

Department of Computer Science and Engineering

Course Title: OOP Lab I (Java)

Course Code: CSE 202

Class Diagram for OOP LAB I Project

Project name: Fruits Store PoS System

Group member info:

Shuhurat Fariha

Id:22201252

Nusrat Nafisa

Id:22201253

Maisha Sameha

Id:22201266

Fruits Store PoS System

The diagrams of the class files which will make this system possible are given below :

Fruit
private String fruitname; private Int quantity_O; private int quantity_C; private double price;
Fruit(String fruitname, Int quantity, double price) Fruit(String fruitname, double price, Int quantity) Fruit(String fruitname, Int quantity) getfruitname() getquantity_C(),getquantity_O(), getprice()

Owner_Info
Static string ownername
Static int id
Boolean can_access(int id) View_owner_info()

OwnerCart extends Fruit
storeitems=(ArrayList<Fruit >)
addstoreItem(Fruit fruit)
remove_Oneitem(String name)
removestoreItem(String name,int quantity)
clearstoreCart()
viewOwnerCartItems()

Customer

```
private customerID  
private customerName  
private password  
  
getCustomerID()  
getCustomerName()  
getPassword()  
getCustomerDetails():
```

Interface CustomerRegistration

```
registerCustomer(int id, String password): int
```

Interface DiscountProvider

```
applyDiscount(): double
```

CustomerCart extend OwnerCart

```
DiscountProvider, CustomerRegistration
```

```
CustomerList (ArrayList<Customer1>)
```

```
customeritems (ArrayList<Fruit >)
```

```
addItem(Fruit fruit )
```

```
removeItem(String name, int quantity)
```

```
clearCart()
```

```
getTotalPrice()
```

```
viewCartDetails()
```

PoS
main()

The functionalities which will be provided in the main method are:

As an owner:

1. To add fruits in store.
2. To remove fruits from store.
3. To view cart details.

As a User:

Check Customer (New or Existing)

1. Add fruits to Cart
2. Remove fruits from Cart
3. Add point in customer membership card
4. Apply Discount (implementing an interface)
5. View Cart Details
6. Exit

IMPLEMENTATION PART

Customer1

```
package fruitstore;

public class Customer1 {
    private int customerID;
    private String customerName;
    private String password;

    public Customer1(int customerID, String customerName, String password)
    {

        this.customerID = customerID;
        this.customerName = customerName;
        this.password = password;
    }

    public int getCustomerID() {
        return customerID;
    }

    public void setCustomerID(int customerID) {
        this.customerID = customerID;
    }

    public String getCustomerName() {
        return customerName;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public String getPassword() {
        return password;
    }
}
```

```

public void setPassword(String password) {
    this.password = password;
}

public void viewCustomerDetails() {
    System.out.println("customer id: "+getCustomerID());
    System.out.println("customer name: "+getCustomerName());

}

```

CustomerCart

```

package fruitstore;

import java.util.ArrayList;
import java.util.Scanner;

import javax.swing.JOptionPane;

public class CustomerCart extends OwnerCart implements DiscountProvider,
CustomerRegistration {

    Scanner input = new Scanner(System.in);

    public CustomerCart(String fruitName, double price, int quantity) {

        super(fruitName, price, quantity);
    }

    ArrayList<Customer1> CustomerList = new ArrayList<>();

    public static ArrayList<Fruit> customeritems = new ArrayList<>();
    //fruit(name,price)

    public int registerCustomer(int id, String pass) {
        int x = 1, y = 2, z = 3;
        for (Customer1 c1 : CustomerList) {

```

```

        if (c1.getCustomerID() == id) {
            if (c1.getPassword().equalsIgnoreCase(pass)) {
                c1.viewCustomerDetails();
                return x;
            } else {
                return y;
            }
        }
        return z;
    }

    public int addItem(Fruit fruit) {
        // searching that fruits and showing its availability
        int c = 0, c1 = 0;

        for (Fruit fru : storeitems) {
if (fru.getFruitName().equalsIgnoreCase(fruit.getFruitName())) {

        if (fruit.getQuantity_C() <= fru.getQuantity_O()) {

            int s = fruit.getQuantity_C();
            fruit.setQuantity_C(fruit.getQuantity_C());

            s = fru.getQuantity_O() - s;
            fru.setQuantity_O(s);

            if (s == 0) {

                remove_Oneitem(fru.getFruitName());// owner cart fruit remove
            } else {
                fru.setQuantity_O(s);// fruits quantity change in owner cart
            }
        }

        Fruit fruit_ = new Fruit(fru.getFruitName(), fruit.getQuantity_C(), fru.getPrice());
        customeritems.add(fruit_);
        JOptionPane.showMessageDialog(null, fruit.getQuantity_C() + " " +
        fru.getFruitName() + " are added to the cart.");
    }
}

```

```

} else if (fruit.getQuantity_C() > fru.getQuantity_O() && fru.getQuantity_O() != 0) {
    // sop(only fru.getQuantity_O() is available);
    // adding fru.getQuantity_O()
    fruit.setQuantity_C(fru.getQuantity_O());
Fruit fruit_ = new Fruit(fru.getFruitName(), fruit.getQuantity_C(), fru.getPrice());// customeritems.add(fruit_);
JOptionPane.showMessageDialog(null, "Only " + fru.getQuantity_O() + " " + fru.getFruitName() + " are available");
JOptionPane.showMessageDialog(null, fruit.getQuantity_C() + " " + fru.getFruitName() + " are added to the cart.");
    fru.setQuantity_O(0);
    remove_Oneitem(fru.getFruitName());
} else {
    // quantity 0
JOptionPane.showMessageDialog(null, fruit.getFruitName() + " are not available.");
}
} else {
    // throw exception maybe for unavailable
}
}
return 0;
}

public void removeItem(String name, int quantity) {
    int n = 0;

    for (Fruit fruit : customeritems) {
        if (fruit.getFruitName().equalsIgnoreCase(name)) {
            if (fruit.getQuantity_C() < quantity) {
                System.out.println(fruit.getFruitName() + "'s quantity is " + fruit.getQuantity_C());
                n = 1;
            }
            break;
        }
    }
    else if (fruit.getQuantity_C() == quantity) {
        customeritems.remove(fruit);
    }
}

```

```

        n = 1;
        viewCartDetails();
        break;
    } else if (fruit.getQuantity_C() > quantity) {
        int s = fruit.getQuantity_C() - quantity;
        fruit.setQuantity_C(s);
        System.out.println(fruit.toString());
        n = 1;
        break;
    }

}
if (n == 0) {
    System.out.println(name + " is not available in customer cart");
}
else if (n == 1) {
    for (int i = 0; i < storeitems.size(); i++) {

if(storeitems.get(i).getFruitName().equalsIgnoreCase(name)) {
    int s = storeitems.get(i).getQuantity_O() +quantity;
    storeitems.get(i).setQuantity_O(s);
}
    }
}
}

public void clearCart() {
    customeritems.clear();
}

public double getTotalPrice() {
    double totalPrice = 0.0;
    for (int i = 0; i < customeritems.size(); i++) {

totalPrice = totalPrice + ((customeritems.get(i).getQuantity_C()) *
(customeritems.get(i).getPrice())));
    }
    return totalPrice;
}

```

```

    }

    @Override
    public double applyDiscount() {
        double price = getTotalPrice();
        if (price >= 1000) {
            double discountprice = price - ((price * 10) / 100);
            return discountprice;
        }
        return 0;
    }

    public void viewCartDetails() {

        for (int i = 0; i < customeritems.size(); i++) {
            JOptionPane.showMessageDialog(null,
customeritems.get(i).getFruitName() + "price : "
                    + customeritems.get(i).getPrice() + " " +
"quantity =" + customeritems.get(i).getQuantity_C());
        }
    }
}

```

CustomerRegistration

```

package fruitstore;

public interface CustomerRegistration {
    abstract public int registerCustomer(int id, String password);
}

```

DiscountProvider

```
package fruitstore;

public interface DiscountProvider {
    abstract public double applyDiscount();

}
```

Fruit

```
package fruitstore;

public class Fruit {

    private String fruitName;
    private double price;
    private int quantity_C;
    private int quantity_O;

    //for owner input
    public Fruit(String fruitName, double price, int quantity) {

        this.fruitName = fruitName;
        this.price = price;
        this.quantity_O = quantity;
    }

    public Fruit(String fruitName, int quantity, double price)
    {
        this.fruitName = fruitName;
        this.price = price;
        this.quantity_C= quantity;
    }

    //for customer input
    public Fruit(String fruitName,int quantity) {

        this.fruitName = fruitName;
        this.quantity_C = quantity;
        this.price = getPrice();
    }
}
```

```
}

//to search by name
public Fruit(String fruitName)
{
    this.fruitName = fruitName;
}

public String getFruitName() {
    return fruitName;
}

public void setFruitName(String fruitName) {
    this.fruitName = fruitName;
}

public double getPrice() {
    return price;
}

public void setPrice(double price) {
    this.price = price;
}

public int getQuantity_C() {
    return quantity_C;
}

public void setQuantity_C(int quantity_C) {
    this.quantity_C = quantity_C;
}

public int getQuantity_O() {
    return quantity_O;
}

public void setQuantity_O(int quantity_O) {
    this.quantity_O = quantity_O;
}
@Override
```

```

        public String toString() {
            return "Fruit [fruitName=" + fruitName + ", price=" + price +",
quantity =" + quantity_C + "]";
        }
        public String viewowneritems()
        {
            return "Fruit [fruitName=" + fruitName + ", price=" + price +",
quantity =" + quantity_O + "]";
        }
    }
}

```

OwnerCart

```

package fruitstore;
import java.util.ArrayList;
import javax.swing.JOptionPane;

public class OwnerCart extends Fruit {

    public OwnerCart(String fruitName, double price, int quantity)
    {
        super(fruitName, price, quantity);
    }

    public static ArrayList<Fruit> storeitems = new ArrayList<>();

    public void addstoreItem(Fruit fruit)
    {
        int n = 0;
        for (int i = 0; i < storeitems.size(); i++)
        {
if (storeitems.get(i).getFruitName().equalsIgnoreCase(fruit.getFruitName()))
{
            int s = storeitems.get(i).getQuantity_O();
            s = s + fruit.getQuantity_O();
            storeitems.get(i).setQuantity_O(s);
}
}
}

```

```

        n = 1;
        break;
    }
}
if (n == 0)
{
    storeitems.add(fruit);

}

}

public void remove_Oneitem(String name)// remove one item
{
    for (int i = 0; i < storeitems.size(); i++)
    {
        if (storeitems.get(i).getFruitName().equalsIgnoreCase(name))
        {
            storeitems.remove(i);

            break;
        }
    }

}

public void removestoreItem(String name, int quantity)// remove specific
items
{
    int n = 0;
    for (int i = 0; i < storeitems.size(); i++)
{
    if (storeitems.get(i).getFruitName().equalsIgnoreCase(name))
    {
        if (storeitems.get(i).getQuantity_O() > quantity)
        {
            int s = storeitems.get(i).getQuantity_O();
            s = s - quantity;
        }
    }
}

```

```

        storeitems.get(i).setQuantity_O(s);

JOptionPane.showMessageDialog(null, quantity+name +"s have been removed
from ownercart");
        n = 1;
    }
    else if (storeitems.get(i).getQuantity_O() == quantity)
    {
        storeitems.remove(i);
JOptionPane.showMessageDialog(null, name + " have been removed from
ownercart");
        n = 1;
    }
    else if(storeitems.get(i).getQuantity_O() < quantity)
    {
JOptionPane.showMessageDialog(null,"Only " +
storeitems.get(i).getQuantity_O()+" "+ storeitems.get(i).getFruitName() + " are
available in owner cart");
        n=1;
    }
}
if (n == 0)
{
JOptionPane.showMessageDialog(null, name + " is unavailable in ownercart");

}

public void clearstoreCart()
{
    storeitems.clear();
    JOptionPane.showMessageDialog(null,"All the fruits from the
owner's cart have been removed.");
}

public void viewOwnerCartItems()
{
    if (storeitems.size() != 0)

```

```

    {
        for (Fruit fru : storeitems)
        {

            JOptionPane.showMessageDialog(null,fru.getFruitName() + " - price : " +
fru.getPrice() + " quantity = " + fru.getQuantity_O());
        }
    }
    else {
        JOptionPane.showMessageDialog(null, "Owner cart is
empty.");
    }
}

```

OwnerInfo

```

package fruitstore;

public class OwnerInfo {
    public static String ownername="rahim";
    public static final String password = "uap";

    public boolean can_access(String pass)
    {
        if(password.equalsIgnoreCase(pass))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

```

PoS

```
package fruitstore;

import java.util.InputMismatchException;
import java.util.Scanner;

import javax.swing.JOptionPane;

public class PoS {

    static double c = 0.0;

    public static void customerSystem()// Accessing user cart by this method
    {
        CustomerCart cart = new CustomerCart(null, 0.0, 0);

        Scanner scan = new Scanner(System.in);

        // customer cart management
        while (true)// 1st while loop in this method
        {
            try {
                String[] customer_Options = { "Add fruits to cart", "Remove fruits from cart",
                    "Apply Discount", "View Cart Details", "Exit" };

                int choice = JOptionPane.showOptionDialog(null, "Customer", "Fruit store - "
                    + "customer '-' ", 0, JOptionPane.QUESTION_MESSAGE, null, customer_Options,
                    customer_Options[0]);
            }

            // for customer menu

            // 1st try block
            try {

                if (choice == 0)// adding fruits in customers fruit list
                {
```

```

String type = JOptionPane.showInputDialog("Enter how many types of fruit
you want to ADD : ");

int type_ = Integer.parseInt(type);

for (int j = 0; j < type_; j++)
{
    String name = JOptionPane.showInputDialog("Enter fruit name:");

    String quantity = JOptionPane.showInputDialog("Enter quantity");

    int quantity_ = Integer.parseInt(quantity);

    Fruit fruit = new Fruit(name, quantity_);

    cart.addItem(fruit);
}

}

else if (choice == 1)// removing fruits from customer fruit list
{
String[] customer_remove_Options = { "Clear the full cart ", "Remove a single
fruit" };

int number = JOptionPane.showOptionDialog(null, "Owner", "Fruit store -Owner",
0,JOptionPane.QUESTION_MESSAGE, null,
customer_remove_Options,customer_remove_Options[0]);

// clear cart
if (number == 0)
{
cart.clearCart();

JOptionPane.showMessageDialog(null, "All fruits are removed successfully");
}
else if (number == 1)// removing one fruit item
{
String name = JOptionPane.showInputDialog("Enter fruit name:");

```

```
String quantity = JOptionPane.showInputDialog("Enter fruit quantity to remove from cart:");

int quantity_ = Integer.parseInt(quantity);

cart.removeItem(name, quantity_);
}

}

else if (choice == 2)// discount on total price
{
JOptionPane.showMessageDialog(null, "Total Price : " + cart.getTotalPrice());

c = cart.applyDiscount();
}

else if (choice == 3)// view cart items
{
JOptionPane.showMessageDialog(null, "Cart Items: ");

cart.viewCartDetails();
if (c == 0.0)
{
JOptionPane.showMessageDialog(null, "Total Price: " + cart.getTotalPrice());
}
else
{
JOptionPane.showMessageDialog(null, "Total Price: " + cart.applyDiscount());

}
}

else if (choice == 4)
{
cart.clearCart();
break;
}
} // end of 1st try block

catch (InputMismatchException e)
{
JOptionPane.showMessageDialog(null, "Exception Message: " + e + "\nPlease enter a valid input.");
```

```
scan.nextLine();
}
} catch (Exception e)
{
JOptionPane.showMessageDialog(null, "An error occurred: " + e.getMessage());
}

} // end of 1st while loop in this method

}// end of customerSystem

// main method starts
public static void main(String[] args) {

OwnerCart cart1 = new OwnerCart(null, 0.0, 0);
CustomerCart cart = new CustomerCart(null, 0.0, 0);
Scanner scanner = new Scanner(System.in);

// 1st while loop starts in main
while (true)
{
try {
String[] options = { "Owner", "Customer", "Exit" };

int option = JOptionPane.showOptionDialog(null, "Select User Type:", "Fruit store :",
0, 1, null,options[0]);

// 1st try block in main
try {

// owner store management
if (option == 0) // 1st if block starts
{

String pass = JOptionPane.showInputDialog("Enter Owner password:");

OwnerInfo owner = new OwnerInfo();

if (owner.can_access(pass))// 2nd if block
```

```

{
while (true)// 2nd while loop starts
{
String[] Owner_Options = { "Add fruits to store ", "Remove fruits from store",
"View store Details", "Exit" };

int choice = JOptionPane.showOptionDialog(null, "Owner", "Fruit store -Owner",
0,JOptionPane.QUESTION_MESSAGE, null, Owner_Options,
Owner_Options[0]);

// 2nd try block in main
try {

if (choice == 0)// add fruits in owner cart
{
String type = JOptionPane
.showInputDialog("Enter how many types of fruit you want to ADD : ");
int type_ = Integer.parseInt(type);

for (int j = 0; j < type_; j++) {
String name = JOptionPane.showInputDialog("Enter fruit name:");

String price = JOptionPane.showInputDialog("Enter fruit price:");

double price_ = Double.parseDouble(price);

String quantity = JOptionPane.showInputDialog("Enter quantity");

int quantity_ = Integer.parseInt(quantity);

Fruit fruit = new Fruit(name, price_, quantity_);

cart1.addstoreItem(fruit);
}

}
else if (choice == 1)// removing fruits from owner cart
{

String[] Owner_remove_Options = { "Clear the full cart ",
```

```
"Remove a single fruit" };

int number = JOptionPane.showOptionDialog(null, "Owner", "Fruit store - Owner",0, JOptionPane.QUESTION_MESSAGE, null, Owner_remove_Options,Owner_remove_Options[0]);

if (number == 0) // clearing full cart
{
    cart1.clearstoreCart();

    JOptionPane.showMessageDialog(null, "All fruits are removed successfully.");
}
else if (number == 1)// removing one fruit from store
{
    String name = JOptionPane.showInputDialog("Enter fruit name:");

    String quantity = JOptionPane.showInputDialog("Enter fruit quantity:");

    int quantity_ = Integer.parseInt(quantity);

    int removed = 0;

    int c = 0;
    cart1.removestoreItem(name, quantity_);
}
else if (choice == 2)// view owner cart items
{
    cart.viewOwnerCartItems();// customer fruit list
}
else if (choice == 3)
{
    break;
}
else
{
    JOptionPane.showMessageDialog(null,"\\nInvalid choice. Please enter a valid option.");
}
```

```
        }
    } // end of 2nd try block
    catch (InputMismatchException e)
    {
        JOptionPane.showMessageDialog(null,"Exception Message: " + e + "\nPlease
enter a valid input.\n");
    }
} // end of 2nd while loop
} // end of 2nd if block
else
{
    JOptionPane.showMessageDialog(null, "Invalid password.Access denied!");
}

}
// end of 1st if block

// user account access starts
else if (option == 1)
{
    // Customer management system

String id = JOptionPane.showInputDialog("Enter Customer Id");

int id_ = Integer.parseInt(id);

String password = JOptionPane.showInputDialog("Enter Password");

int s = cart.registerCustomer(id_, password);

if (s == 1)
{
    JOptionPane.showMessageDialog(null, "customer is already registered!");

customerSystem();

}
else if (s == 2)
{
```

```
JOptionPane.showMessageDialog(null, "invalid password!");

}

else if (s == 3)
{
JOptionPane.showMessageDialog(null, "customer has not been registered yet");

String name = JOptionPane.showInputDialog("Enter Customer name");

Customer1 customer = new Customer1(id_, name, password);

cart.CustomerList.add(customer);

JOptionPane.showMessageDialog(null, "Customer has been registered!");

customerSystem();

}

}

// to exit the whole program
else if (option == 2)
{
JOptionPane.showMessageDialog(null,"Thank you for visiting our online fruit
store.Hope you enjoyed!!\n");

break;
}
} // end of try block 2

catch (InputMismatchException e)
{
JOptionPane.showMessageDialog(null, "Exception Message: " + e + "\nPlease
enter a valid input.\n");

}

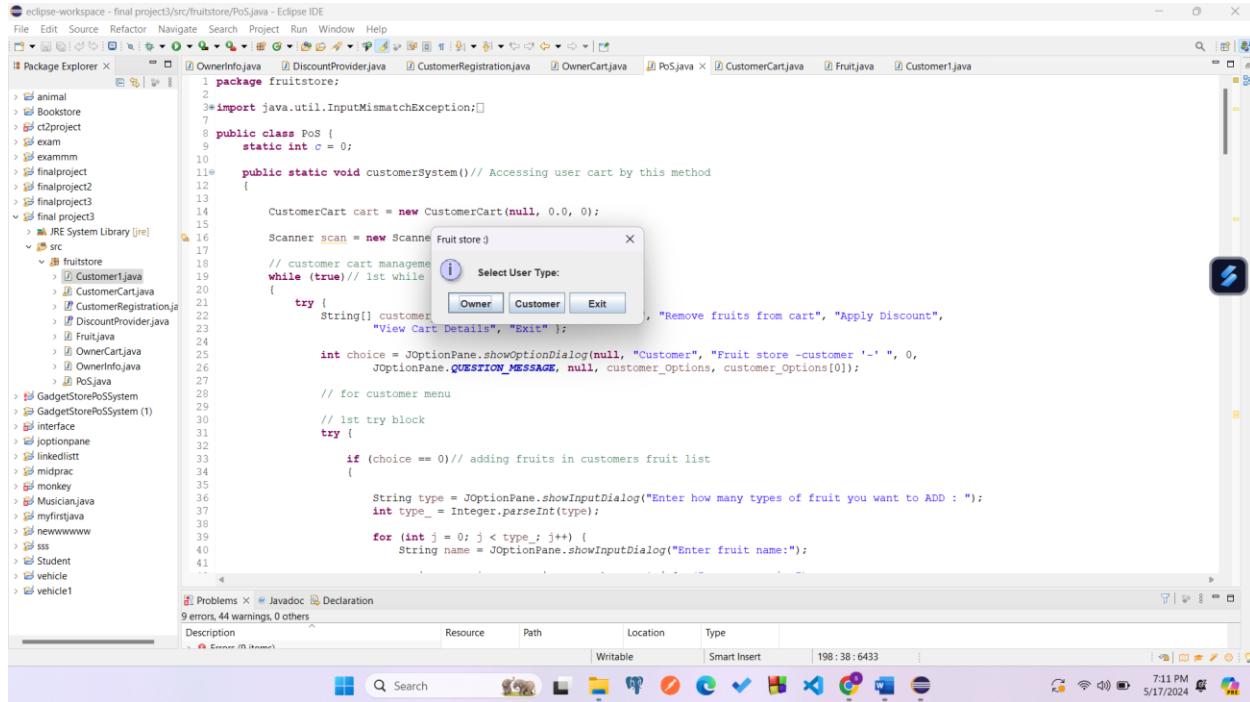
}

catch (Exception e)
{
```

```
JOptionPane.showMessageDialog(null, "An error occurred: " + e.getMessage());  
}  
  
} // end of 1st while loop in main method  
  
}// end of main method  
  
}// end of poS class
```

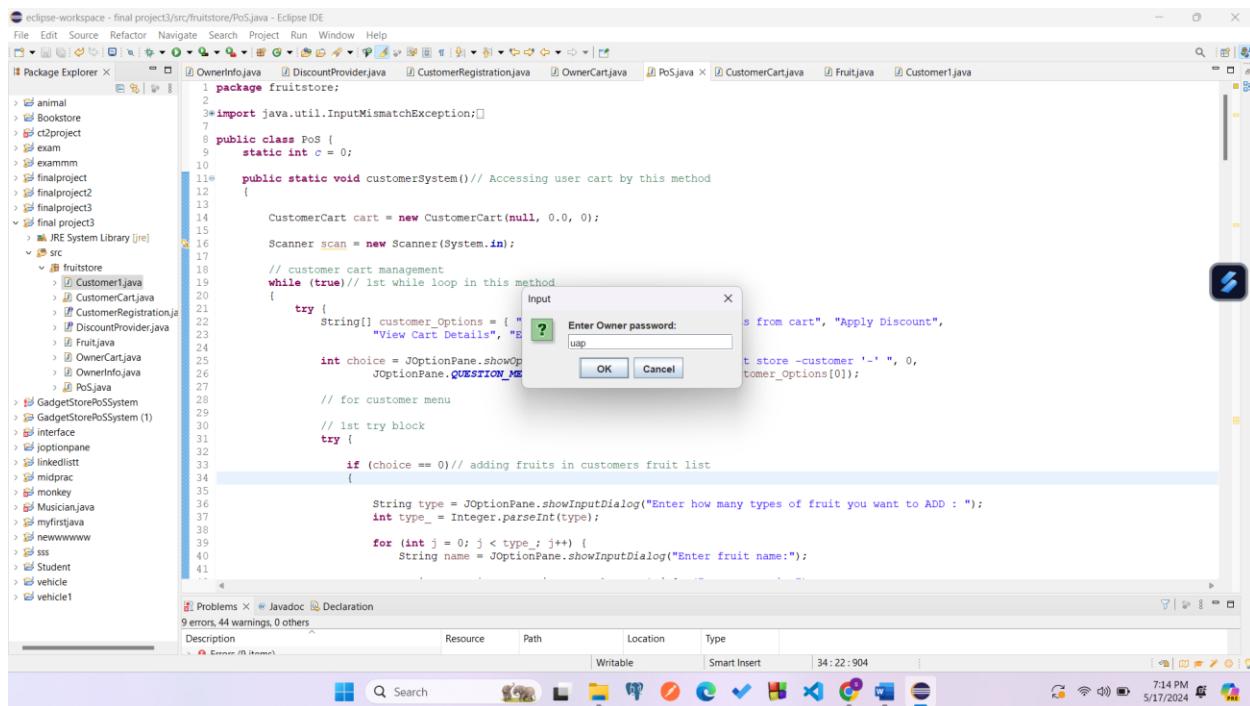
Input output:

1.owner login:



Screenshot of the Eclipse IDE interface showing the PoS.java file in the editor. A modal dialog titled "Select User Type:" is displayed, containing three buttons: "Owner", "Customer", and "Exit". The code in the editor shows a switch statement based on the user type selected.

