Report: Inventory Management System

Purpose: The purpose of the code is to create a simple inventory management system with user authentication, product management, and sales tracking. It allows users to add, view, update, delete, and search for products in an inventory. Additionally, it tracks sales, updates stock levels, and supports restocking products that are low in quantity. The system ensures secure user authentication and stores data persistently using text files.

Importance: This inventory management system is important for businesses or small enterprises to efficiently manage product inventories. It helps track stock levels, monitor sales, and ensure products are always available. The code provides an organized way to handle inventory data and automate tasks like updating stock, restocking, and recording sales, thus improving business operations and decision-making. The use of file handling ensures that data is saved and can be accessed even after program termination, making the system more reliable and practical.

Features:-

1. User Authentication:

- Sign Up: Allows new users to register by providing a username and password.
- Sign In: Validates existing users and ensures secure access to inventory data.

2. Product Management:

- Add Products: Enables adding new products to the inventory.
- Update Products: Allows modification of product details such as name, price, and quantity.
- Delete Products: Removes products from the inventory.

3. Inventory Operations:

- Search Products: Retrieves product details by ID.
- Sort Products: Organizes inventory based on price, quantity, or name.
- Sold Products: Tracks and updates the quantity of sold products.

4. Data Persistence:

- Inventory data is stored in text files specific to each user.
- Ensures data is saved across program sessions.

5. Interactive Console UI:

- Colored text output for better user experience.
- Clear menu-driven navigation.

6. File Handling:

- Ensuring data integrity during simultaneous read and write operations.
- Handling cases where files do not exist or are inaccessible.

7. Error Handling:

- ❖ Validating user input to prevent crashes caused by invalid data.
- ❖ Managing edge cases, such as attempting to delete a product that does not exist.

8. Sorting Algorithms:

- ❖ Implementing sorting based on multiple criteria while maintaining code simplicity.

9. User Authentication:

- ❖ Protecting user credentials and ensuring secure login processes.

Function Descriptions

Function Descriptions

1. fileExists()

- Checks if a specified file exists.
- **Prototype:** `int fileExists();`
- **Parameters:** `filename`: File name to check
- **Return Value:** 1 if file exists, 0 otherwise

2. createFile()

- Creates a new file with the specified name.
- **Prototype:** `void createFile(char *name);`
- **Parameters:** `name`: File name to create
- **Return Value:** None

3. sign_in()

- Handles user sign-in by verifying credentials and preparing the user's inventory file.
- **Prototype:** `int sign_in(struct Users user[], int *user_count);`
- **Parameters:** `user[]`: Array of users, `user_count`: Pointer to user count
- **Return Value:** 1 if sign-in successful, 0 otherwise

4. sign_up()

- Allows a new user to register and sign in.
- **Prototype:** `void sign_up(struct Users user[], int *user_count);`
- **Parameters:** `user[]`: Array of users, `user_count`: Pointer to user count
- **Return Value:** None

5. write_product()

- Adds a new product to the inventory, ensuring unique IDs.
- **Prototype:** `void write_product();`
- **Parameters:** None
- **Return Value:** None

6. read_products()

- Reads and displays products from the user's inventory file.
- **Prototype:** `void read_products();`
- **Parameters:** None
- **Return Value:** None

7. del_product()

- Deletes a product from the inventory based on its ID.
- **Prototype:** `void del_product();`
- **Parameters:** None
- **Return Value:** None

8. update_product()

- Updates product details based on its ID.
- **Prototype:** `void update_product();`
- **Parameters:** None
- **Return Value:** None

9. search_product()

- Searches for a product by its ID and displays details.
- **Prototype:** `void search_product();`
- **Parameters:** None
- **Return Value:** None

10. sort_products()

- Sorts products based on price, quantity, or name.
- **Prototype:** `void sort_products();`
- **Parameters:** None
- **Return Value:** None

11. sold_products()

- Records product sales and updates stock levels.
- **Prototype:** `void sold_products();`

- **Parameters:** None
- **Return Value:** None

12. restock_products()

- Identifies and displays products that need restocking.
- **Prototype:** `void restock_products();`
- **Parameters:** None
- **Return Value:** None

13. main()

- The main function that handles user interaction and drives the application flow.

