In this mini project, you are going to develop a console-based e-commerce application! This software will be used by customers to order grocery items. The detailed explanations about how you are going to develop this application are provided below (Note: example console output from the application is written in ORANGE).

Text-based User Interface:

When your program first runs, it will ask the user to log in by providing a user name and password as follows. In your code, at least, the following users should be already defined with user name 'ahmet' and password '1234', and 'zeynep' with password '4444'. You will have one **administrator** account with name "admin" and password "qwerty" You may define additional users if you like.

****Welcome to Medipol Online Market****

Please log in by providing your user credentials:

User Name:

If the user enters invalid credentials, the program should ask the user to try again. The user successfully logs in when the provided credentials are valid. If the user enters invalid credentials for the same user 3 time in row, the account will be blocked and the user will not be allowed to login. The program will warn the user in such a case. The following shows the interface for the initial login screen with a scenario where the user first enters incorrect credentials, and then successfully logs in during the second trial.

****Welcome to Medipol Online Market****

Please log in by providing your user credentials:

User Name: ahmet

Password: 123

Your user name and/or password is not correct. Please try again!

User Name: ahmet

Password: 1234

Successfully logged in!

The following shows the interface for the initial login screen again with a scenario where the user enters 3 incorrect credentials, and then a warning message will be shown (Note: Even if the user tries to log in later, their account should remain blocked).

****Welcome to Medipol Online Market****

Please log in by providing your user credentials:

User Name: ahmet

Password: 123

Once the user enters valid credentials, your program should greet the user with the following message, and provide a menu of the options to do. User will do a selection by entering the corresponding menu number, and accordingly different information will be shown and/or requested.

If the user enters an invalid menu entry, the program should warn the user to provide a valid menu number.

When **main menu item 1** is chosen, the user will be asked to provide search term(s). Then, your program will display a list of all items from inventory whose name includes the user-provided search term(s), and whose stock amount is greater than 0. For instance, when the user searches for "Juice", the program should display all items whose name includes 'juice' and available in the stocks as follows. Product name checking should be case insensitive (i.e., you should not differentiate between lower and upper case letters).

Note that each matching item is assigned a number, and the program asks the customer to enter the number corresponding to a product to add to his/her basket. A customer may cancel the current search session and go back to main menu by entering 0 at this stage. Once the customer chooses a product to add to the basket, the program will ask for the amount. If the requested amount is not exceeding the stock amount of the selected product in the inventory, the product should be added to the basket, and the customer should be notified about it, and then the main menu should be displayed.

If the requested amount is over the stock amount, then the program should ask the user to provide another amount (or Enter 0 to go back to main menu).

If the search result is empty, the user will be requested to provide another search term (or, the user may go back to main menu by entering 0).

Anywhere in the program, if the user makes a selection outside of the menu shown, the program must ask the user to re-enter a valid menu item.

When **main menu item 2** is chosen, the user will be able to see the things he/she added to their basket, the price of the item per piece, the amount of the items in the basket, and lastly, the total price of the items in the basket.

Your basket contains:

1.grape juice price=9$ amount=1 total=9.0$

2.egg price=2$ amount=2 total=4.0$

Total 13.0$

Please note that if the basket is empty, your program should show that the user's basket is empty, and the total price of items in the user's basket is equal to 0$. After displaying items in your basket, you'll be directed to another menu (to avoid confusion, refer to this menu as basket sub menu):

Please choose an option:

1.Update amount

2.Remove an item

3.Check out

4.Go back to main menu

Your selection:

If user chooses **basket sub-menu option 1**, user will select an item from their basket and update its amount. You should display the content of user's basket, after the changes are applied. Reminder: Again, the new amount should be checked to make sure that it is not exceeding the stock amount (Note: the inventory stock amounts of items in basket are not decreased until checkout).

Please select which item to change its amount: 1

Please type the new amount: 2

Your basket now contains:

1.grape juice price=9$ amount=2 total=18.0$

After this user will be redirected to the **basket sub-menu** again. If user chooses **basket sub-menu option 2**, user will select an item from their basket that will be removed from their basket. Again, please display the content of user's basket, after the removal. After this, user will be redirected to the **basket sub-menu** again.

If user chooses **basket sub-menu option 3**, or **main menu item 3**, user will proceed to check out and your program will print the receipt as shown below. For the last part of your receipt, program should display the real date and time of the transaction.

During checkout, the program should decrease the stock amount of each user-bought product properly in the inventory. After checkout, the program should show the main menu.

If user chooses **main menu item 4**, the user should log out, and the program should ask for user name and password again as explained at the beginning. At this stage, **the user's basket content should still be stored, even after logging out**. When the same user logs in

again, s/he should be able to see items that s/he added into the basket in his previous session(s).

If user chooses **main menu item 5**, the program should exit.

Now, let's come back to the admin account, if the admin enters valid credentials, your program should greet the admin and provide a menu of 3 options to do. Admin will do a selection by entering the corresponding menu number, and accordingly different information will be shown and/or requested.

Welcome, admin! Please choose one of the following options by entering the corresponding menu number.

Please choose one of the following services:

1. Activate User Account

2. Deactivate User Account

3. Add User

4. Remove User

5. Logout

6. Exit

Your Choice

In option 1, the blocked user account can be reactivated (unblocked) because of the entered 3 wrong credentials.

In option 2, the admin have the ability to deactivate a user account (block).

In option 3, the admin can add a new user.

In option 4, the admin can remove user.

In option 5, the admin should log out, and the program should ask for user name and password again as explained at the beginning.

In option 6, the program should exit.

The inventory is as follows

Product	Stock Amount	Price
Asparagus	10	5
Broccoli	15	6
Carrots	18	7
Apples	20	5
Banana	10	8
Berries	30	3
Eggs	50	2

Mixed fruit juice	0	8
Fish Sticks	25	12
Ice Cream	32	6
Apple Juice	40	7
Orange Juice	30	8
Grape Juice	10	9