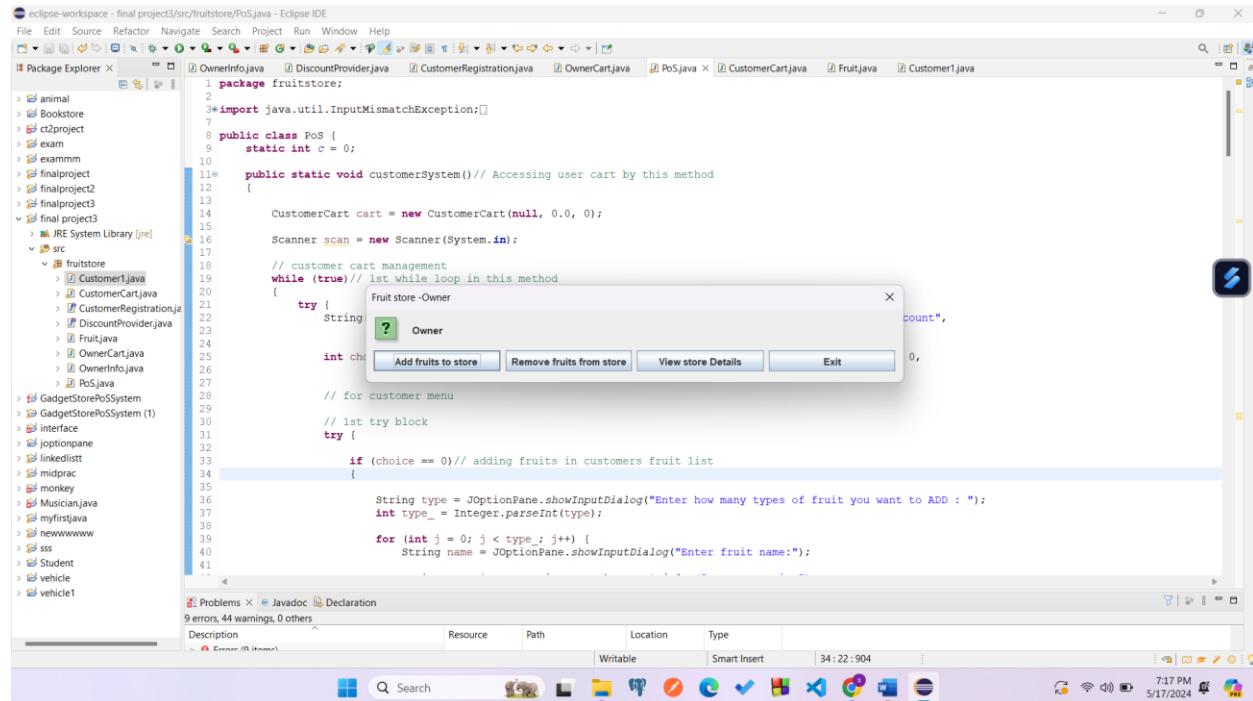
```
1 package fruitstore;
2
3 import java.util.InputMismatchException;
4
5 public class PoS {
6     static int c = 0;
7
8     public static void customerSystem()// Accessing user cart by this method
9     {
10
11         CustomerCart cart = new CustomerCart(null, 0.0, 0);
12
13         Scanner scan = new Scanner(Fruit store);
14
15         // customer cart management
16         while (true)// 1st while
17         {
18             try {
19                 String[] customer_Options = ["View Cart Details", "Remove fruits from cart", "Apply Discount",
20                     "Exit"];
21
22                 int choice = JOptionPane.showOptionDialog(null, "Customer", "Fruit store -customer '-' ", 0,
23                     JOptionPane.QUESTION_MESSAGE, null, customer_Options, customer_Options[0]);
24
25                 if (choice == 0)// adding fruits in customers fruit list
26                 {
27
28                     String type = JOptionPane.showInputDialog("Enter how many types of fruit you want to ADD : ");
29                     int type_= Integer.parseInt(type);
30
31                     for (int j = 0; j < type_; j++) {
32                         String name = JOptionPane.showInputDialog("Enter fruit name:");
33
34                     }
35
36                     String type = JOptionPane.showInputDialog("Enter how many types of fruit you want to ADD : ");
37                     int type_= Integer.parseInt(type);
38
39                     for (int j = 0; j < type_; j++) {
40                         String name = JOptionPane.showInputDialog("Enter fruit name:");
41
42                     }
43
44                 }
45
46             }
47             catch (InputMismatchException e) {
48                 System.out.println("Please enter integer value");
49             }
50
51             // for customer menu
52
53             // 1st try block
54             try {
55
56                 if (choice == 0)// adding fruits in customers fruit list
57                 {
58
59                     String type = JOptionPane.showInputDialog("Enter how many types of fruit you want to ADD : ");
60                     int type_= Integer.parseInt(type);
61
62                     for (int j = 0; j < type_; j++) {
63                         String name = JOptionPane.showInputDialog("Enter fruit name:");
64
65                     }
66
67                 }
68
69             }
70             catch (InputMismatchException e) {
71                 System.out.println("Please enter integer value");
72             }
73
74             // for customer menu
75
76             // 1st try block
77             try {
78
79                 if (choice == 0)// adding fruits in customers fruit list
80                 {
81
82                     String type = JOptionPane.showInputDialog("Enter how many types of fruit you want to ADD : ");
83                     int type_= Integer.parseInt(type);
84
85                     for (int j = 0; j < type_; j++) {
86                         String name = JOptionPane.showInputDialog("Enter fruit name:");
87
88                     }
89
90                 }
91
92             }
93             catch (InputMismatchException e) {
94                 System.out.println("Please enter integer value");
95             }
96
97         }
98
99     }
100 }
```



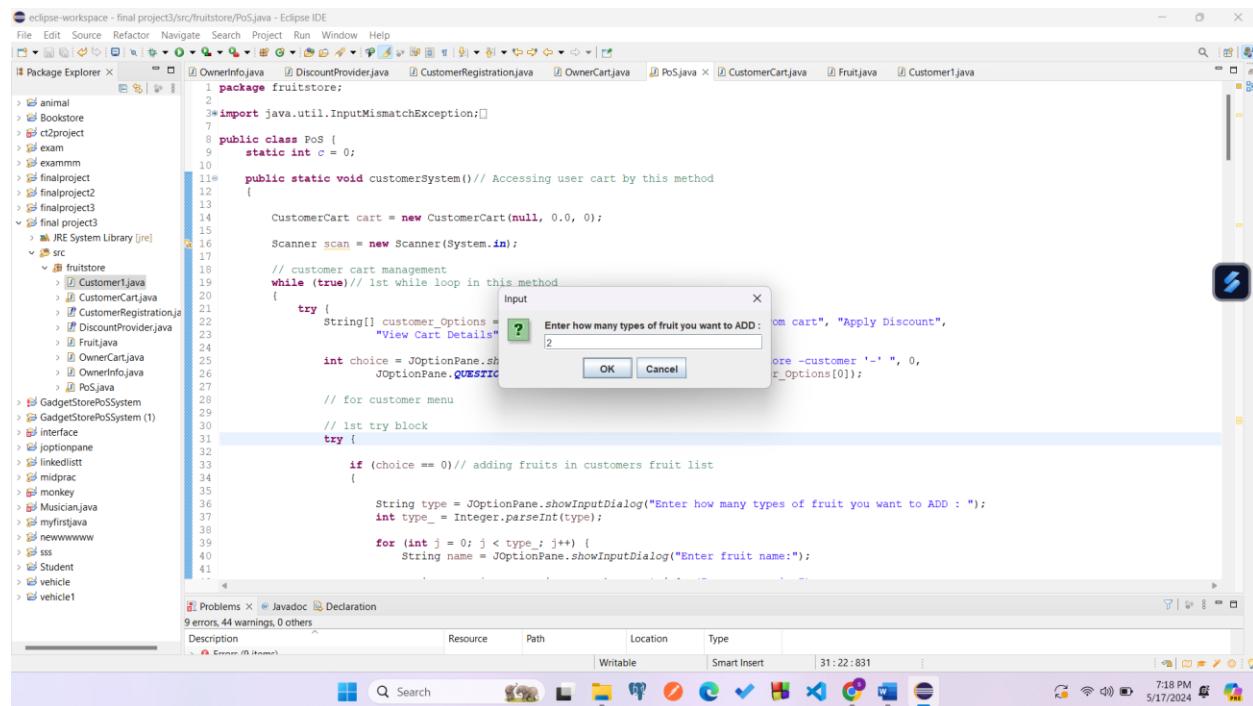
Screenshot of the Eclipse IDE interface showing the PoS.java file in the editor. A modal dialog titled "Enter Owner password:" is displayed, containing an input field and "OK" and "Cancel" buttons. The code in the editor shows a switch statement based on the user type selected.

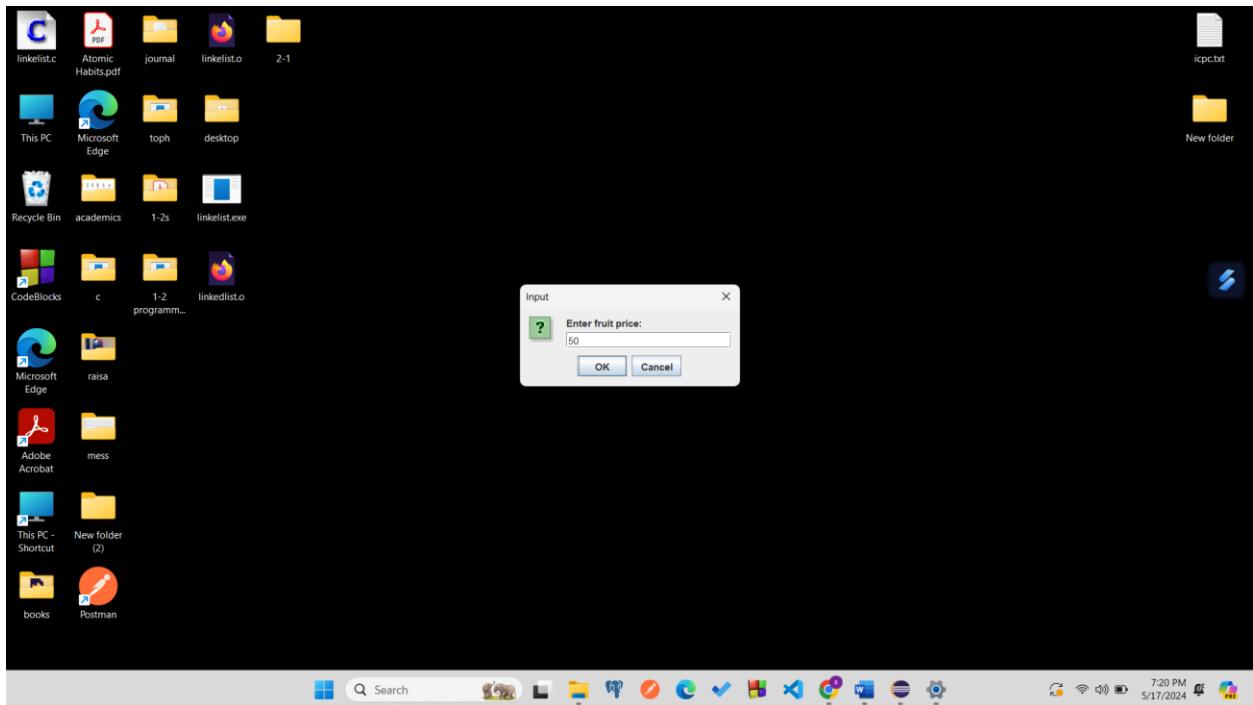
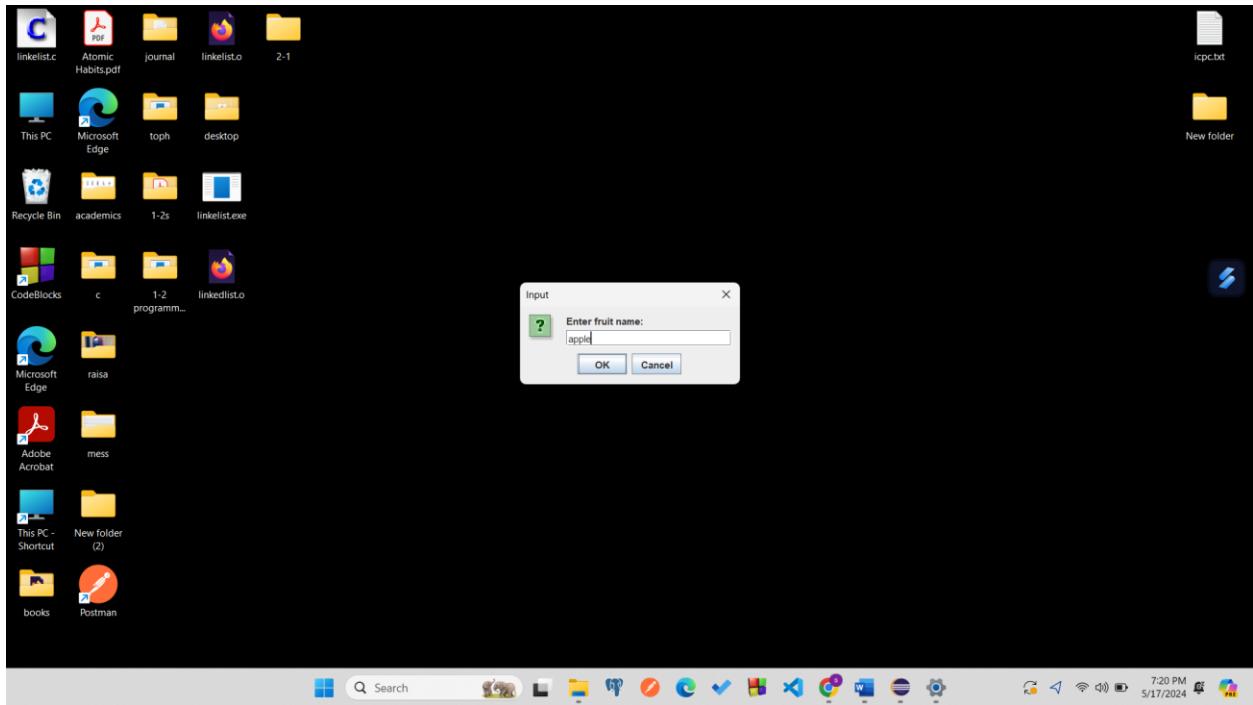
```
1 package fruitstore;
2
3 import java.util.InputMismatchException;
4
5 public class PoS {
6     static int c = 0;
7
8     public static void customerSystem()// Accessing user cart by this method
9     {
10
11         CustomerCart cart = new CustomerCart(null, 0.0, 0);
12
13         Scanner scan = new Scanner(System.in);
14
15         // customer cart management
16         while (true)// 1st while loop in this method
17         {
18             try {
19                 String[] customer_Options = ["View Cart Details", "Remove fruits from cart", "Apply Discount",
20                     "Exit"];
21
22                 int choice = JOptionPane.showOptionDialog(null, "Customer", "Fruit store -customer '-' ", 0,
23                     JOptionPane.QUESTION_MESSAGE, null, customer_Options, customer_Options[0]);
24
25                 if (choice == 0)// adding fruits in customers fruit list
26                 {
27
28                     String type = JOptionPane.showInputDialog("Enter how many types of fruit you want to ADD : ");
29                     int type_= Integer.parseInt(type);
30
31                     for (int j = 0; j < type_; j++) {
32                         String name = JOptionPane.showInputDialog("Enter fruit name:");
33
34                     }
35
36                     String type = JOptionPane.showInputDialog("Enter how many types of fruit you want to ADD : ");
37                     int type_= Integer.parseInt(type);
38
39                     for (int j = 0; j < type_; j++) {
40                         String name = JOptionPane.showInputDialog("Enter fruit name:");
41
42                     }
43
44                 }
45
46             }
47             catch (InputMismatchException e) {
48                 System.out.println("Please enter integer value");
49             }
50
51             // for customer menu
52
53             // 1st try block
54             try {
55
56                 if (choice == 0)// adding fruits in customers fruit list
57                 {
58
59                     String type = JOptionPane.showInputDialog("Enter how many types of fruit you want to ADD : ");
60                     int type_= Integer.parseInt(type);
61
62                     for (int j = 0; j < type_; j++) {
63                         String name = JOptionPane.showInputDialog("Enter fruit name:");
64
65                     }
66
67                 }
68
69             }
70             catch (InputMismatchException e) {
71                 System.out.println("Please enter integer value");
72             }
73
74             // for customer menu
75
76             // 1st try block
77             try {
78
79                 if (choice == 0)// adding fruits in customers fruit list
80                 {
81
82                     String type = JOptionPane.showInputDialog("Enter how many types of fruit you want to ADD : ");
83                     int type_= Integer.parseInt(type);
84
85                     for (int j = 0; j < type_; j++) {
86                         String name = JOptionPane.showInputDialog("Enter fruit name:");
87
88                     }
89
90                 }
91
92             }
93             catch (InputMismatchException e) {
94                 System.out.println("Please enter integer value");
95             }
96
97         }
98
99     }
100 }
```

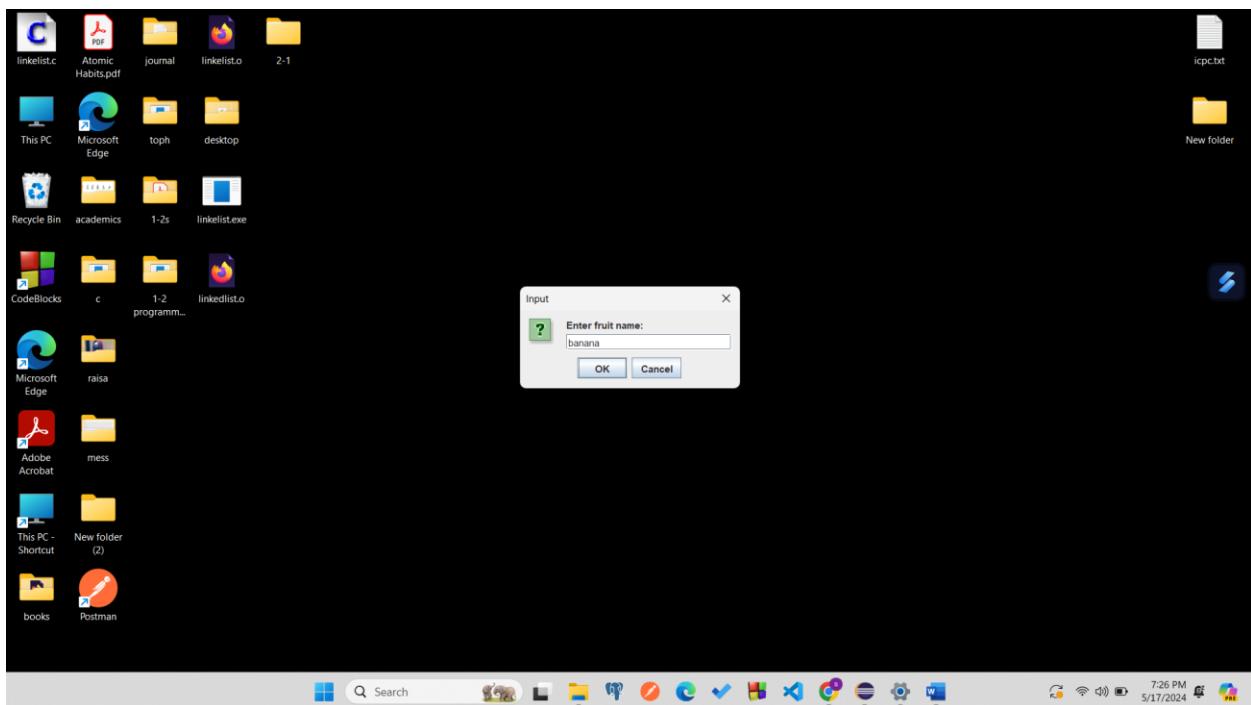
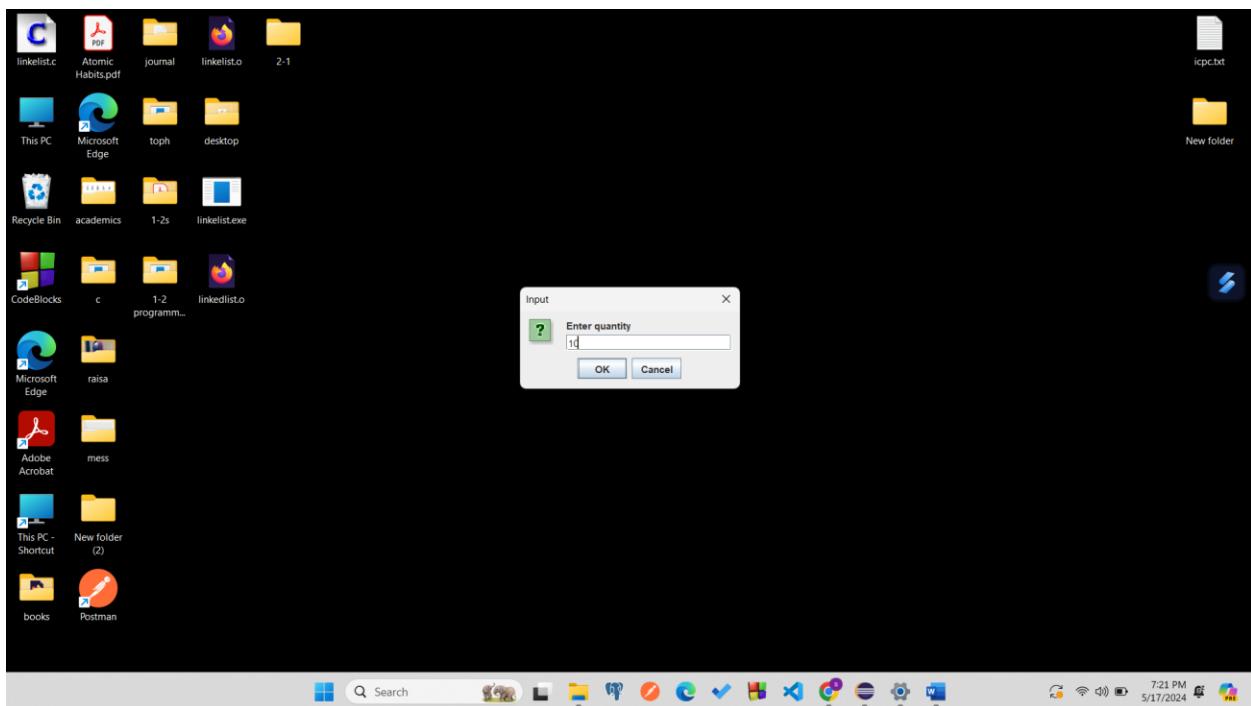
Owner options:

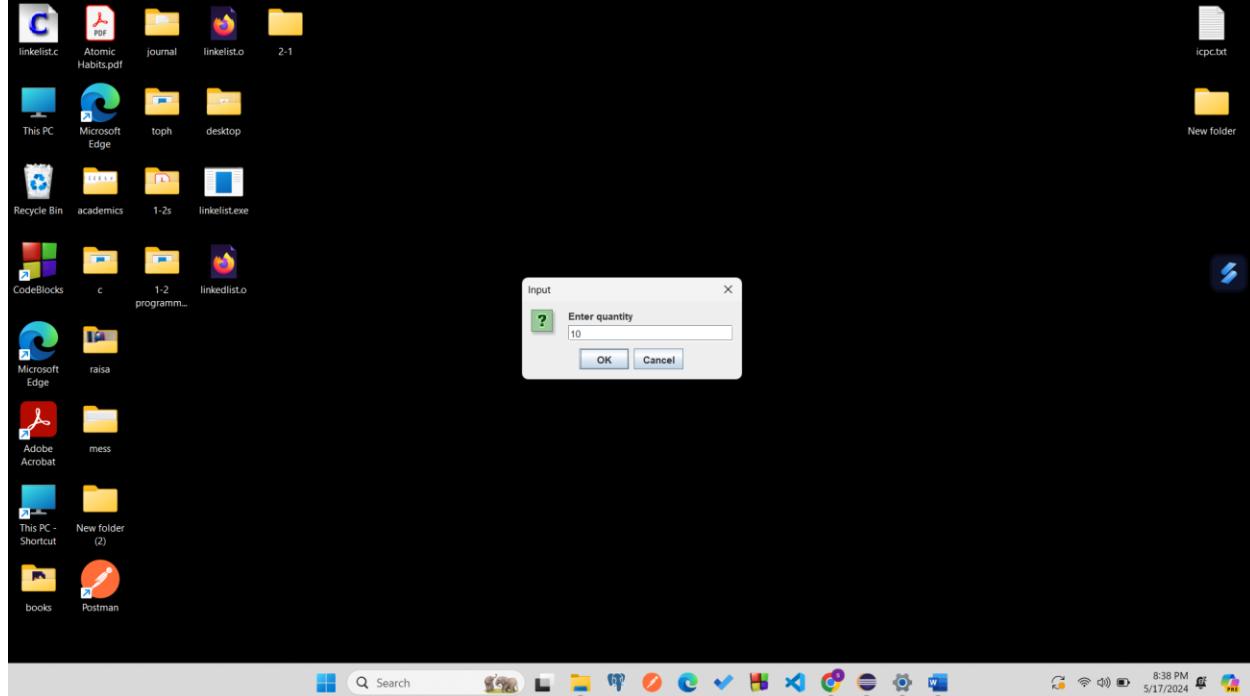
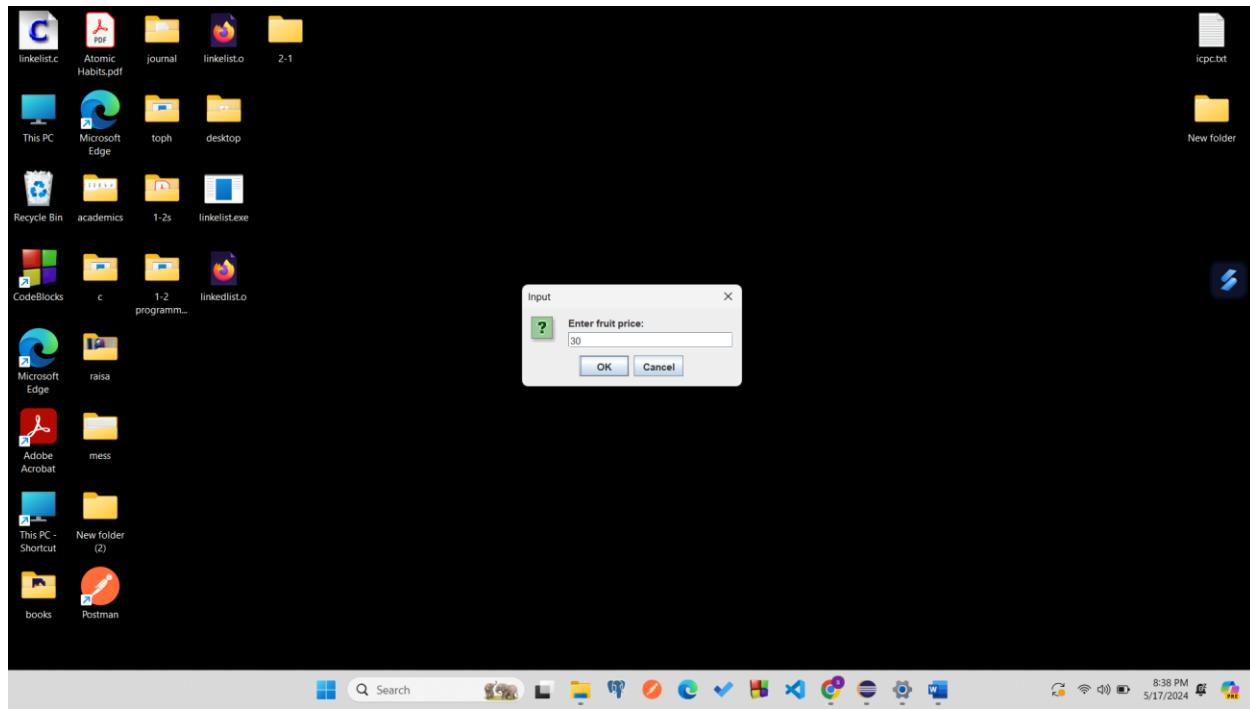


Adding fruits:

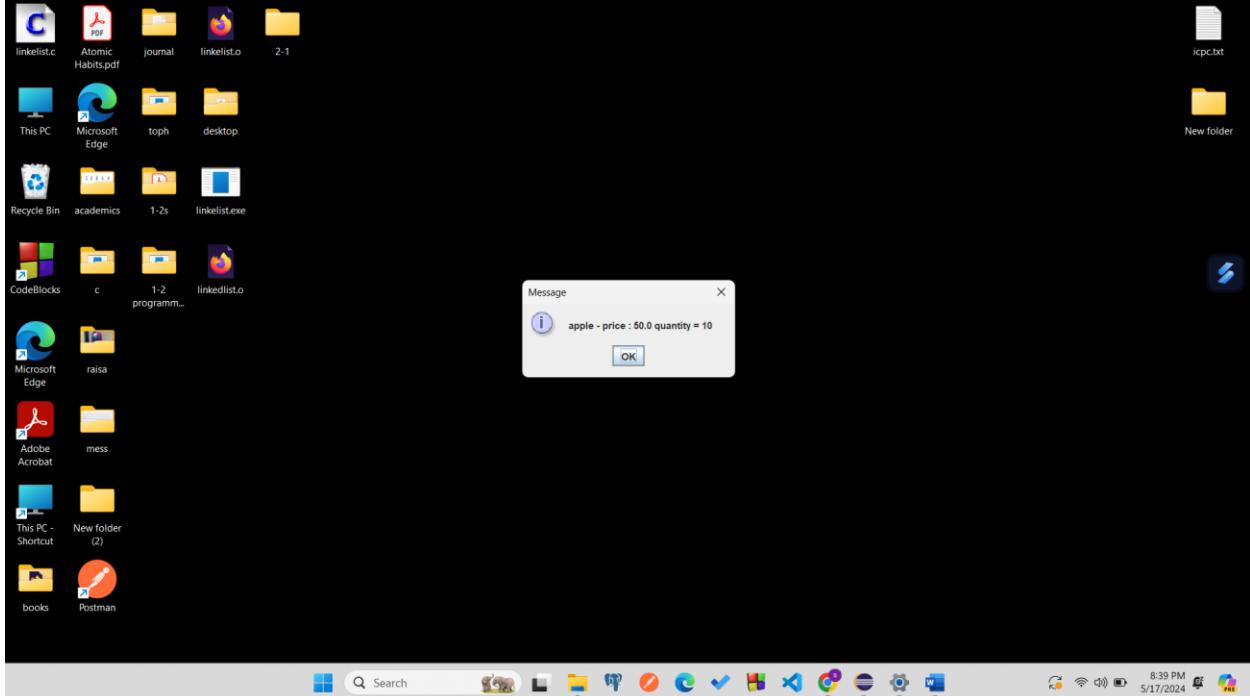
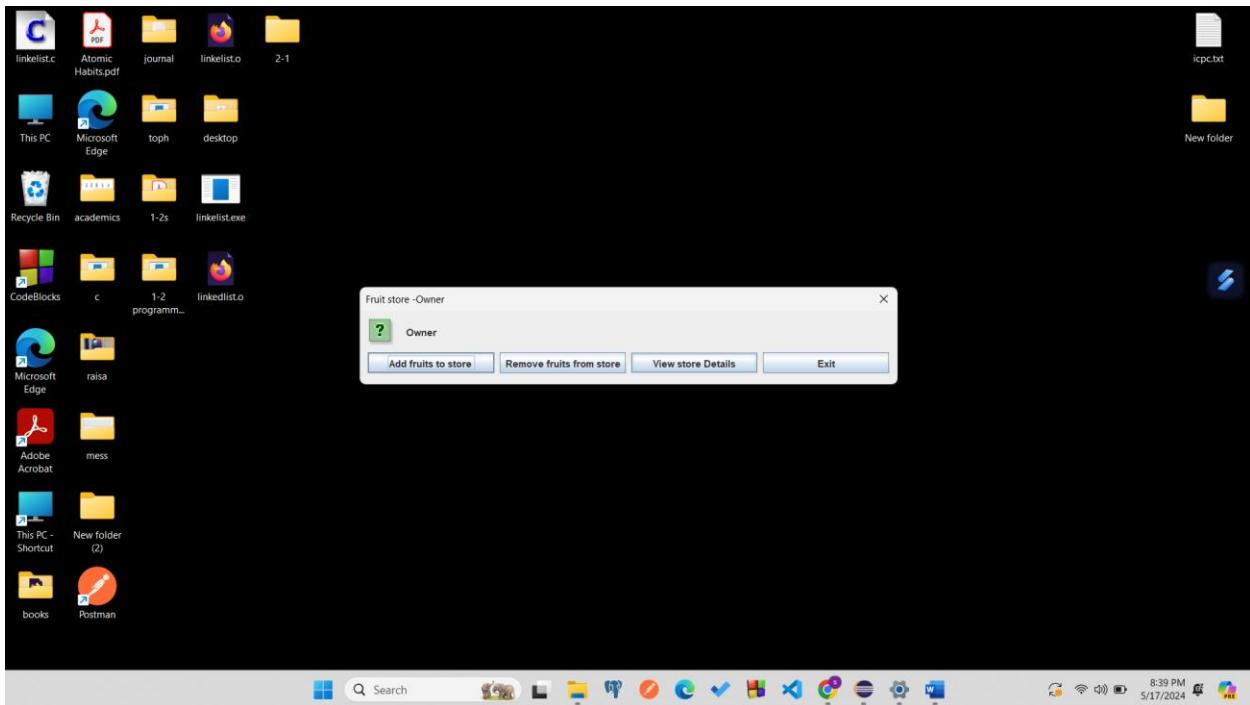


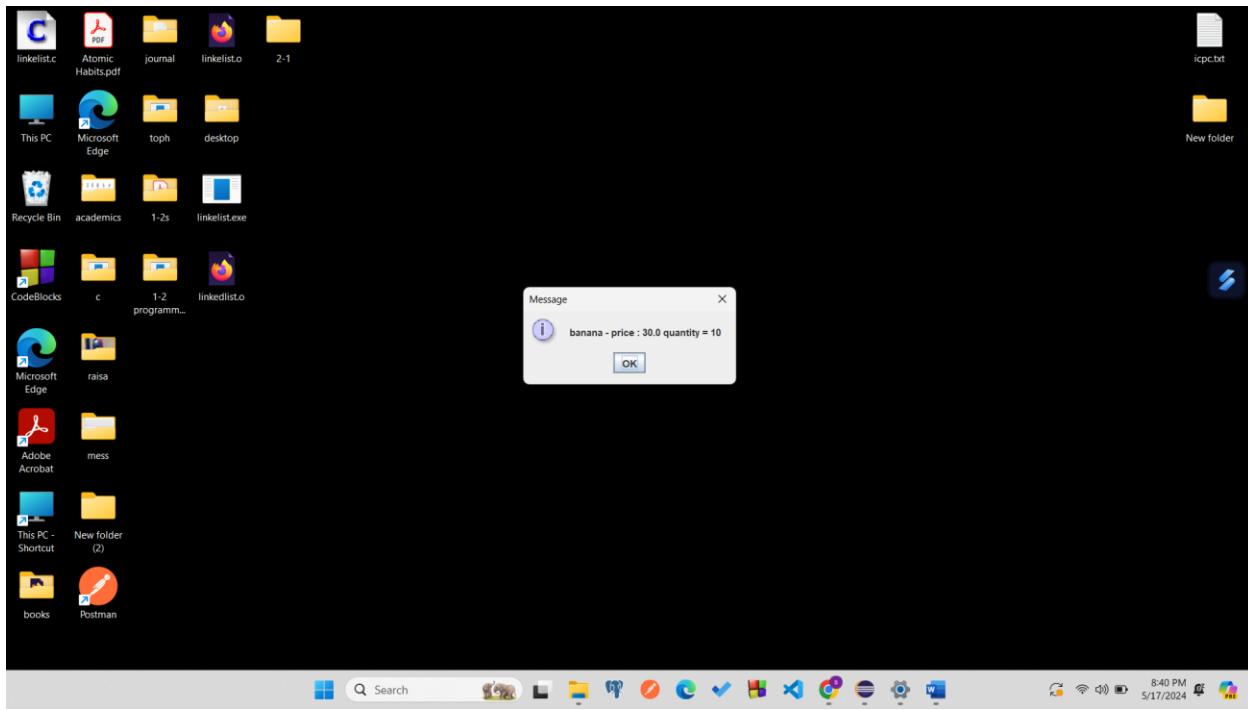




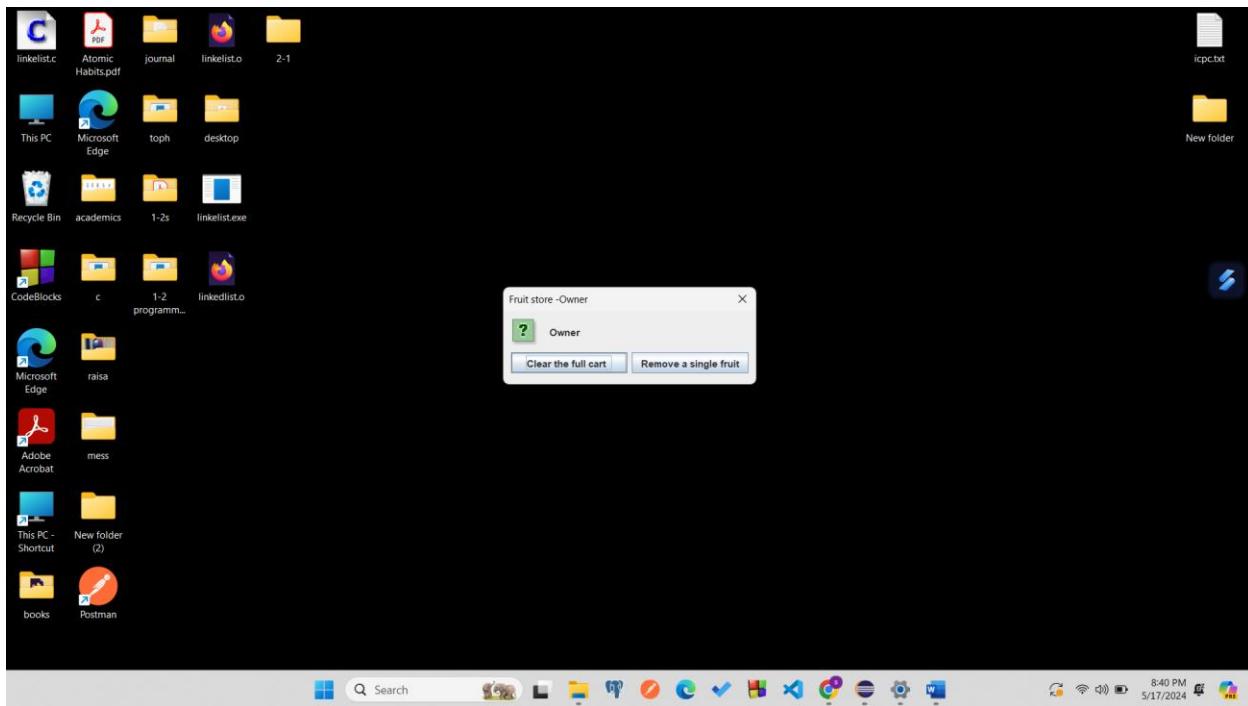


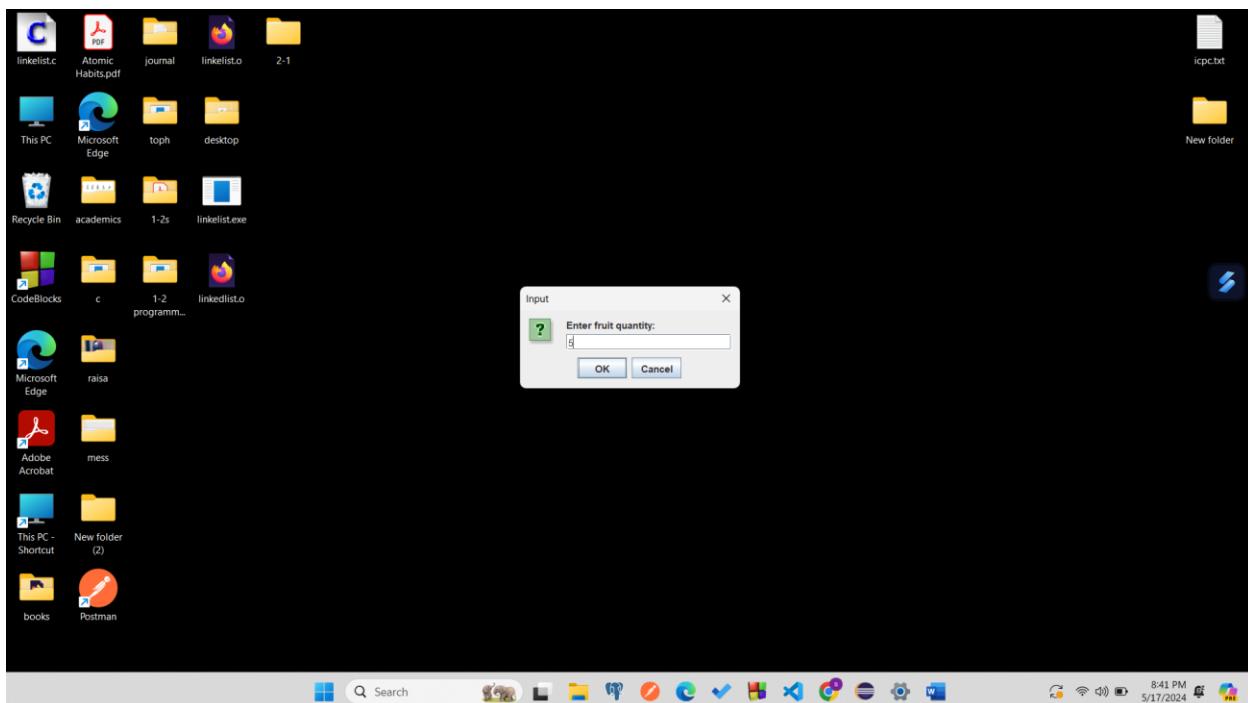
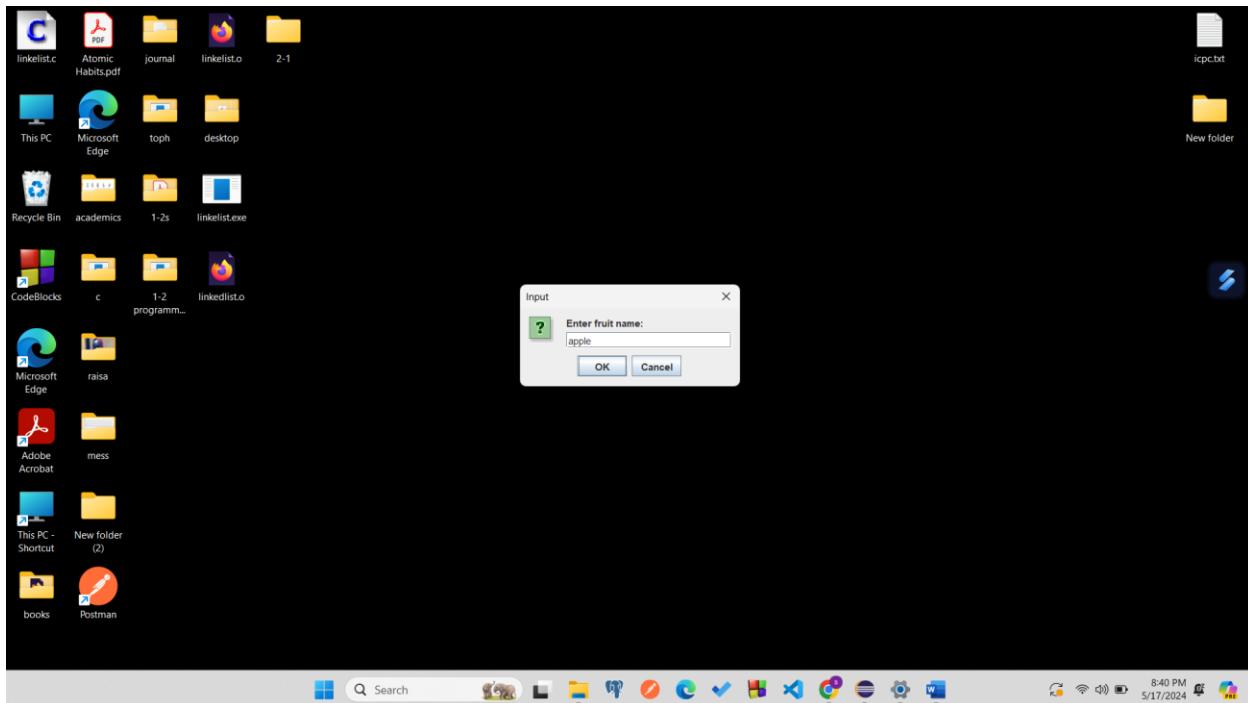
View cart details:

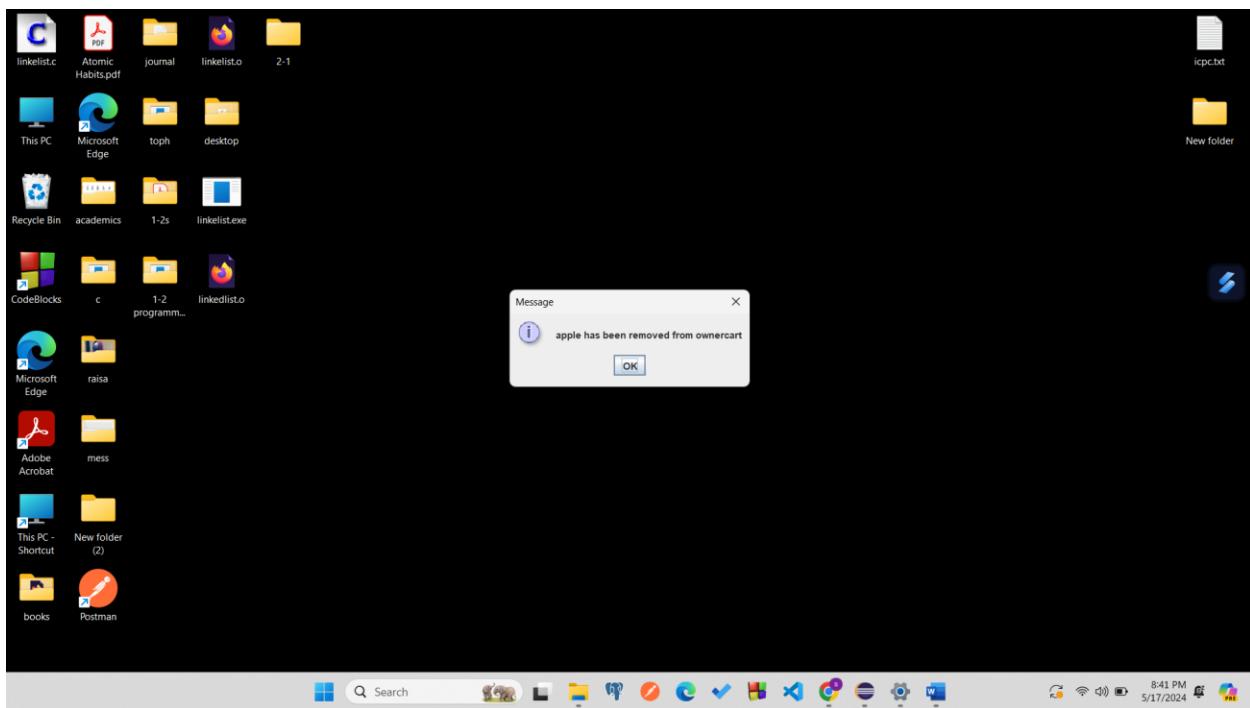




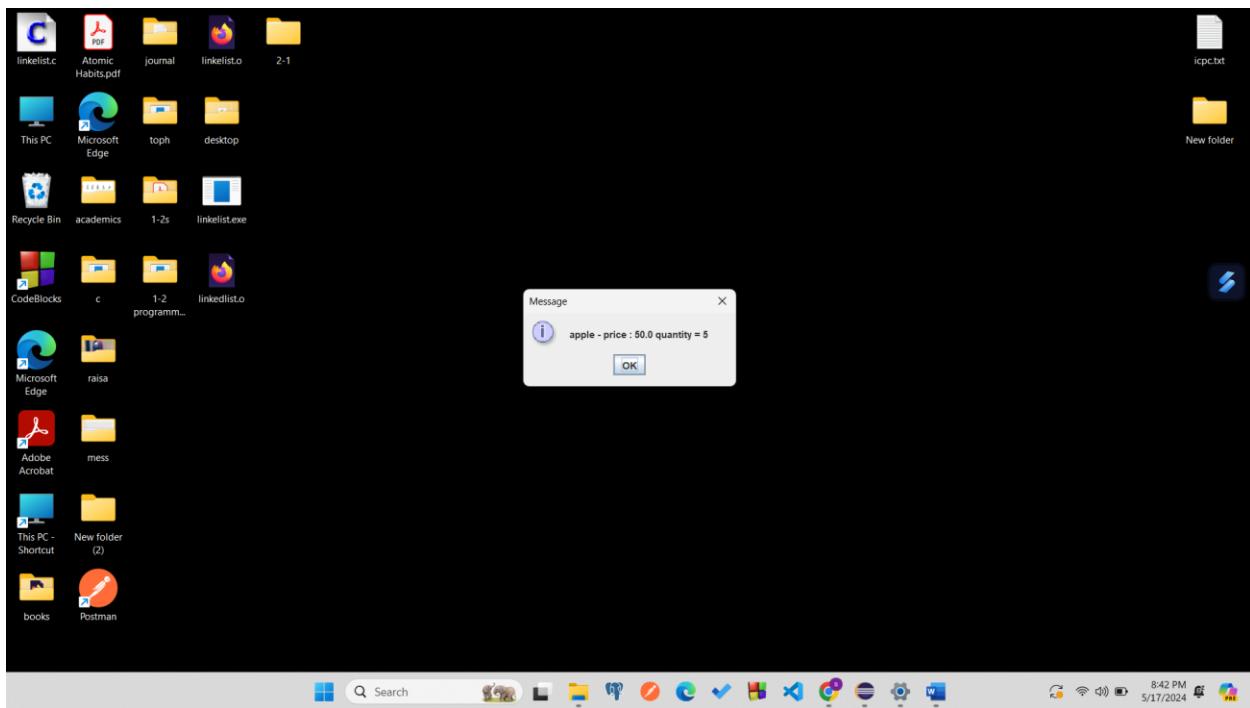
Remove single fruit :



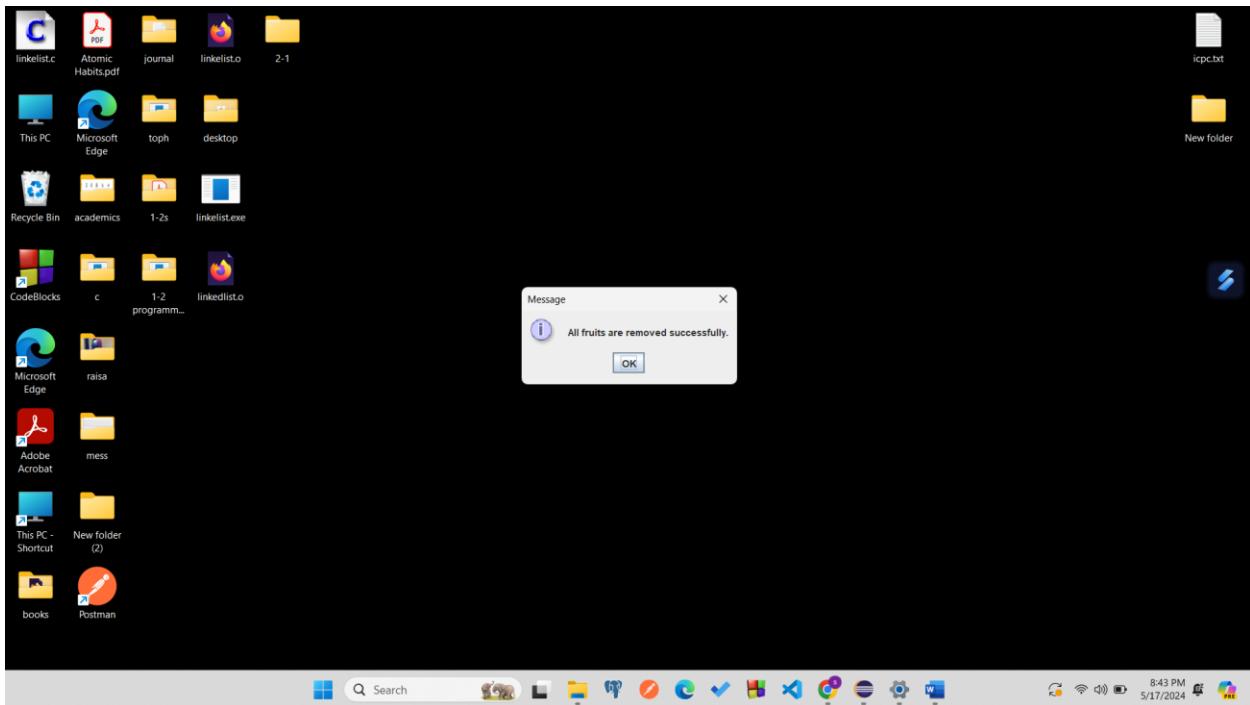
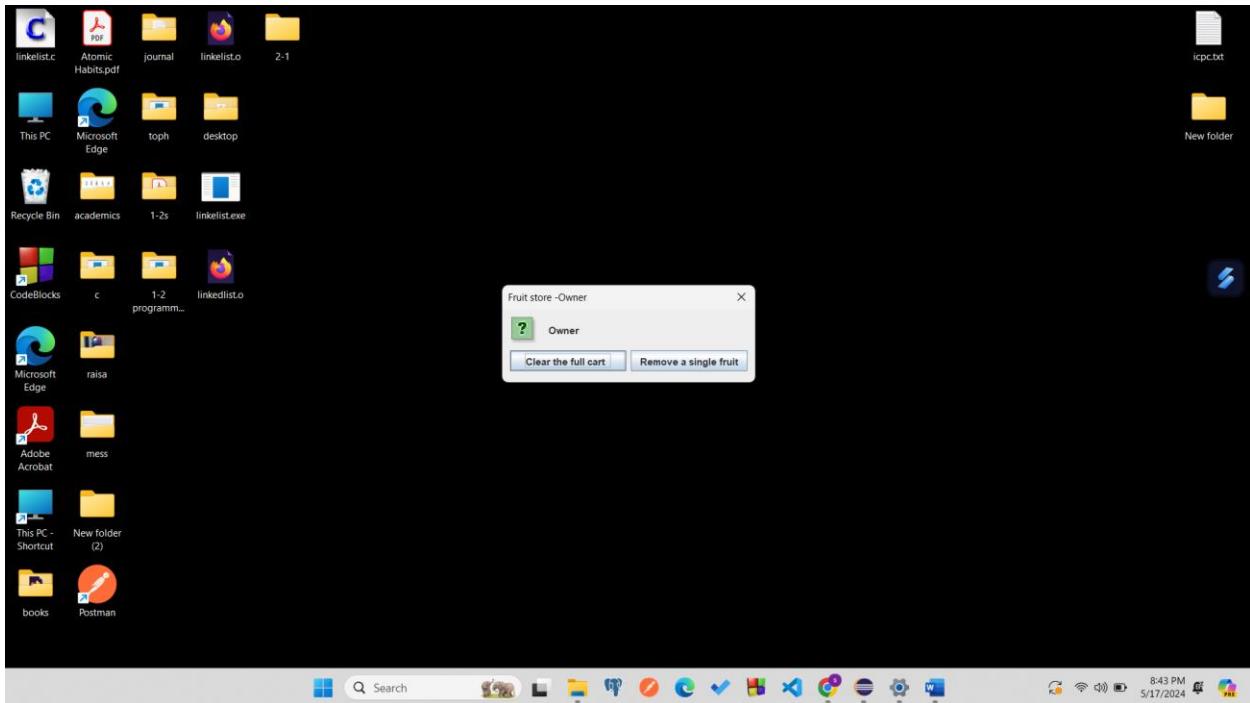




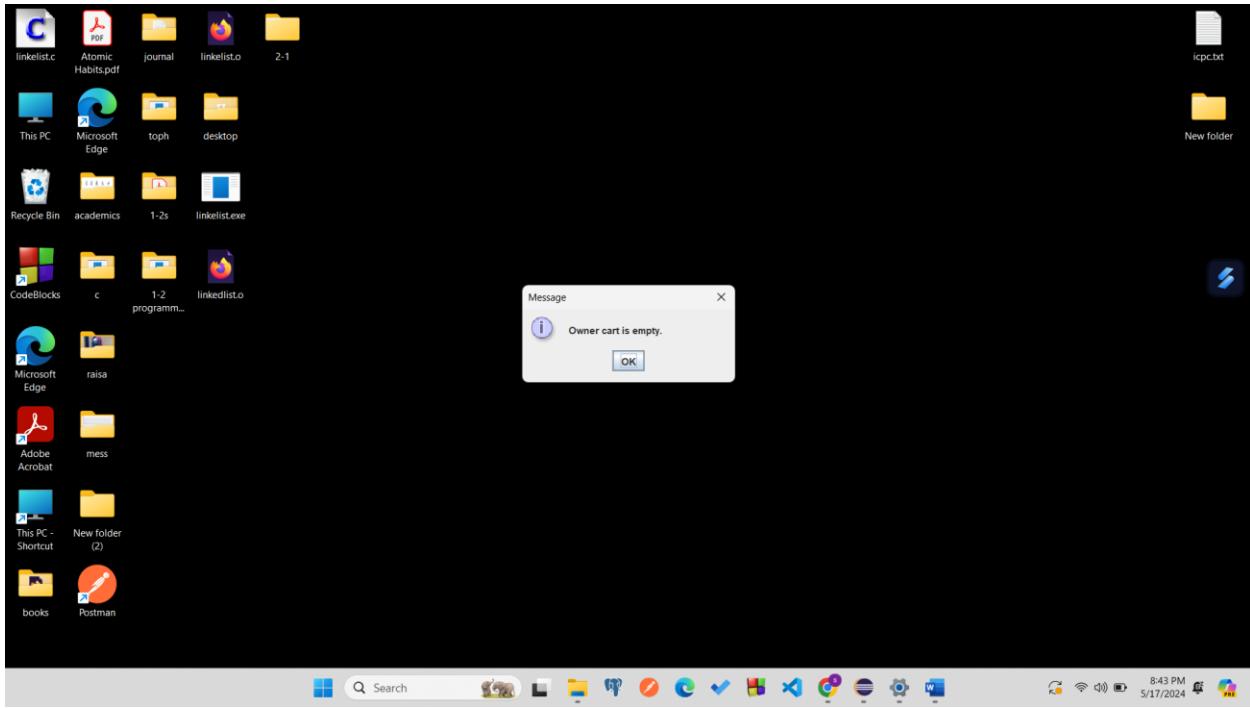
After removing, cart details:



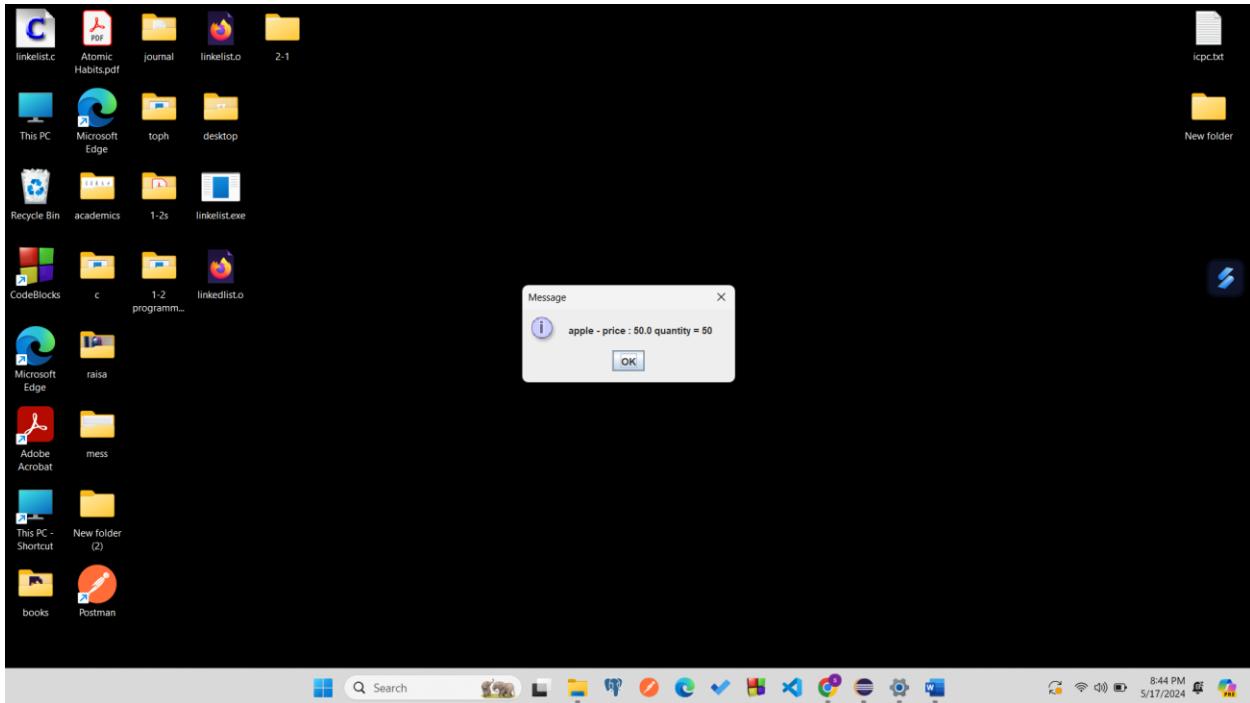
Clearing the whole cart:

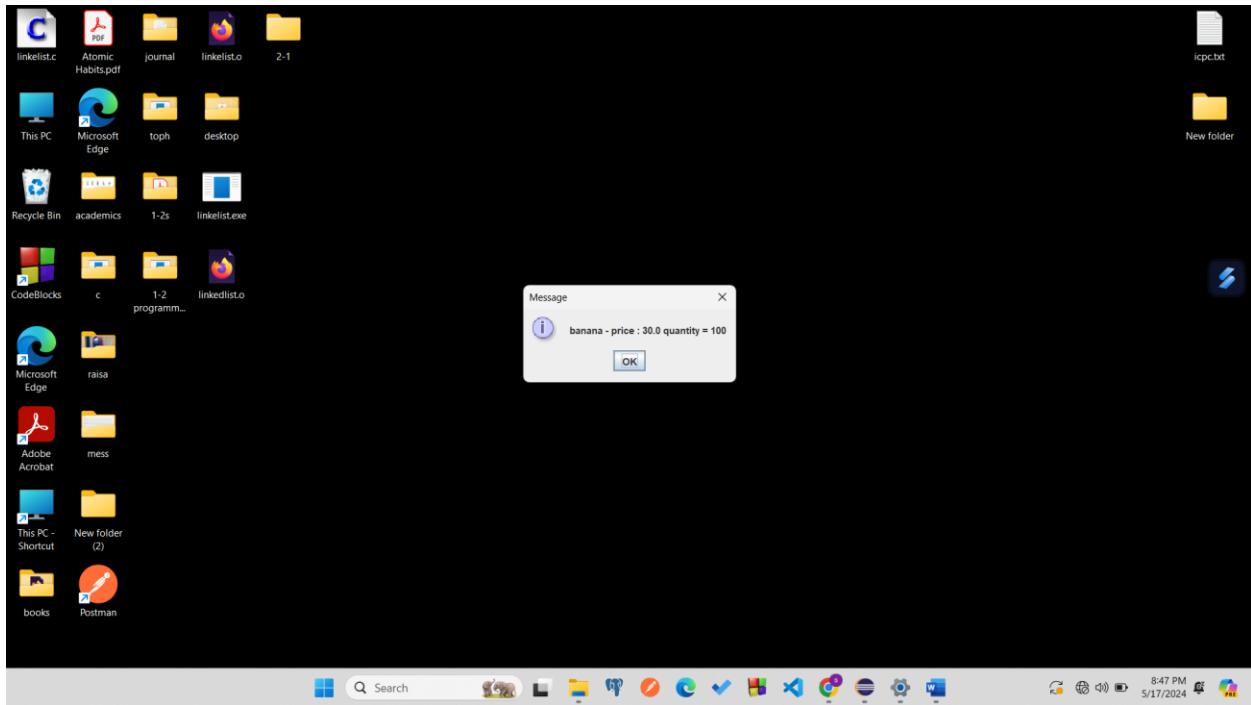


After Clearing the whole cart, cart items:

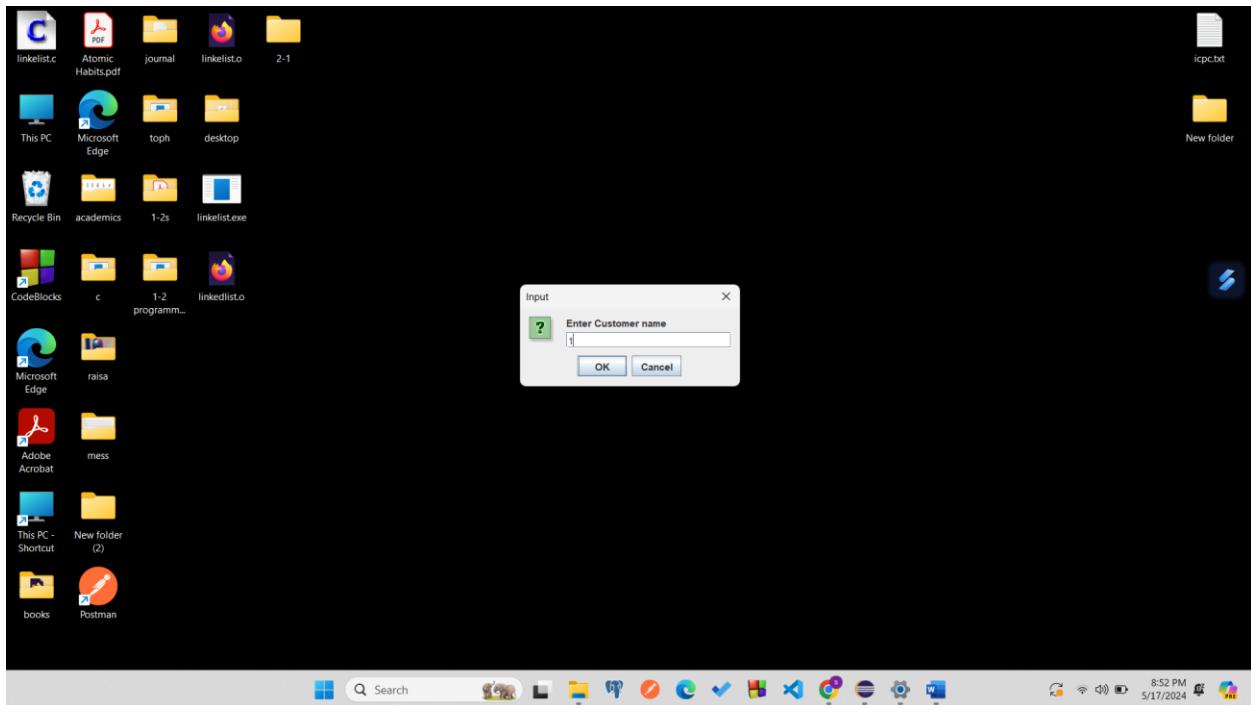


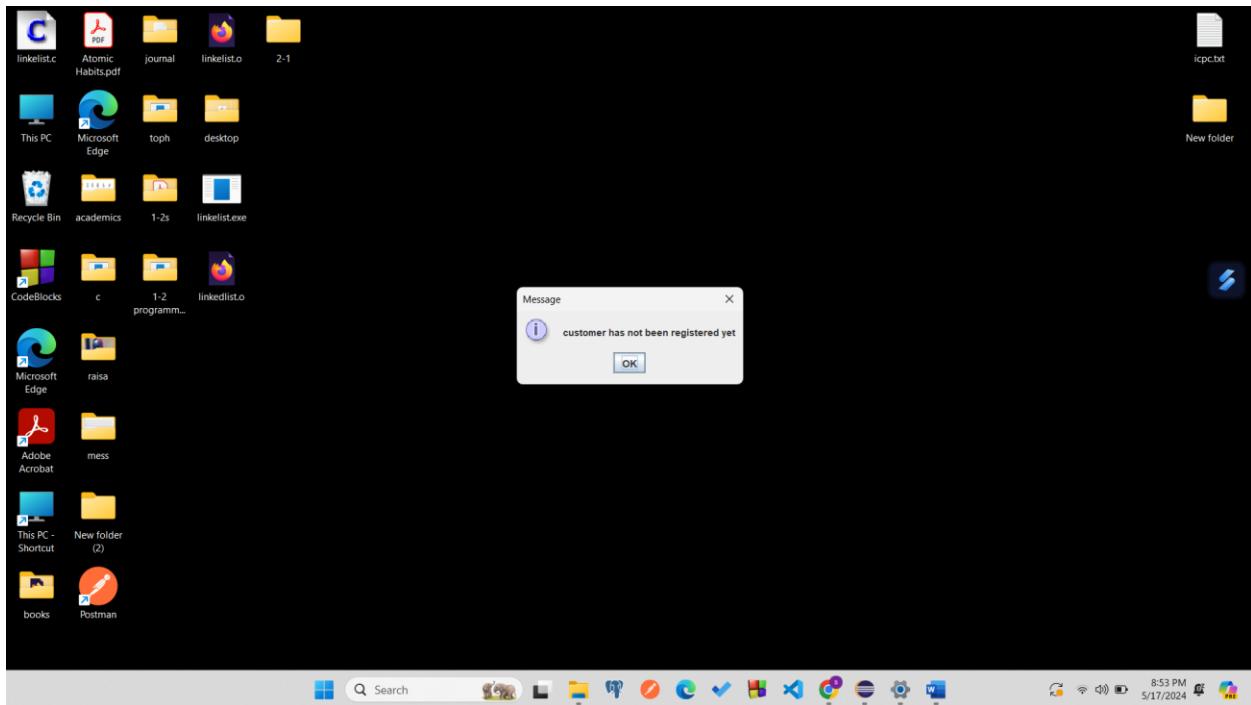
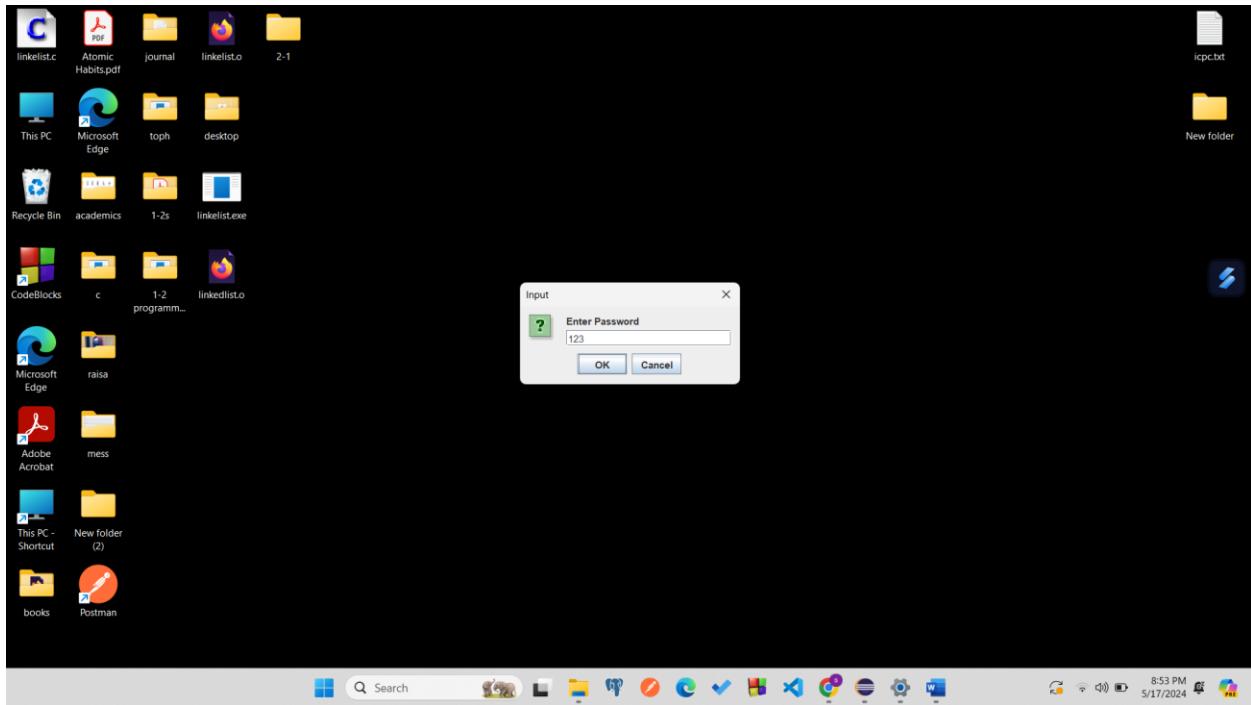
Cart item details (after adding fruits for customer to add):