Data Structures

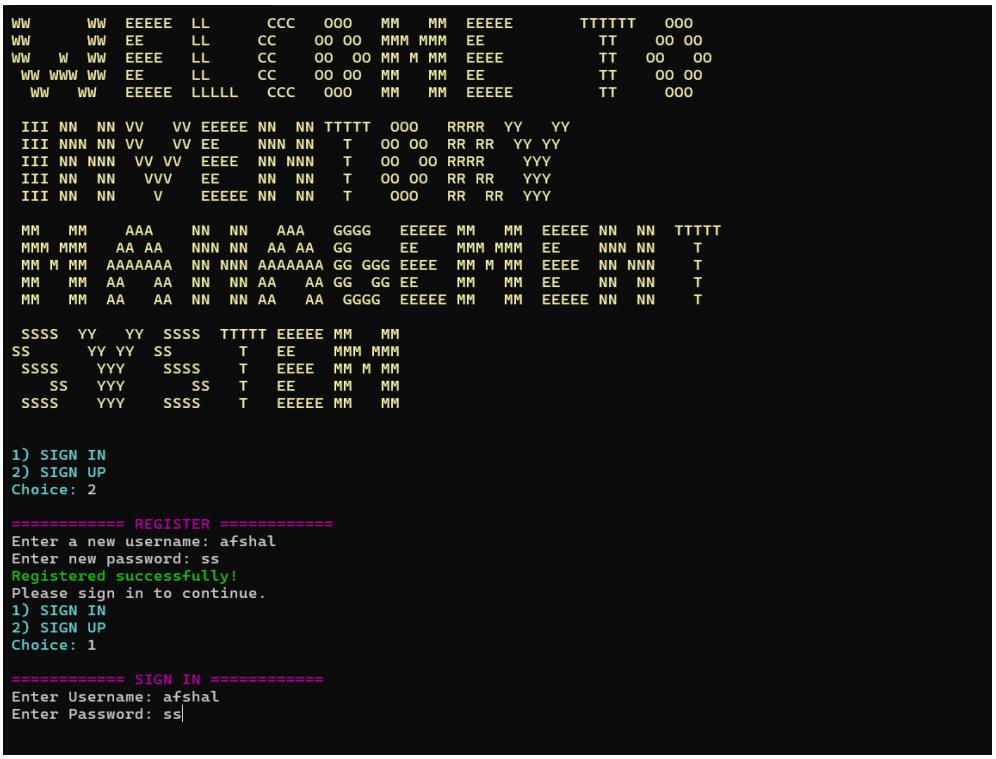
- **Product Structure:**
  - `id`: Unique product identifier
  - `name`: Product name
  - `price`: Product price
  - `quantity`: Current stock quantity
  - `items_sold`: Total items sold
- **User Structure:**
  - `username`: User's username
  - `password`: User's password

Global Variables

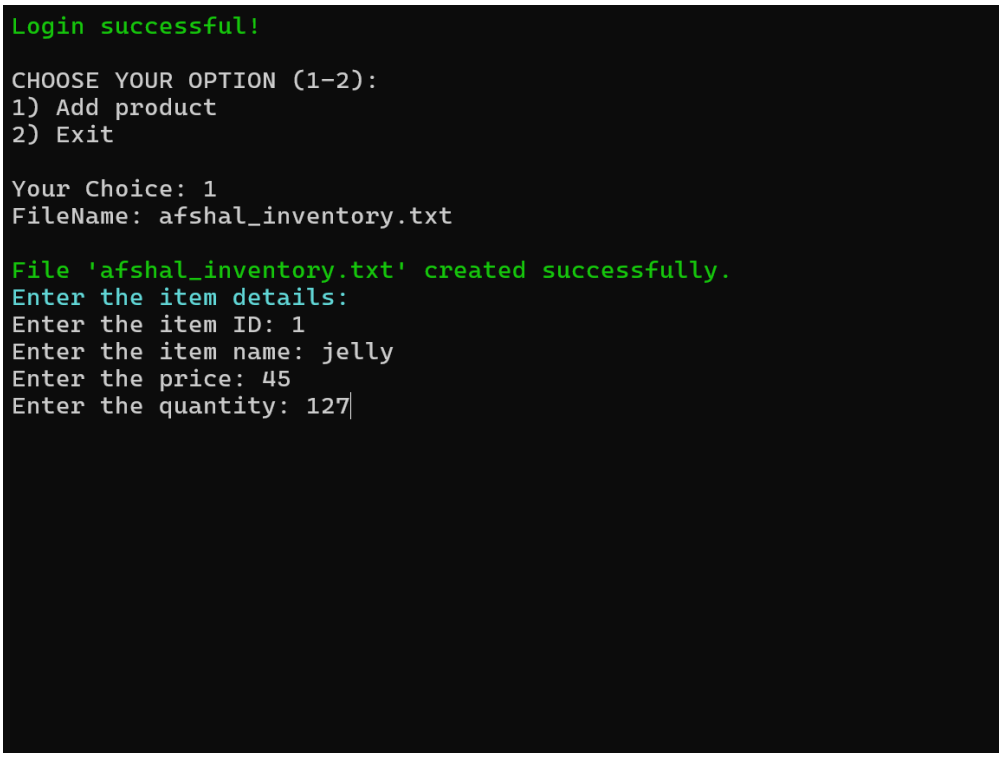
- `fileName`: Stores the unique inventory file name for the logged-in user
- `products`: Array to store product information
- `product_count`: Tracks the number of products in the inventory

Sample Test Case:-

1:- signing up and signing in.



