window prompts the user to enter their username and password.
 - For admin login, the user needs to enter their admin ID and password.
 - For user login, the user needs to enter their username and ration card number.
 - Clicking the "Login" button initiates the authentication process.

6. Admin Authentication

- The `authenticate_admin` function handles the admin authentication process.
- It retrieves the admin credentials from the "admin" table in the database based on the provided admin ID and password.
- If the credentials are valid, it opens the admin window.
- If the credentials are invalid, an error message is displayed.

7. Admin Window

- The admin window provides options for the authenticated admin user.
- It includes buttons to view suppliers and view users.

8. View Suppliers

- The `view_supplier_id` function retrieves the supplier IDs, first names, and addresses from the "suppliers" table in the database.
- It displays the supplier information in a message box.

9. View Users

- The `view_user_id` function retrieves the user IDs, names, and order history from the "users" and "orders" tables in the database.
- It displays the user information and their order count in a message box.

10. User Authentication

- The `authenticate_user` function handles the user authentication process.
- It retrieves the user credentials from the "users" table in the database based on the provided username and ration card number.
- If the credentials are valid, it opens the user options window and displays a welcome message.
- If the credentials are invalid, an error message is displayed.

11. User Options Window

- The user options window provides options for the authenticated user.
- It includes buttons to check the cart, view order history, view the product directory, and order items.

12. View Product Directory

- The `view_product_directory` function retrieves the available products from the "products" table in the database.
- It displays the product information, including ID, name, description, price, stock quantity, and availability, in a message box.

13. Order Item

- The `order_item` function allows the user to order an item by entering the product ID and quantity.
- It retrieves the product details from the database based on the provided product ID.
- If the stock quantity is sufficient, it adds the item to the user's cart and updates the stock quantity in the database.
- If the stock quantity is insufficient or the product ID is invalid, an appropriate warning message is displayed.

14. Place Order from Cart

- The `place_order_from_cart` function allows the user to place an order from their cart.
- It retrieves the cart items of the logged-in user from the "cart" table in the database.
- It inserts the order details into the "orders" table, updates the stock quantity in the "products" table, and clears the user's cart.
- The order placement is performed within a transaction to ensure data consistency.

15. Delete from Cart

- The `delete_from_cart` function allows the user to remove an item from their cart.
- It deletes the specified product from the user's cart in the database.

16. View Cart Items

- The `view_cart_items` function displays the cart items of the logged-in user.
- It retrieves the cart items from the "cart" and "products" tables in the database.
- It displays the product name, quantity, and price of each item in a treeview widget.
- The user can place an order from the cart or remove items from the cart.

17. View Order History

- The `view_order_history` function displays the order history of the logged-in user.
- It retrieves the order details from the "orders" and "products" tables in the database.
- It displays the order ID, product name, price, and order date in a treeview widget.

18. Error Handling

- The code includes error handling using try-except blocks to catch and handle exceptions.
- If an exception occurs during database operations or user interactions, appropriate error messages are displayed using message boxes.

19. Database Connection

- The code establishes a connection to the MySQL database using the PyMySQL library.
- The connection is closed when the program ends.
- Main Event LoopThe main event loop is started using `root.mainloop()` to handle user interactions and keep the GUI running.