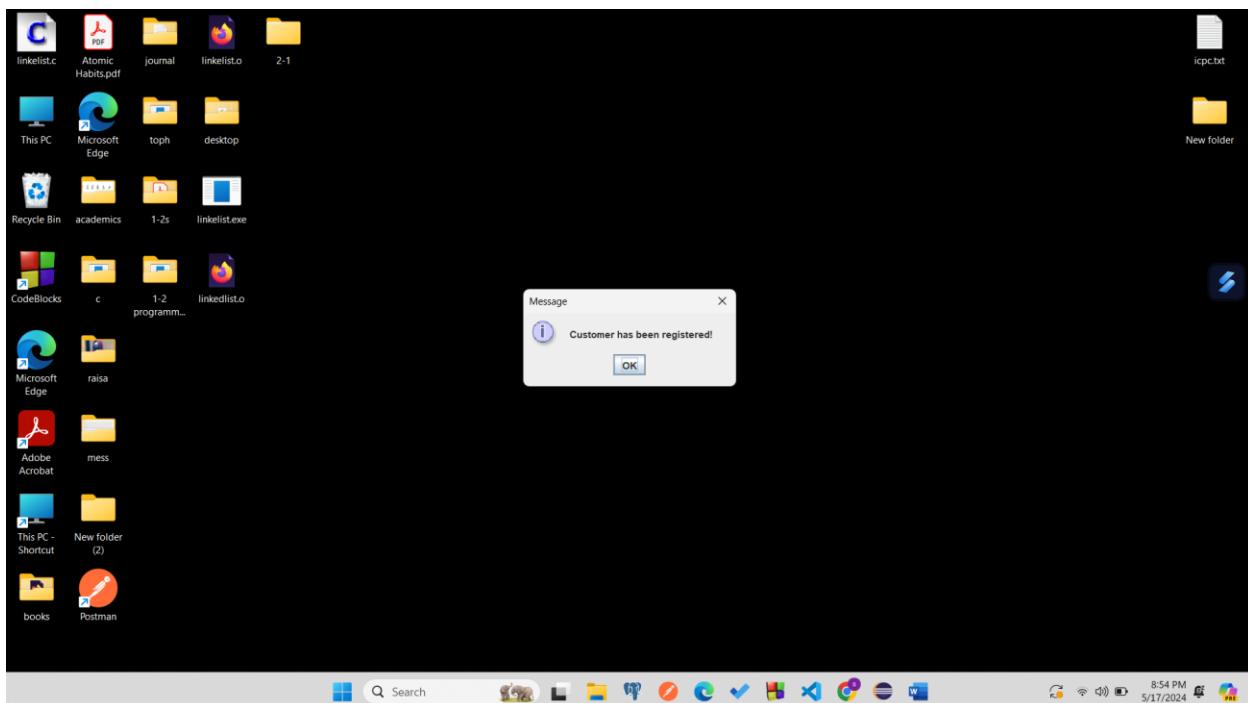
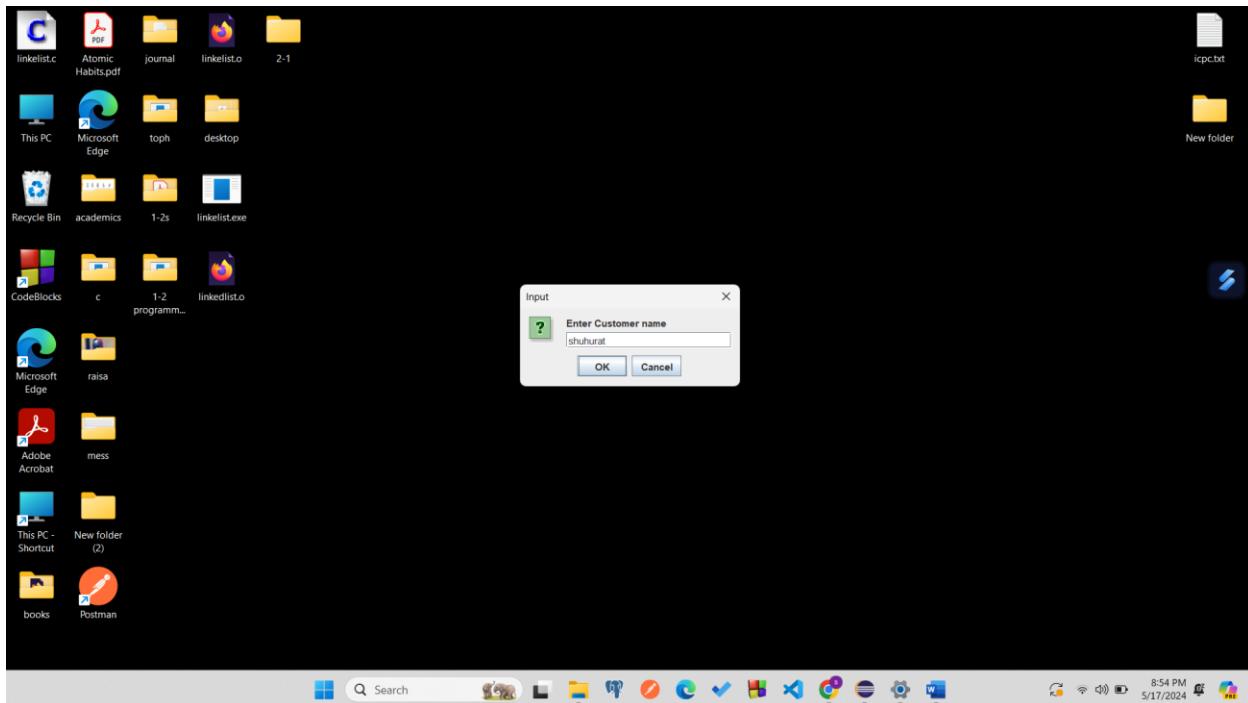




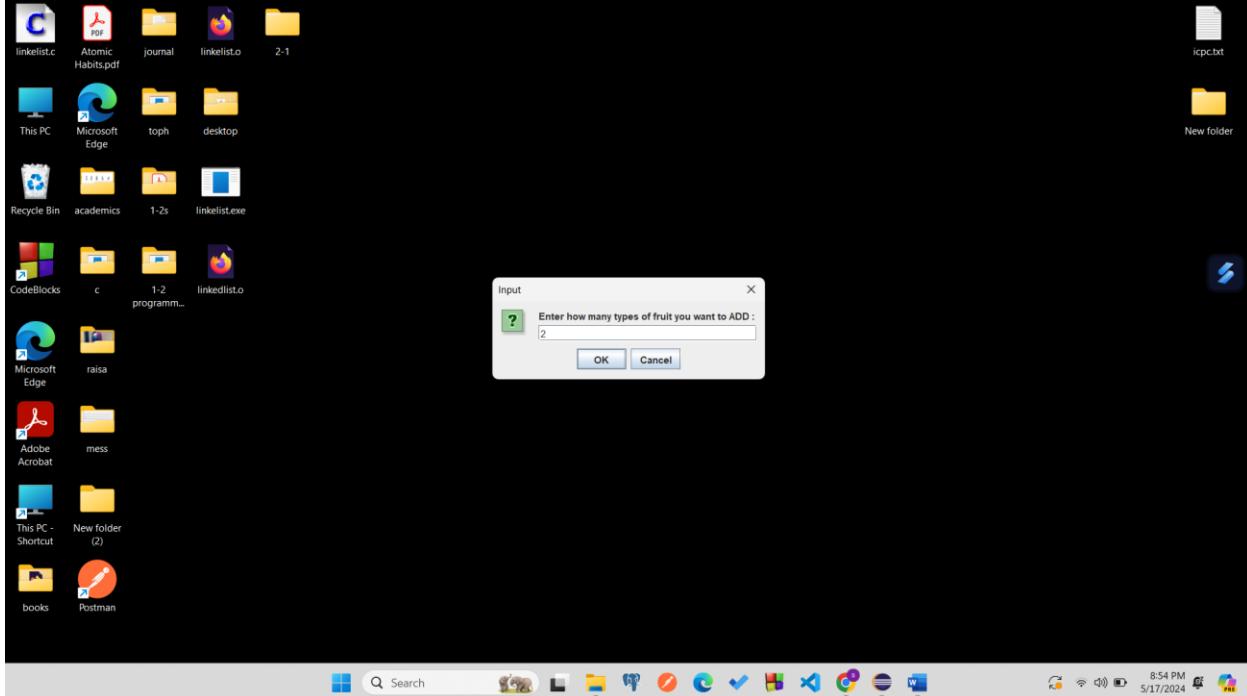
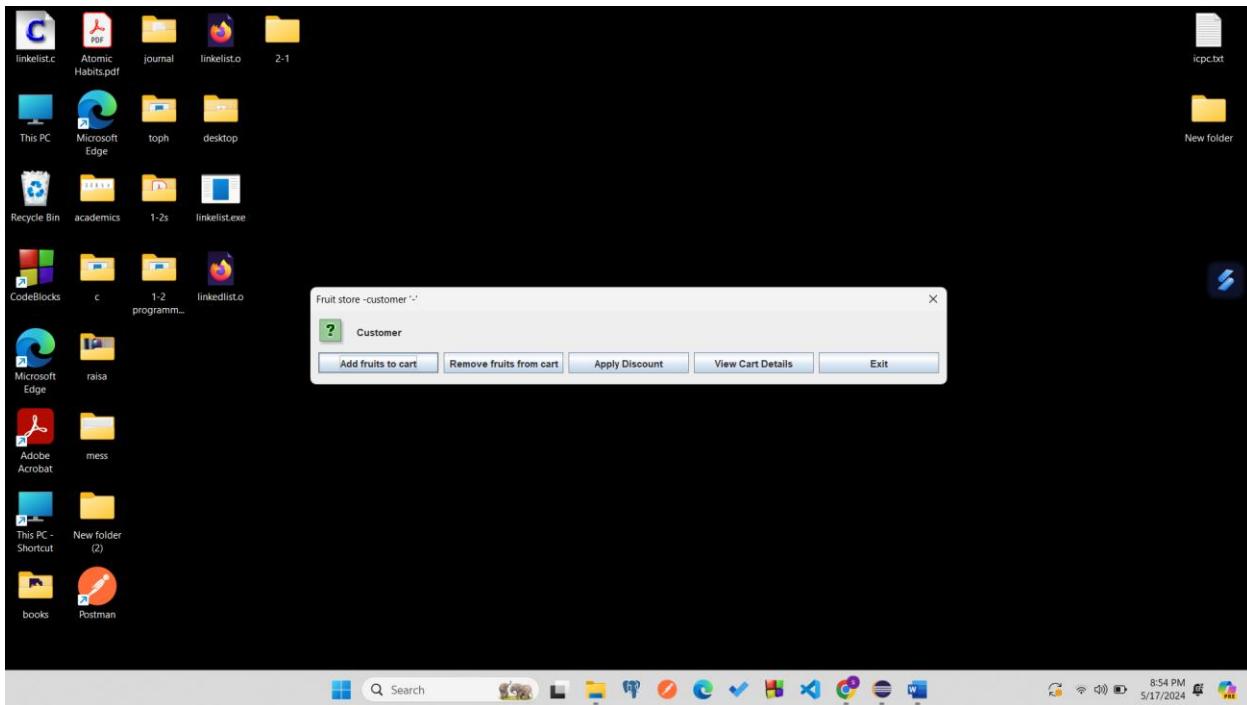
Customer login:

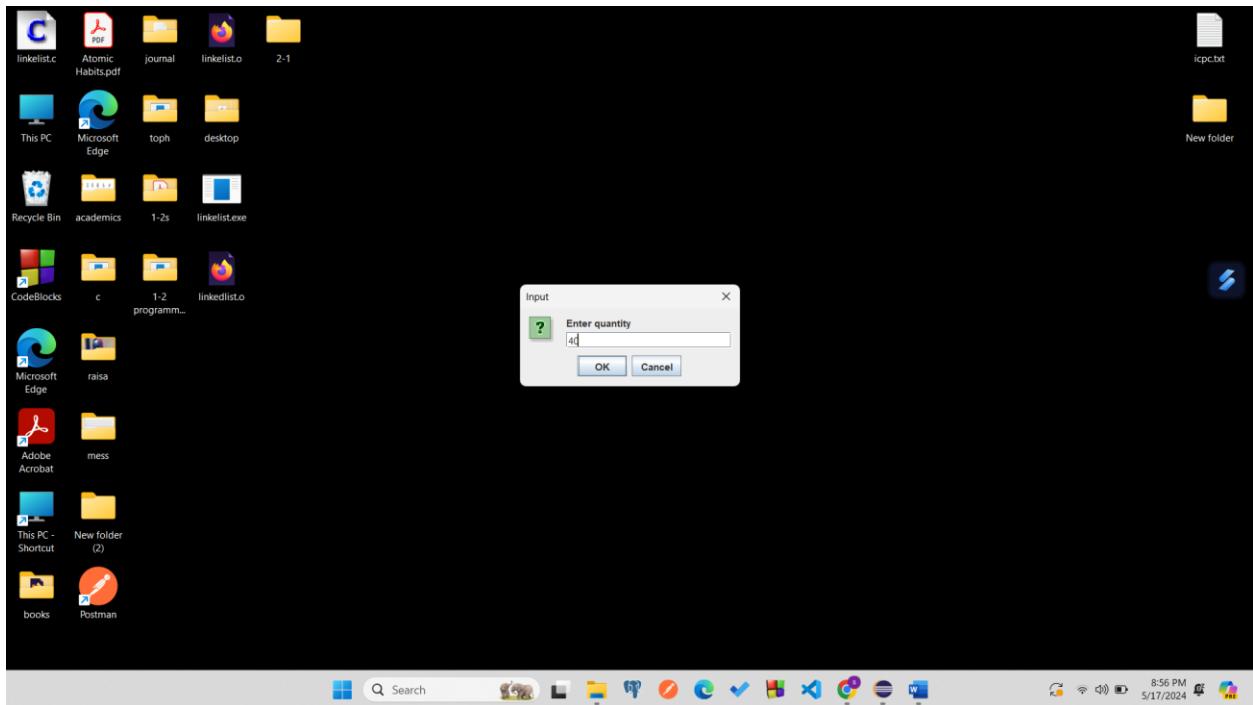
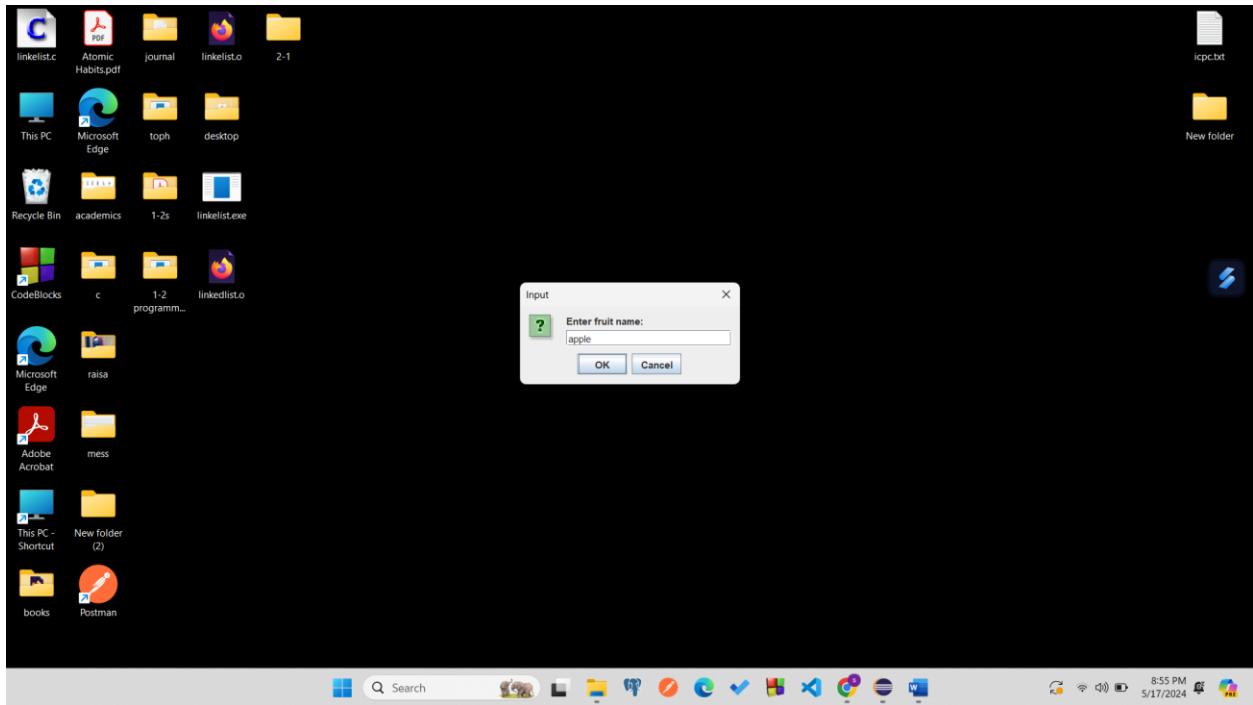


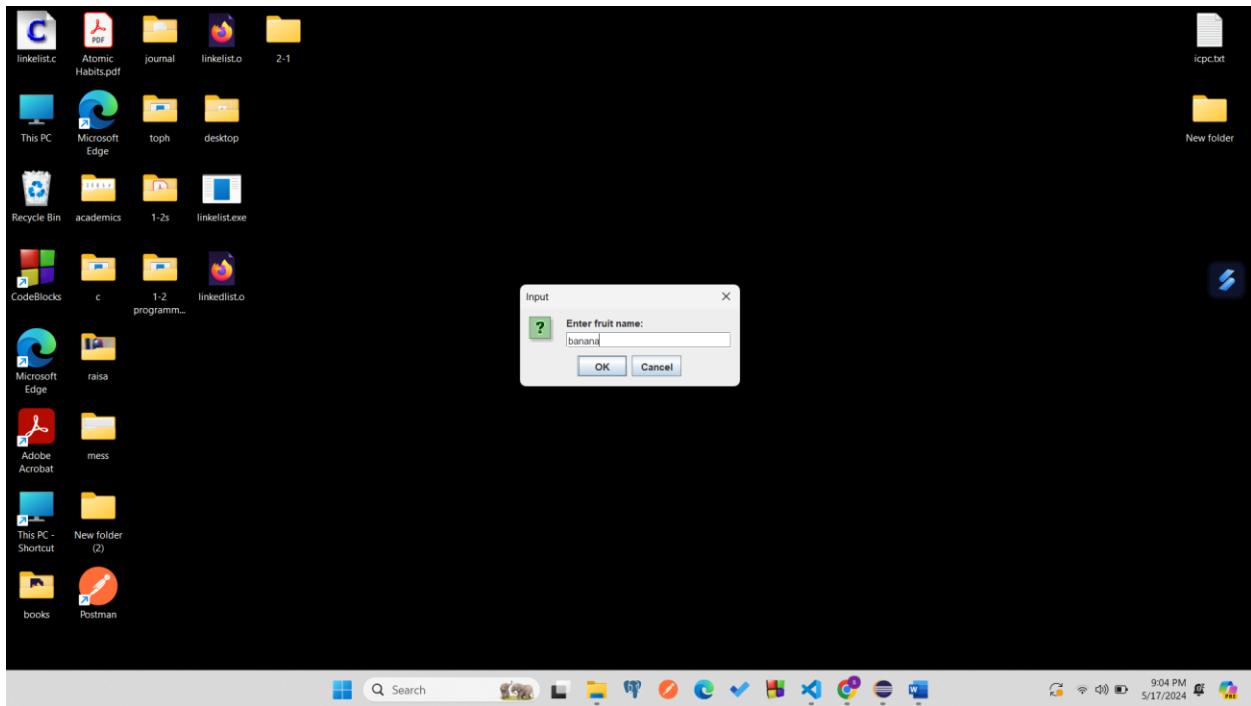
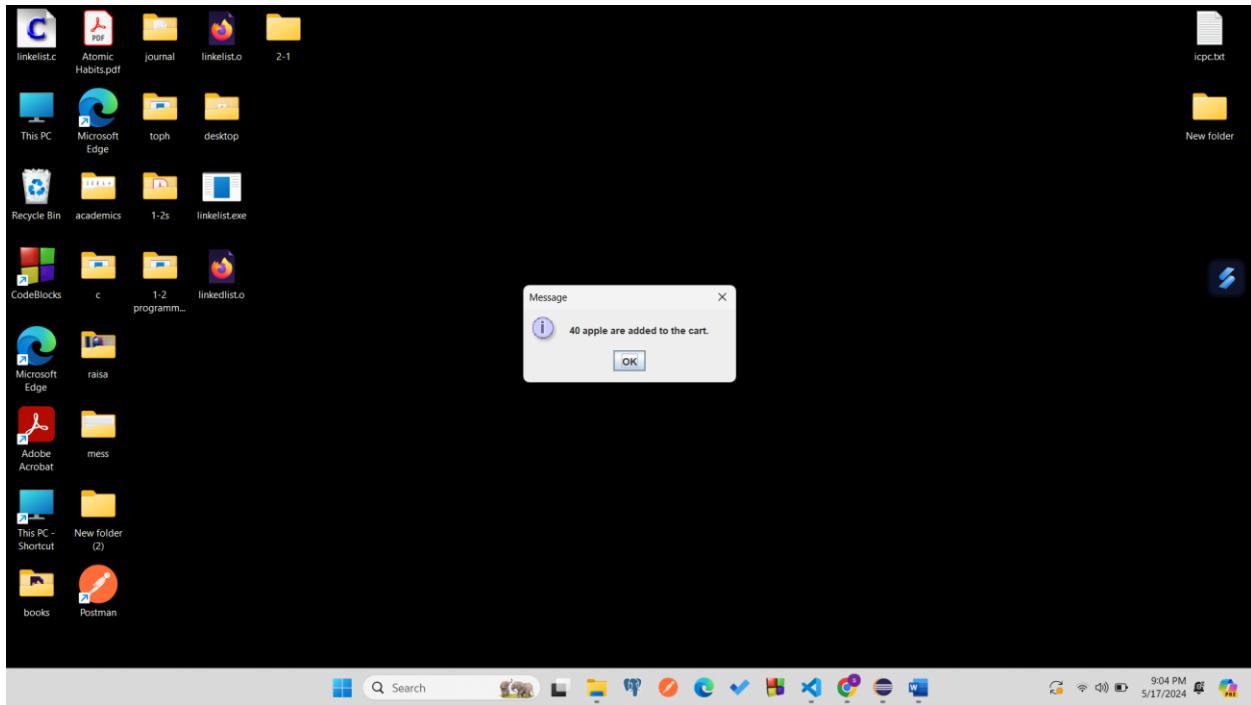