2:- Creating file upon adding a product:-



3:- Choosing the option of add product:-

```
CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 1
```

4:- Adding another item and choosing display option.

```
Enter the item details:
Enter the item ID: 2
Enter the item name: biscuit
Enter the price: 34
Enter the quantity: 98
Product added successfully.

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 2
```

5:- Product is displayed and choosing delete option.

```
ID      Product Name                    Price   Quantity        Items Sold
-------------------------------------------------------------------------------
1       jelly                           45.00   127             0
2       biscuit                         34.00   98              0

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 3
```

6:- Product has been deleted and now choosing update option.

```
Enter the ID of the product to delete: 2
Product with ID 2 has been deleted.

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 4
```

7:- Updated item with respect to price and choosing display option again.

```
Enter the item ID to update: 1

Product found:
ID: 1
Name: jelly
Price: 45.00
Quantity: 127
Items Sold: 0
What do you want to update?
1) Name
2) Price
3) Quantity
Enter your choice: 2
Enter new price: 67
Product updated successfully!

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 2
```

8:- products displayed and choosing search option.

```
ID      Product Name                    Price   Quantity        Items Sold
-------------------------------------------------------------------------------
1       jelly                           67.00   127             0
2       chocolate                       80.00   167             0
6       candy                           98.00   67              0

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 5
```

9:- searching ID in inventory and proceeding towards product sold function.

```
Enter the product ID to search: 2

Product found:
ID: 2
Name: chocolate
Price: 80.00
Quantity: 167
Items Sold: 0

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 6
```

10:- Entering id and amount of item sold and choosing display option again.

```
Enter the product ID which was sold: 1
Enter the quantity sold: 23
Sale recorded: 23 items sold.
Product data updated successfully!

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 2
```

11:- Products are displayed and now choosing sorting option.

```
ID      Product Name                    Price   Quantity        Items Sold
-------------------------------------------------------------------------------
1       jelly                           67.00   104             23
2       chocolate                       80.00   167             0
6       candy                           98.00   67              0

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 7
```

12:- Sorting completed and displaying products again.

```
Choose sorting criterion:
1) Price
2) Quantity
3) Product Name
Enter your choice: 3
Sorting completed.

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 2
```

13:- Products displayed and choosing restock option (restock set to quantity<10)

```
ID      Product Name                    Price   Quantity        Items Sold
-------------------------------------------------------------------------------
6       candy                           98.00   67              0
2       chocolate                       80.00   167             0
1       jelly                           67.00   7               120

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 8
```

14:- Products needed to Restock are displayed and now choosing the exit option.

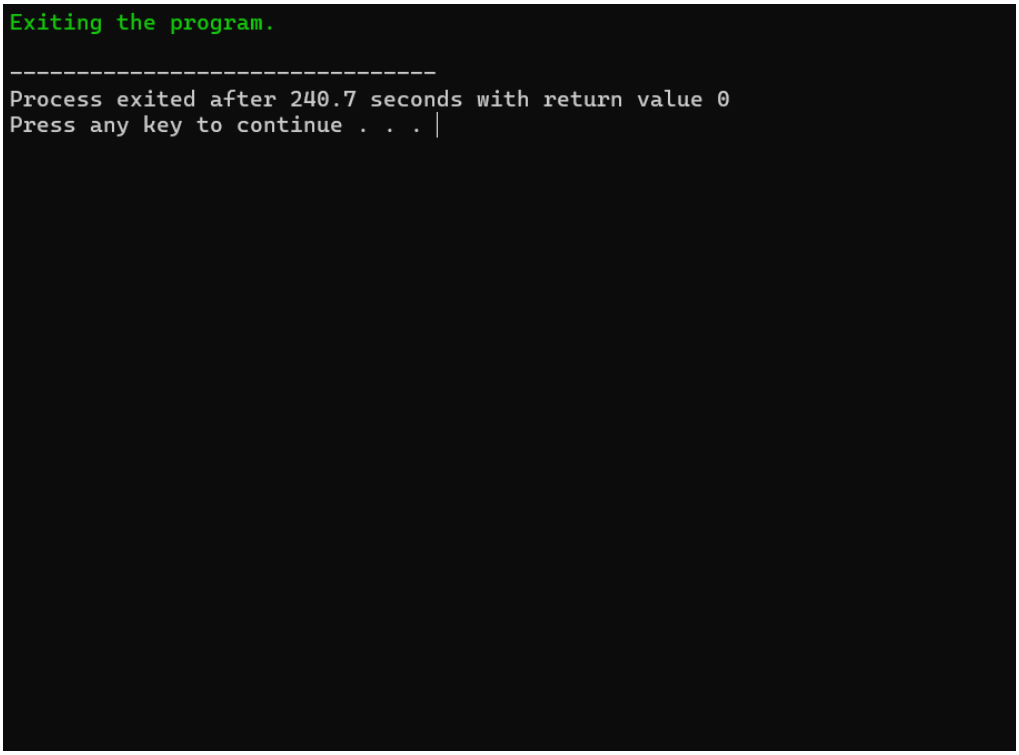
```
PRODUCTS TO BE RESTOCKED:

ID      Product Name                    Price   Quantity        Items Sold
-------------------------------------------------------------------------------
1       jelly                           67.00   7               120

CHOOSE YOUR OPTION (1-9):
1) Add product
2) Display products
3) Delete product
4) Update product
5) Search product
6) Product sold
7) Sorting
8) Restock
9) Exit

Your Choice: 9
```

15:- Exiting the inventory.


Exiting the program.

Process exited after 240.7 seconds with return value 0
Press any key to continue . . .