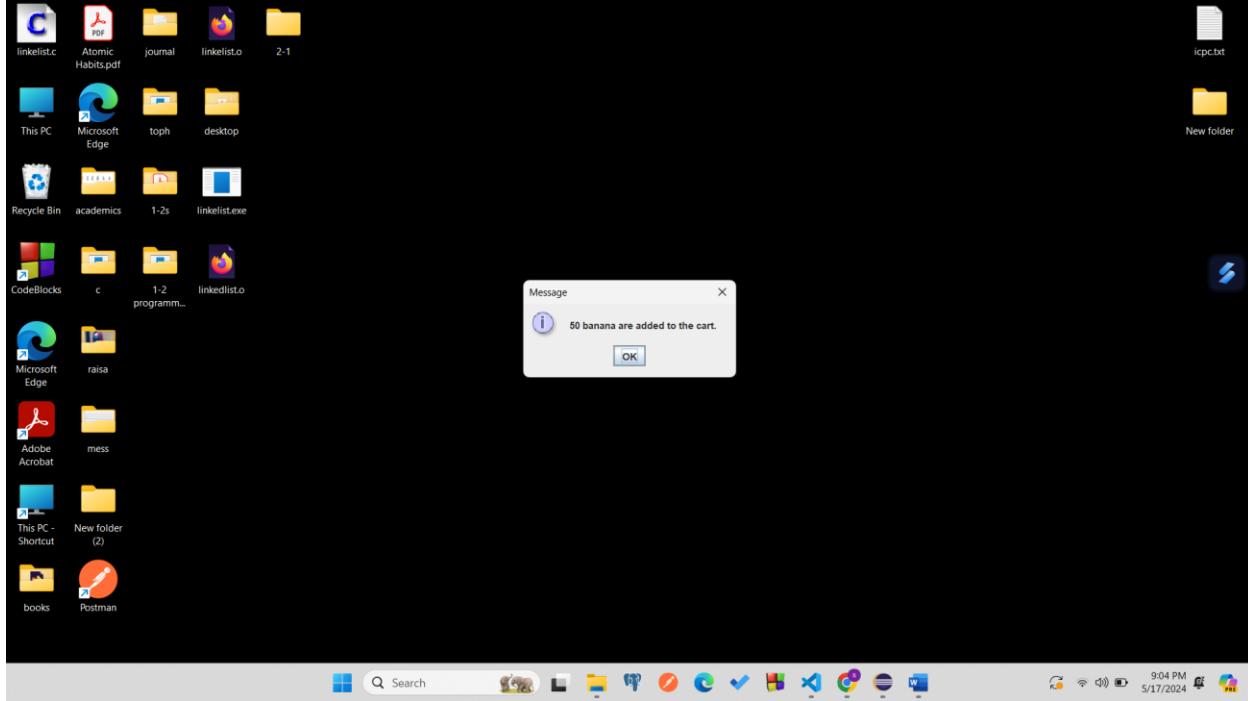
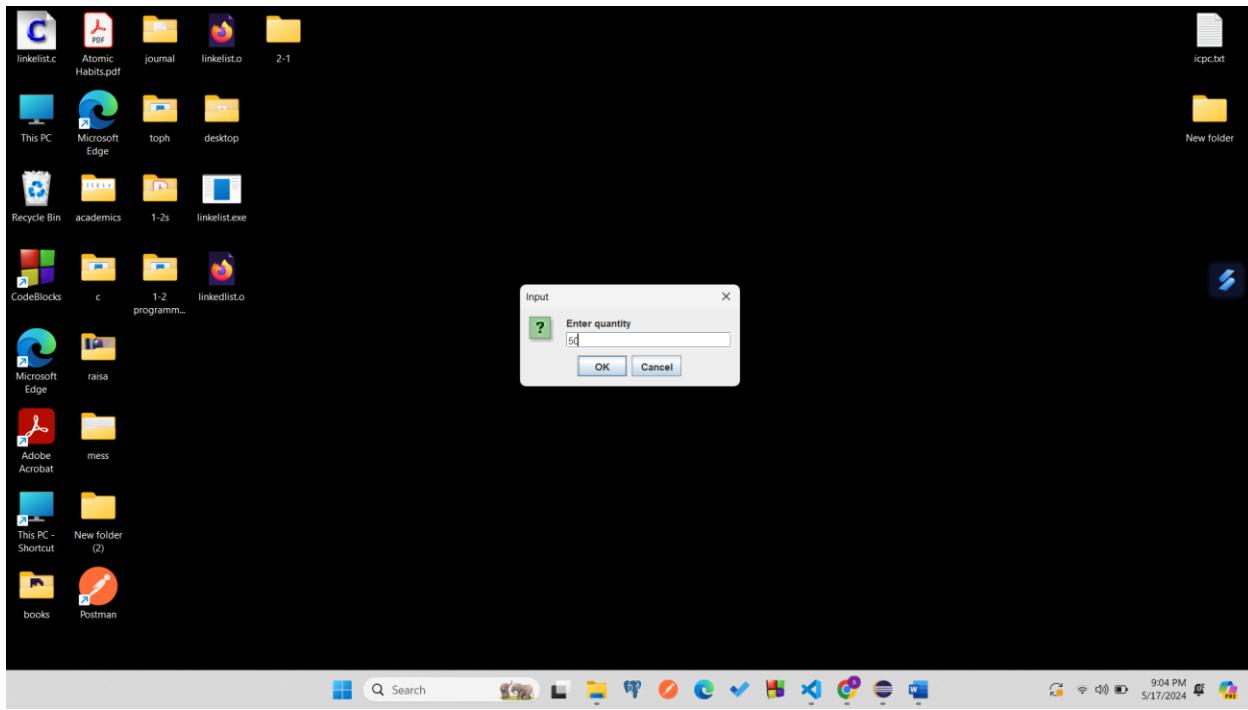


Adding fruits to cart:

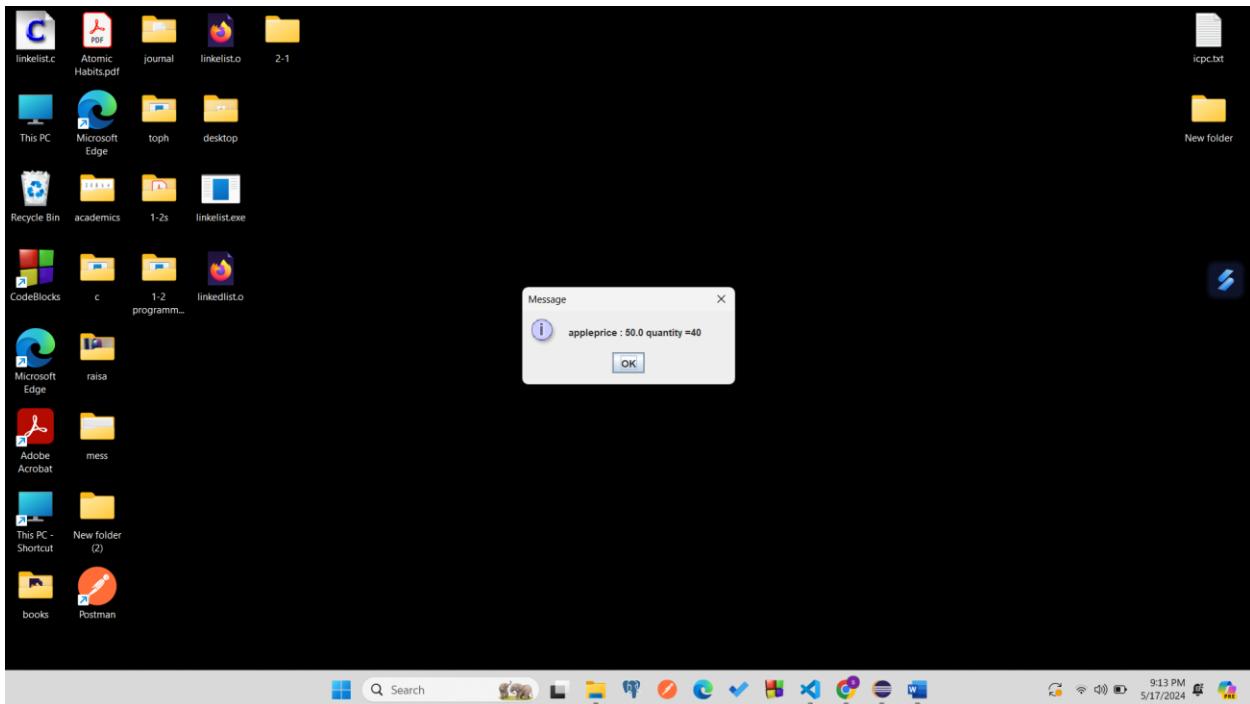
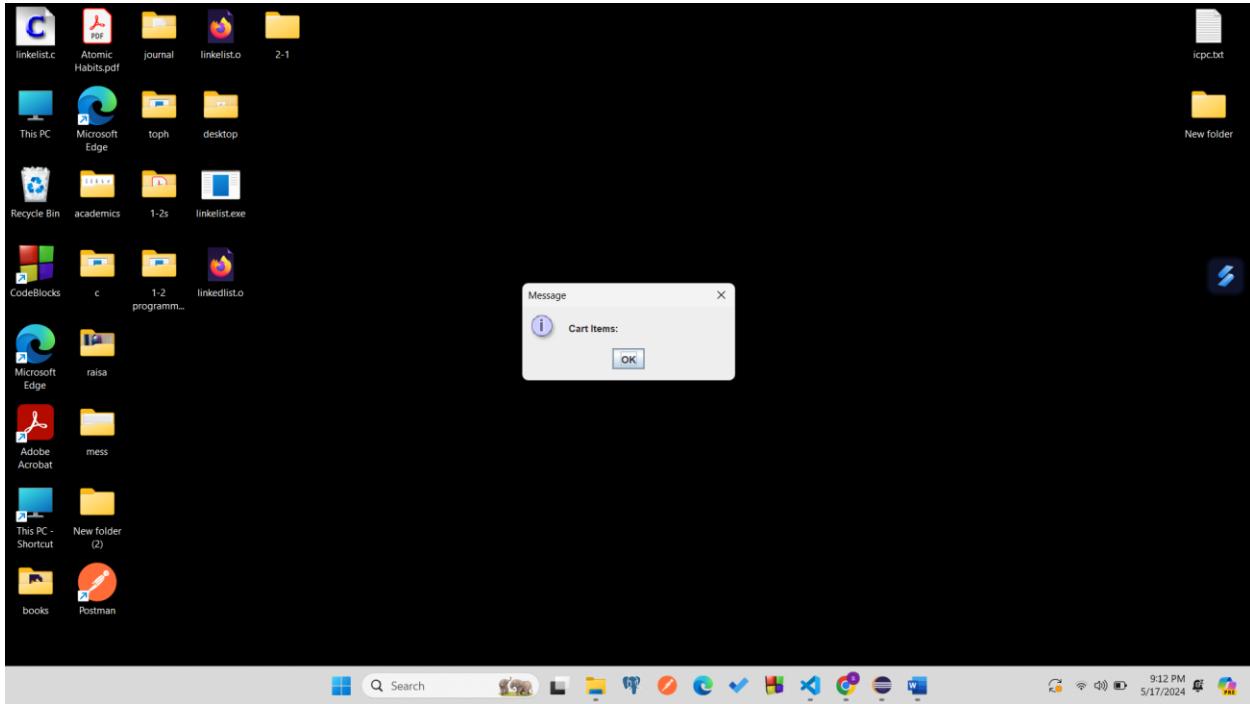


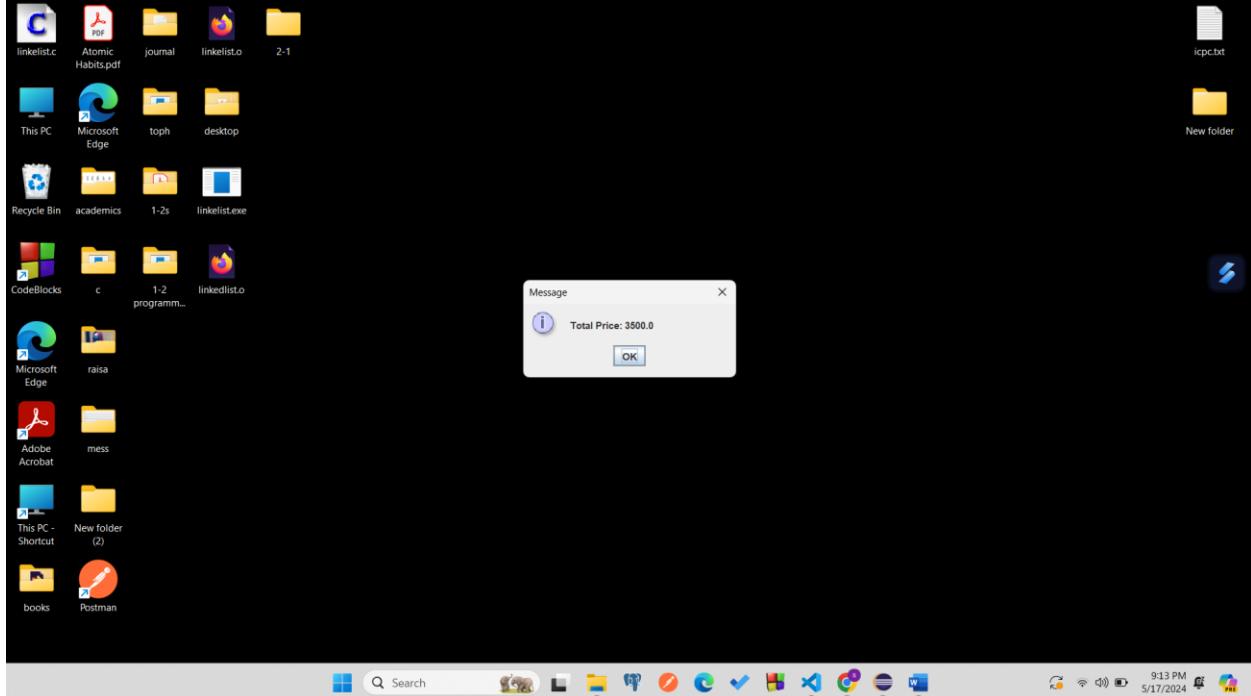
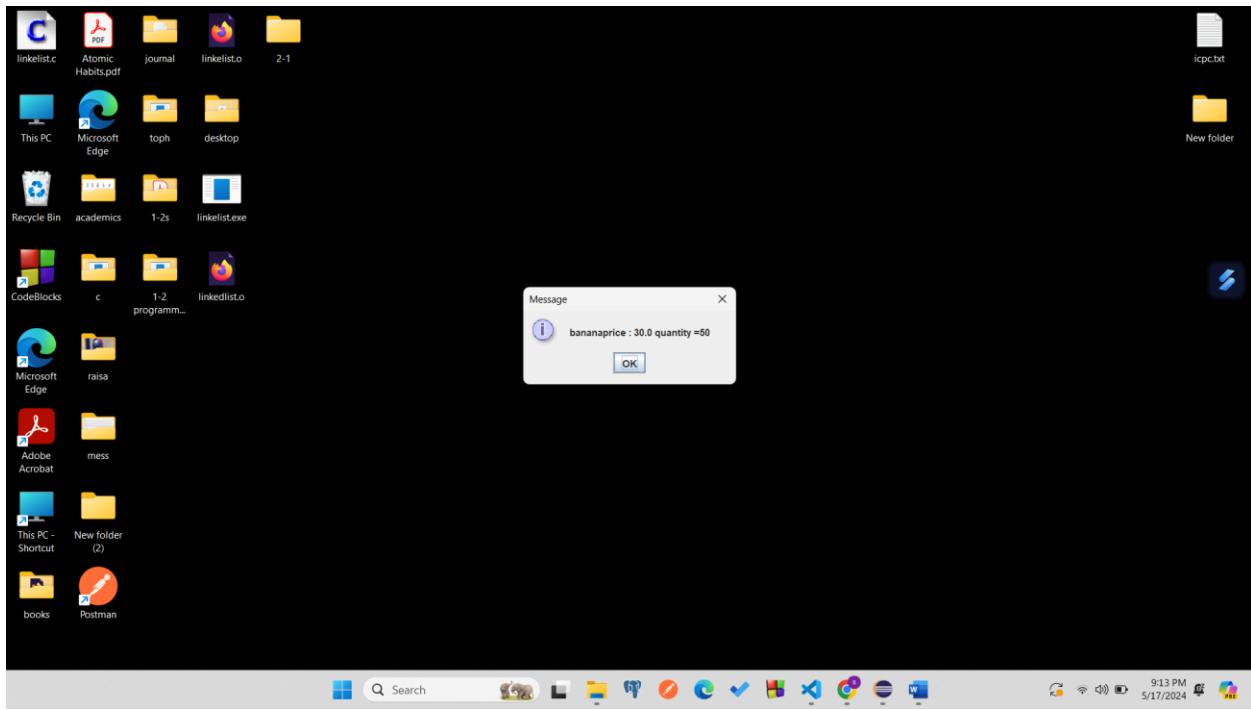




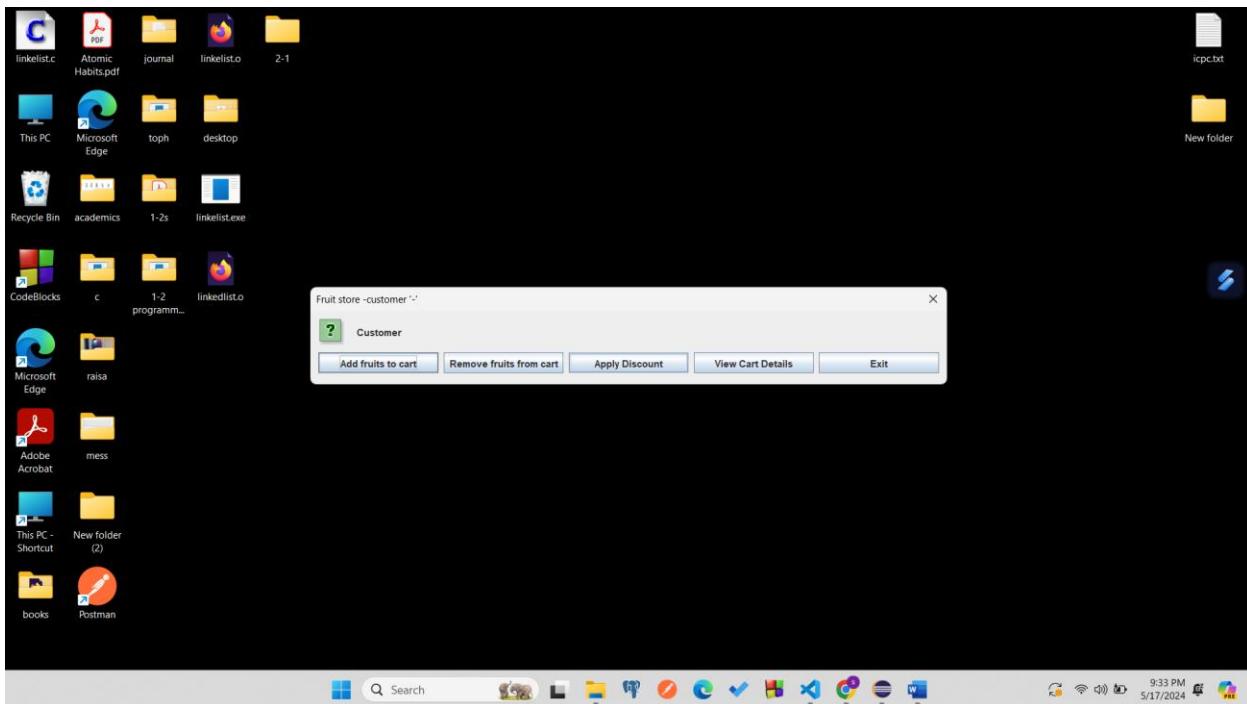


After adding cart item details:

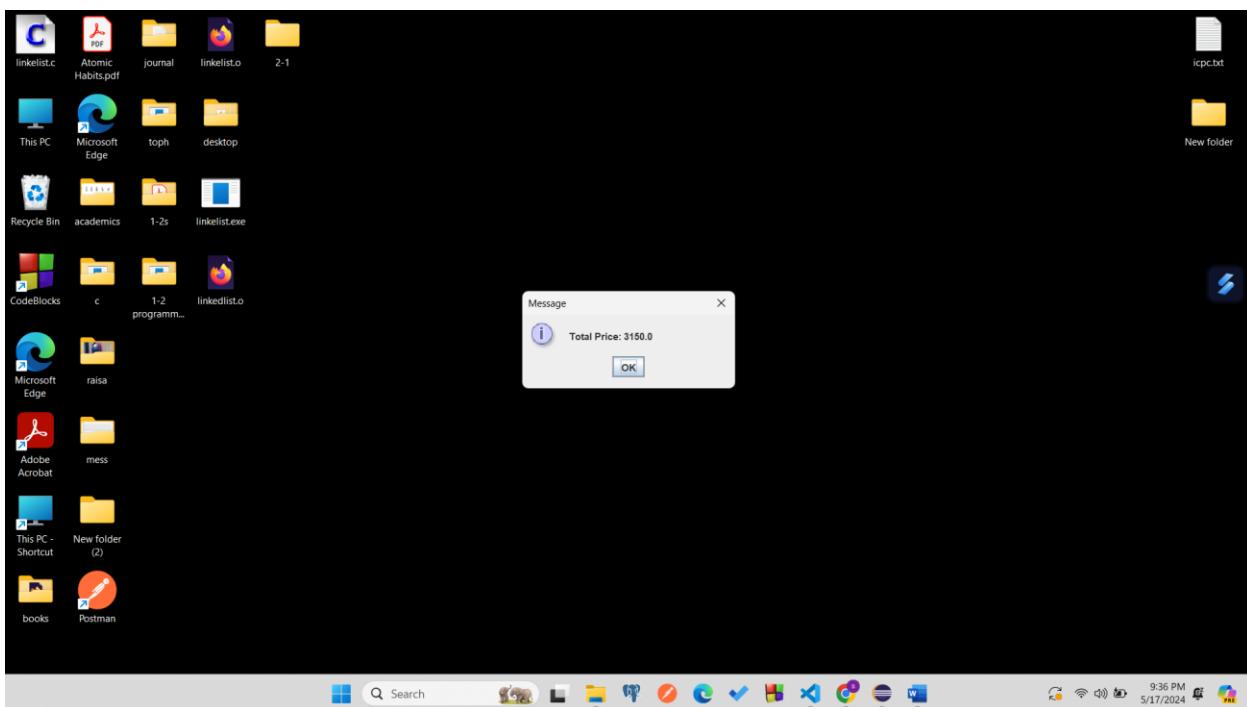




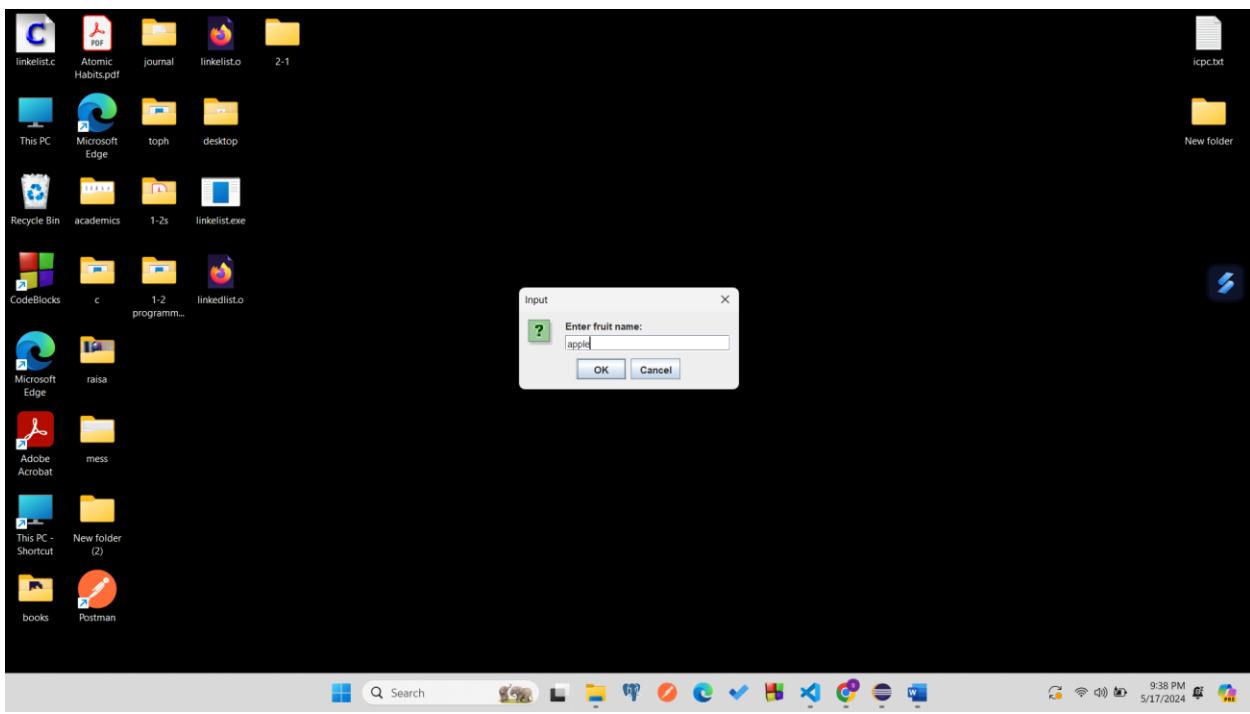
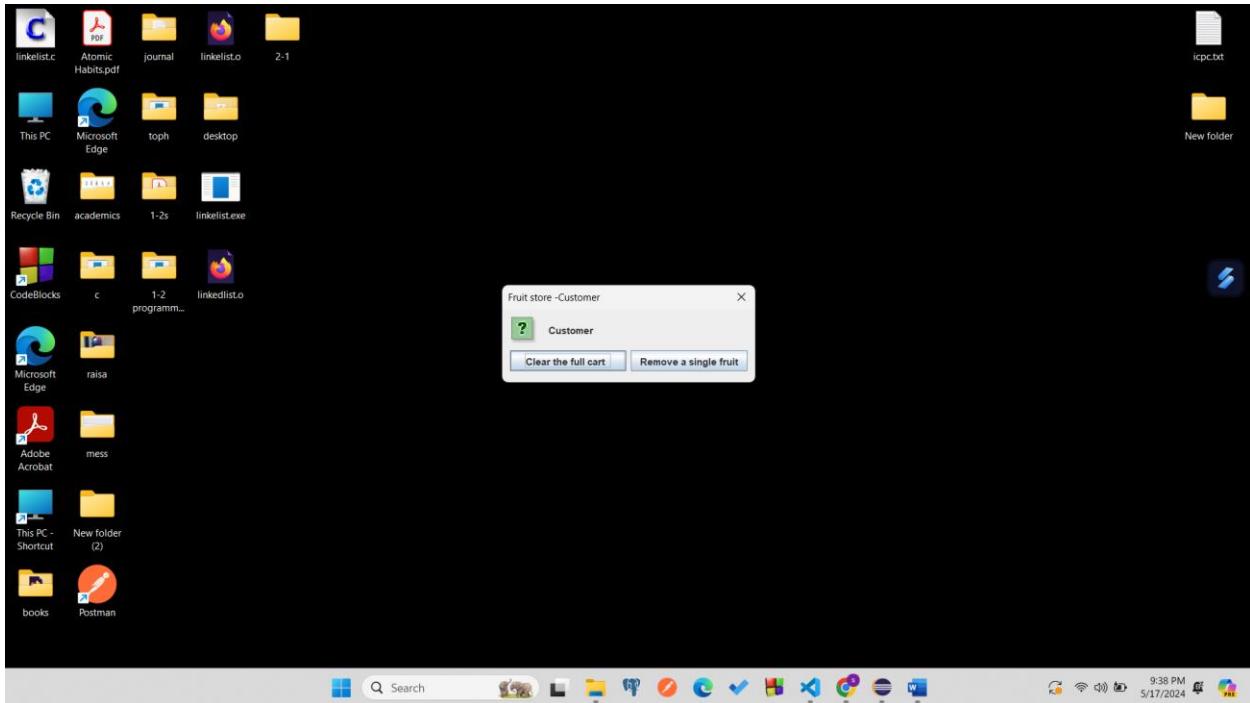
Applying discount:

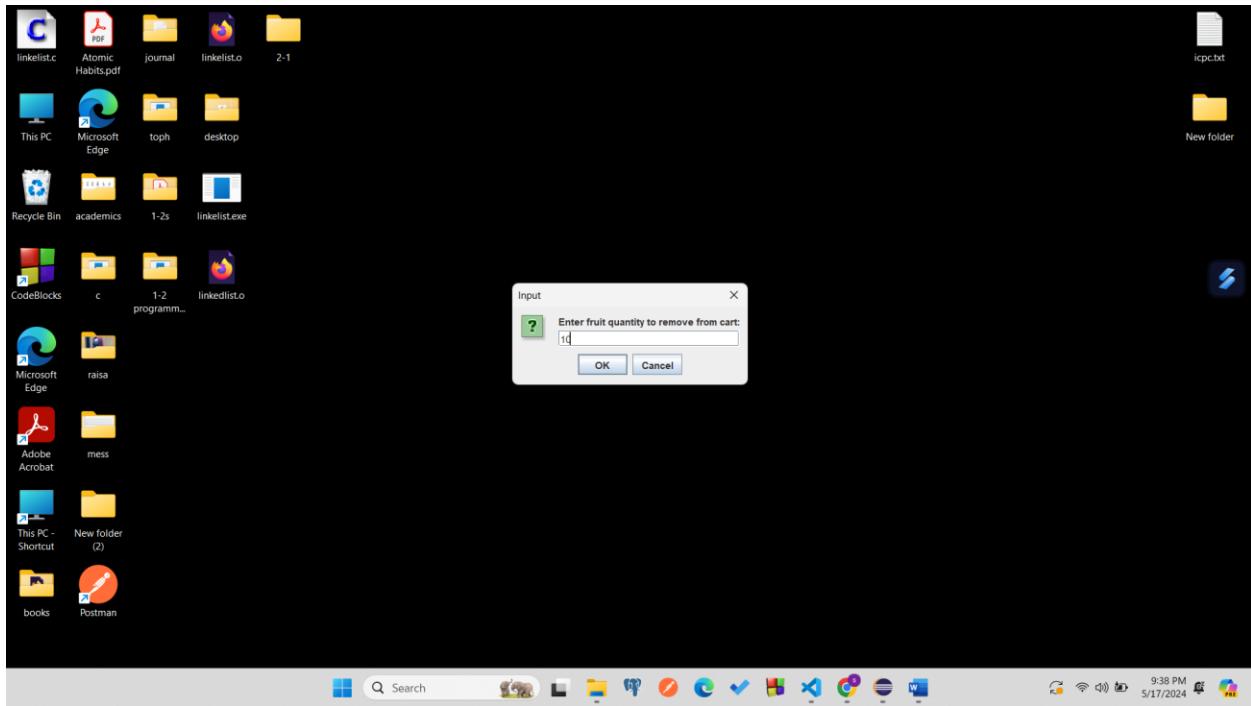


After applying discount price:

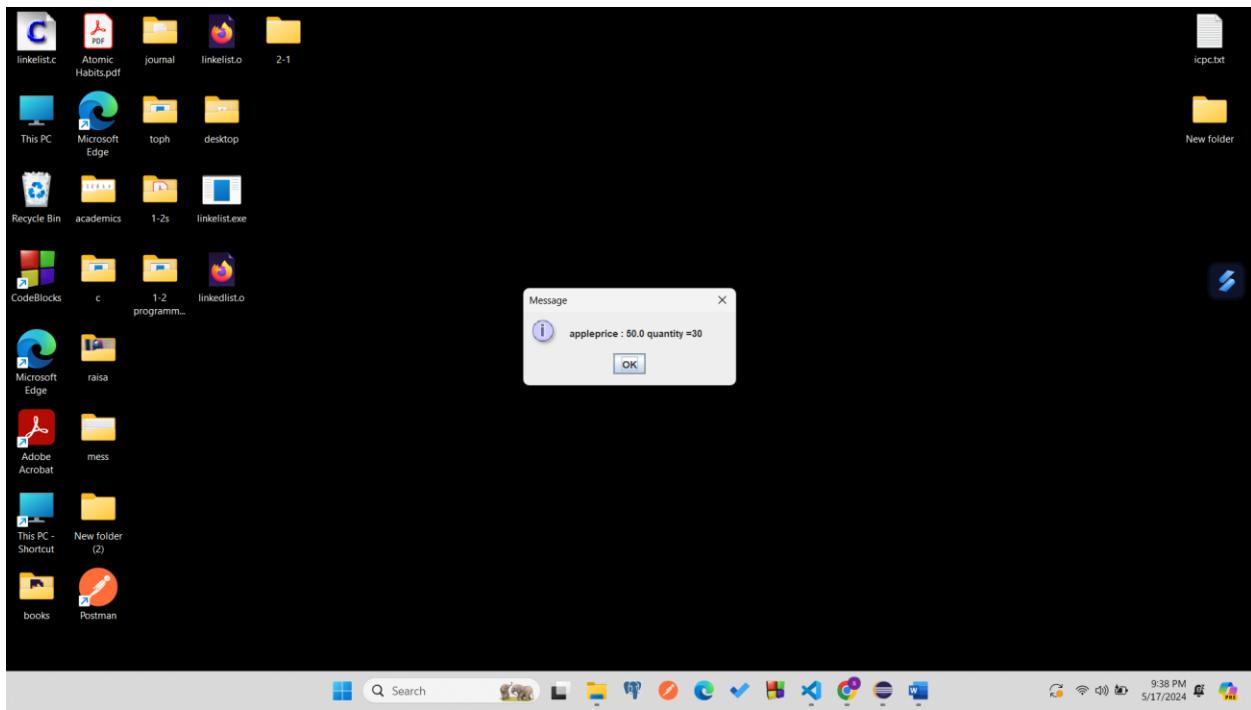


Removing fruit from cart:

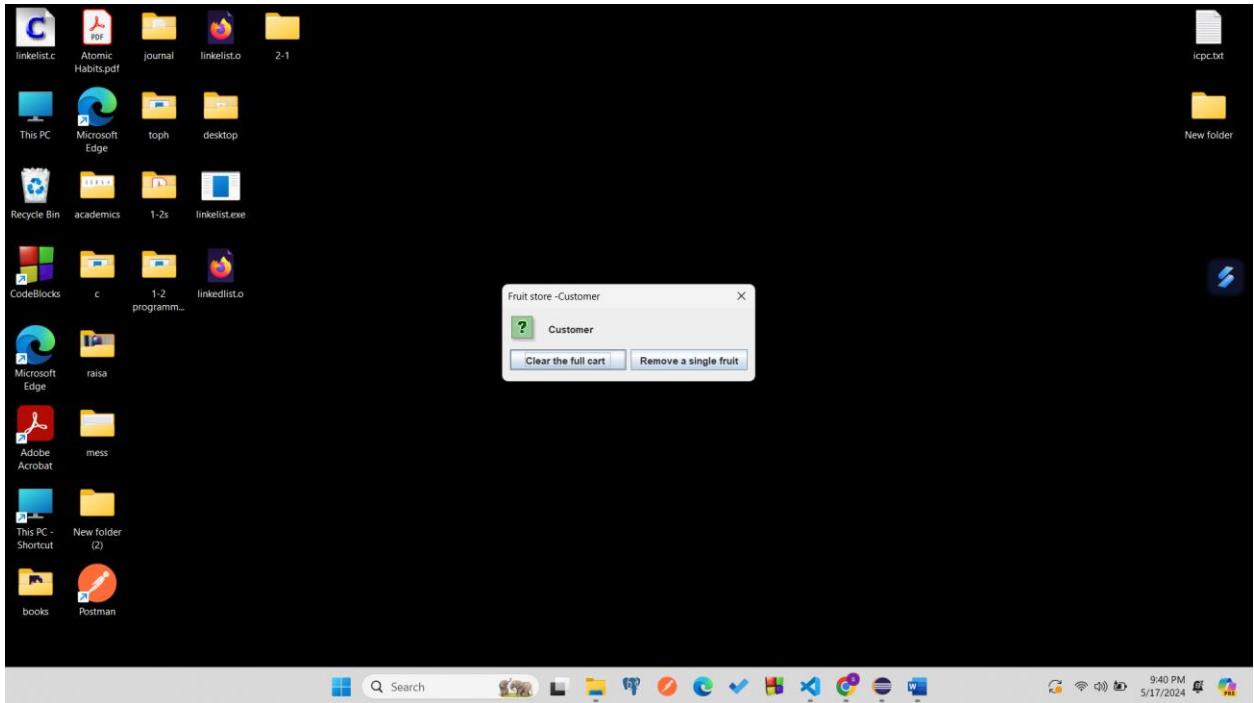




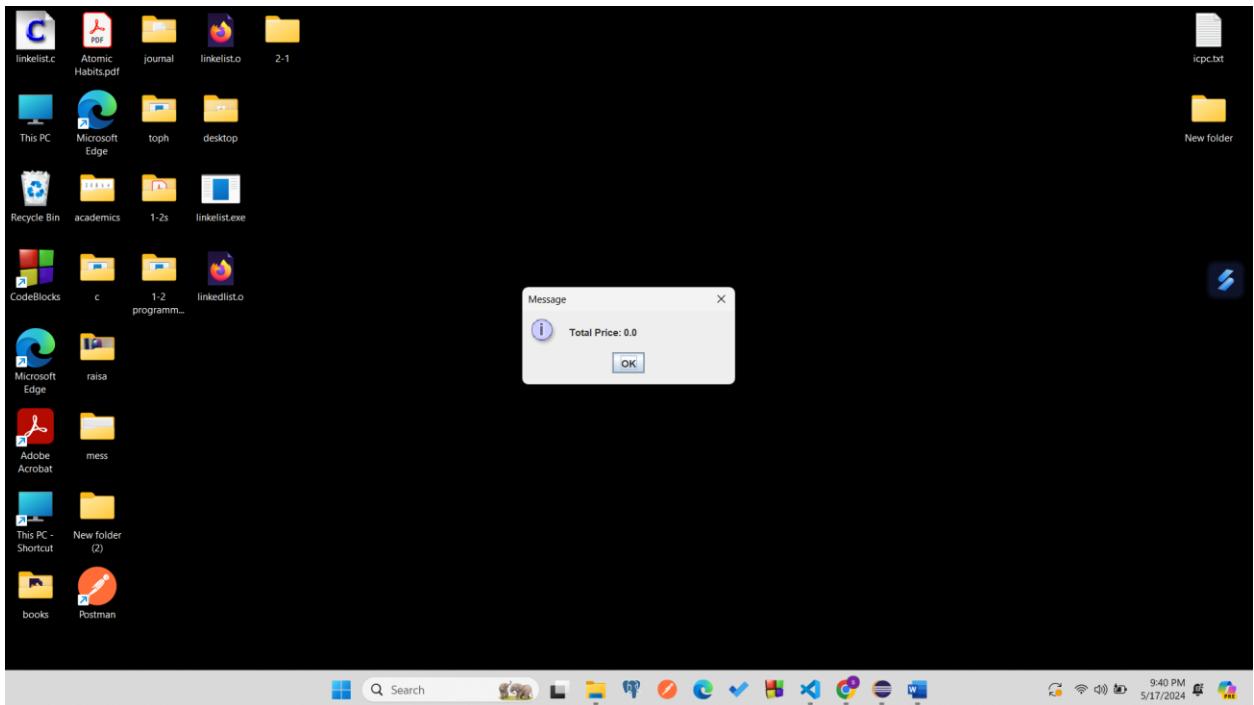
After removing cart details:



Clearing the whole cart:

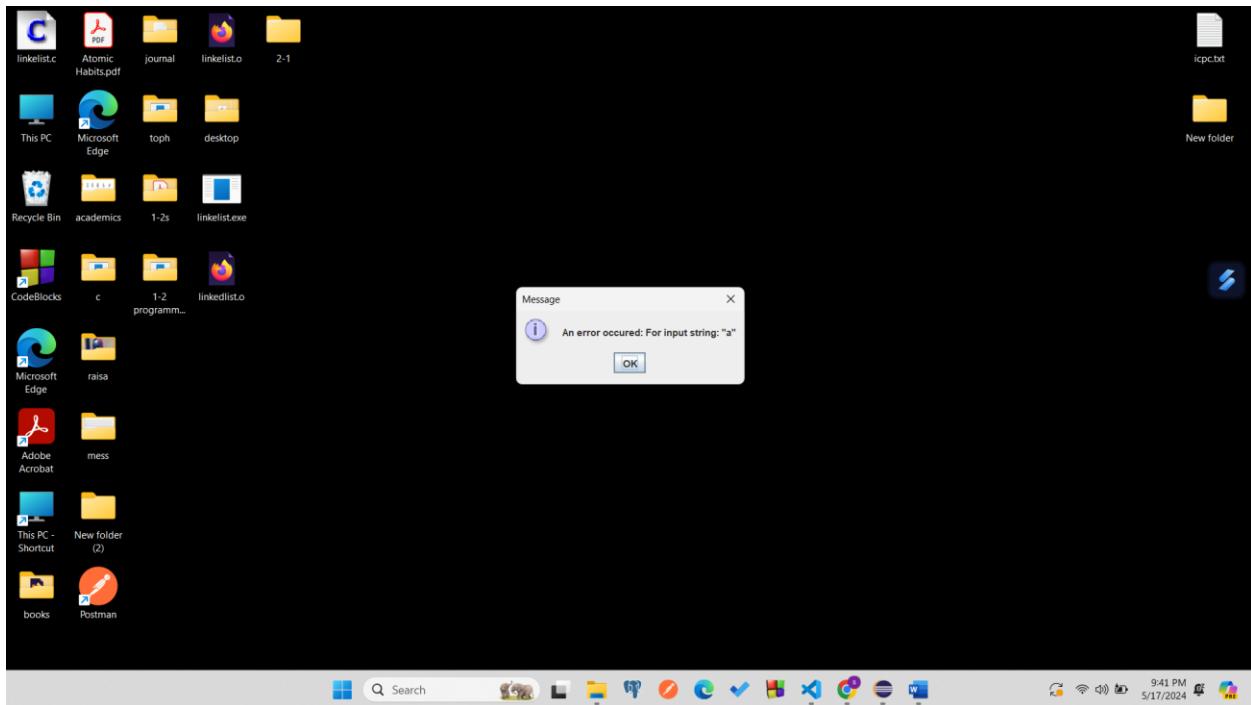
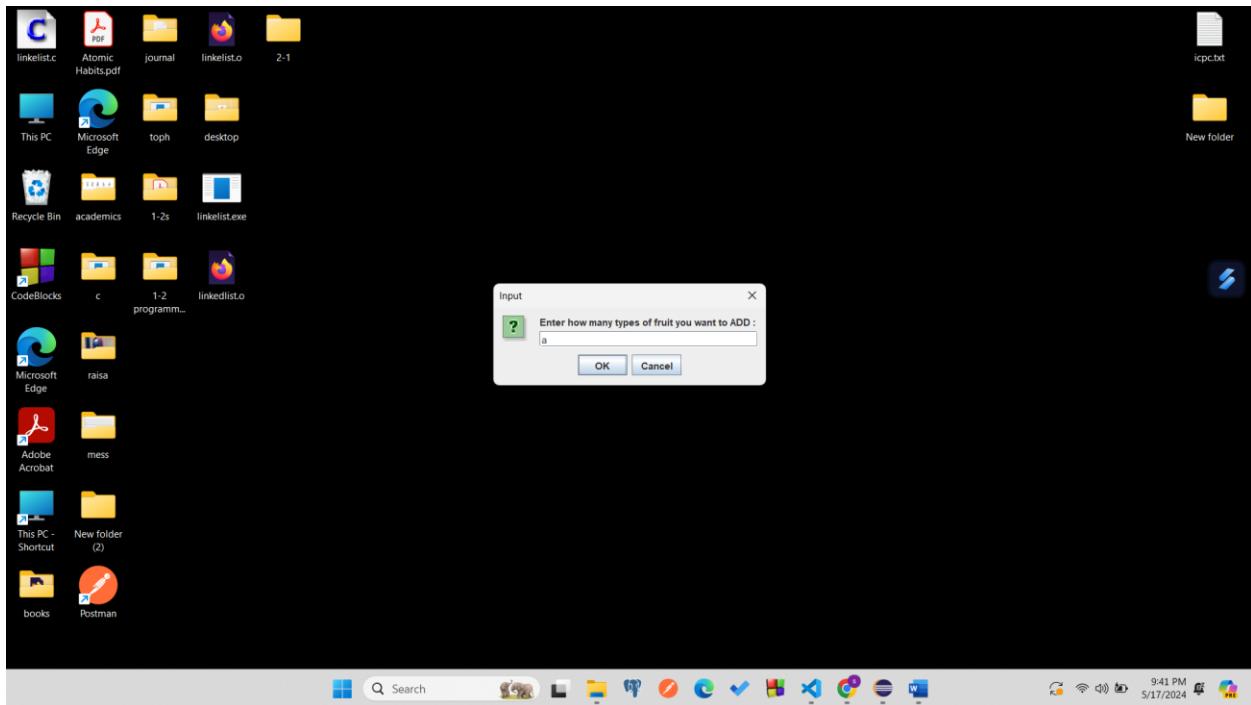


After clearing the whole cart price:

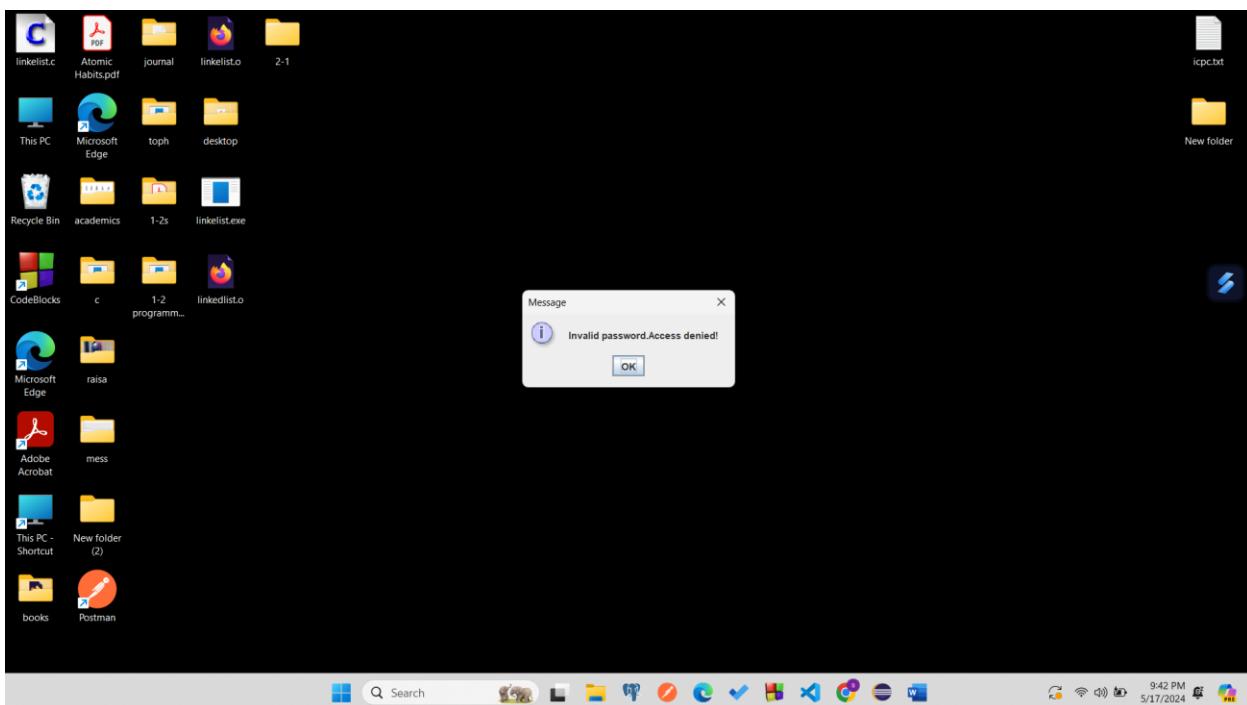
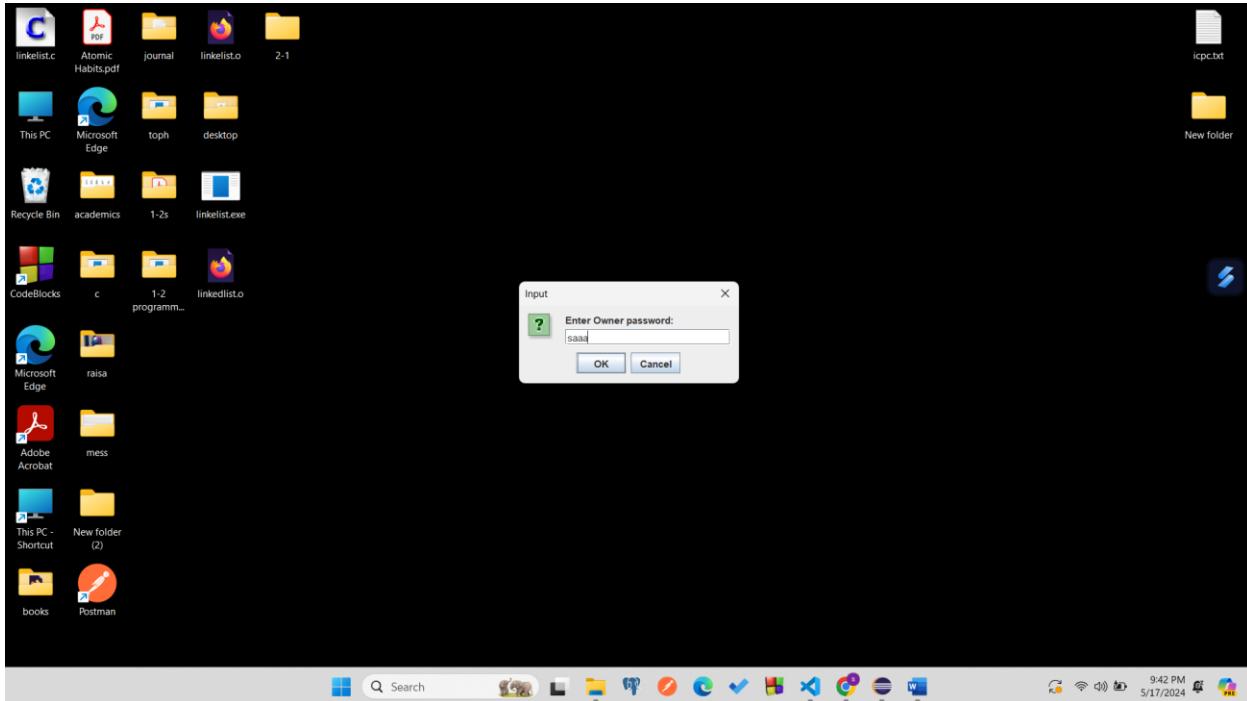


Exceptions:

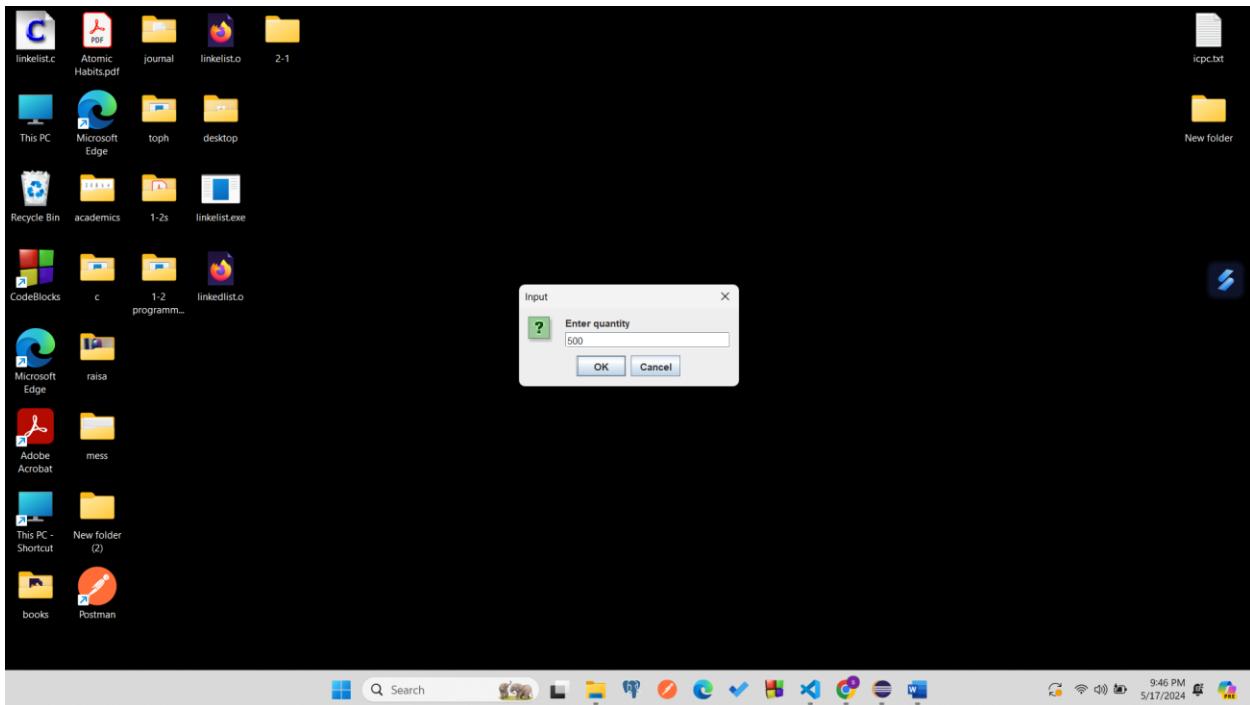
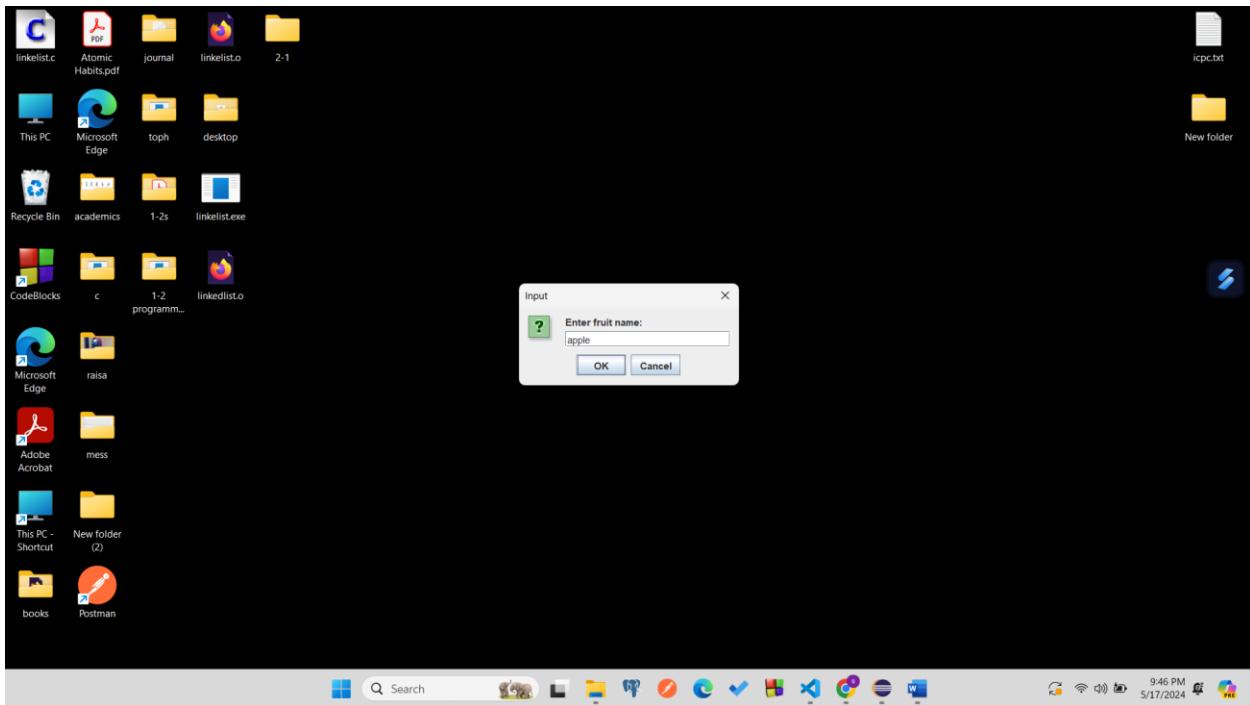
if instead of number, string input is given:

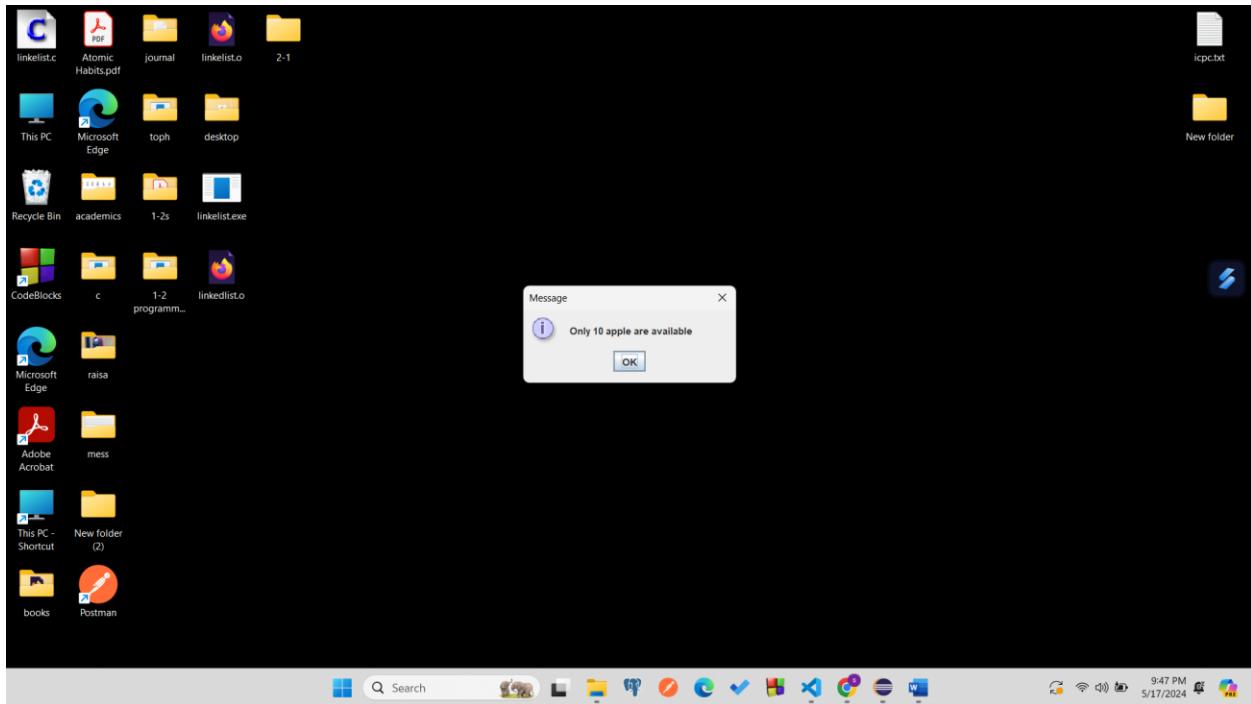


If owner password is wrong:

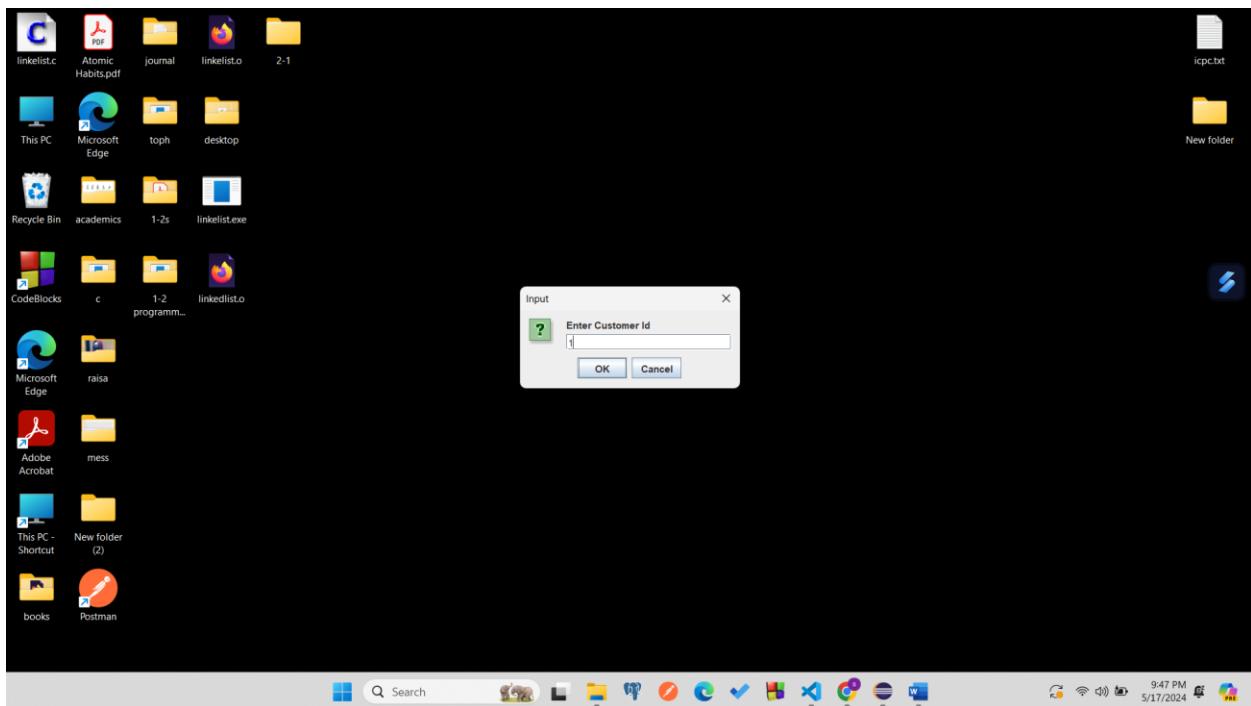


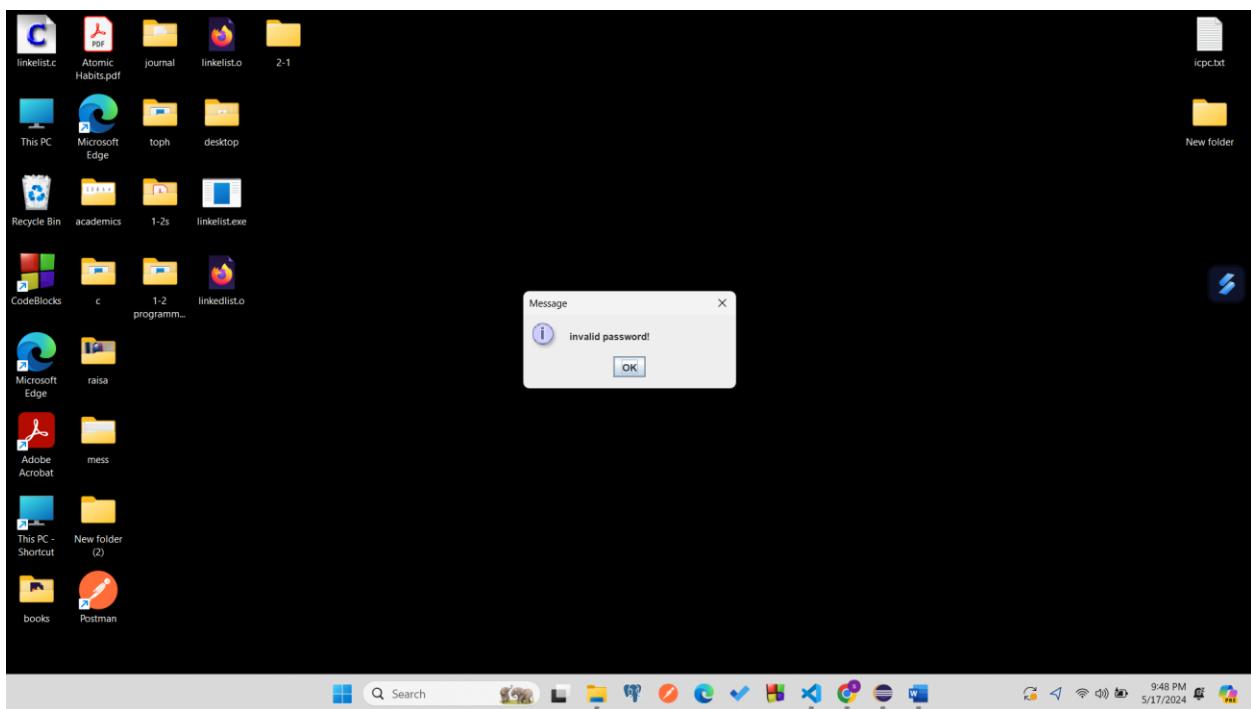
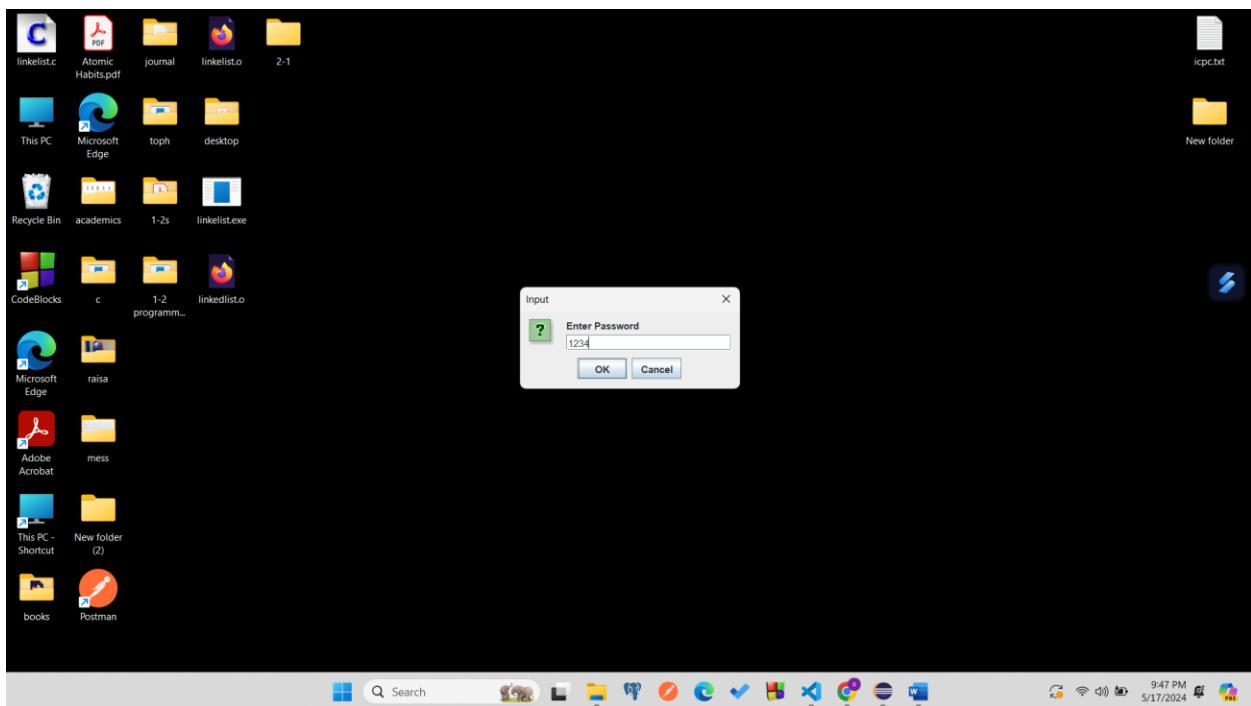
If quantity requested by customer is less than available quantity:



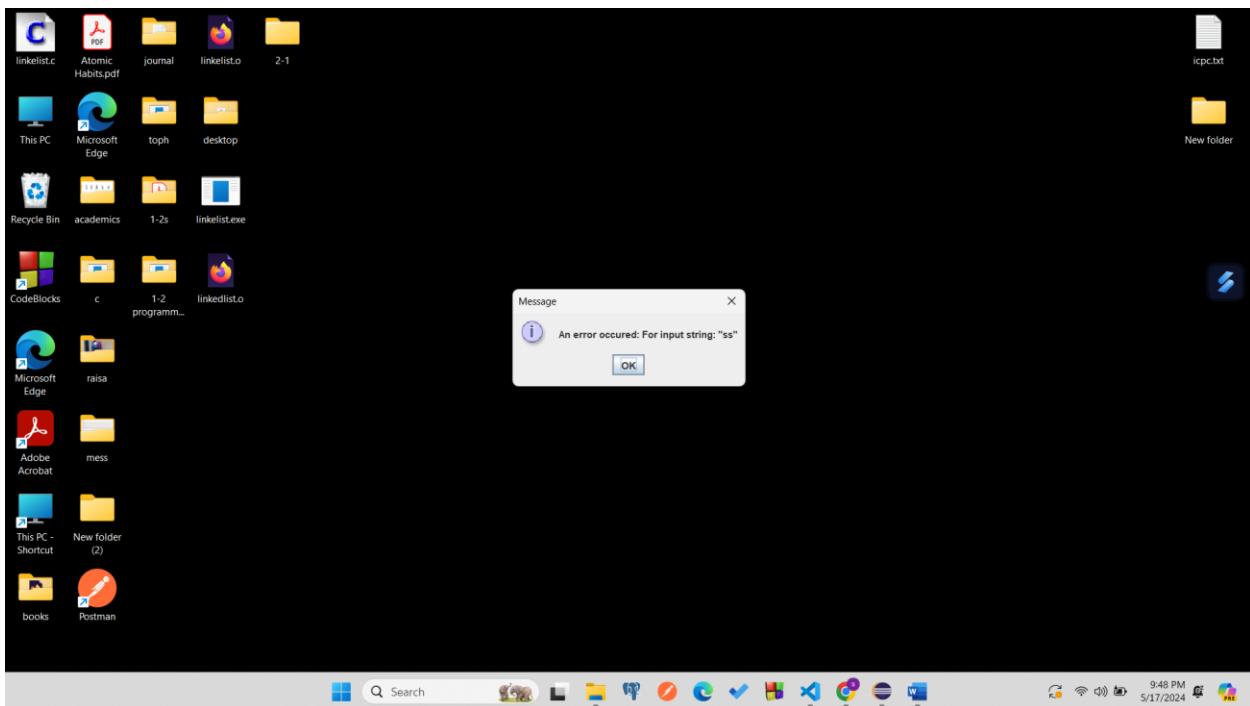
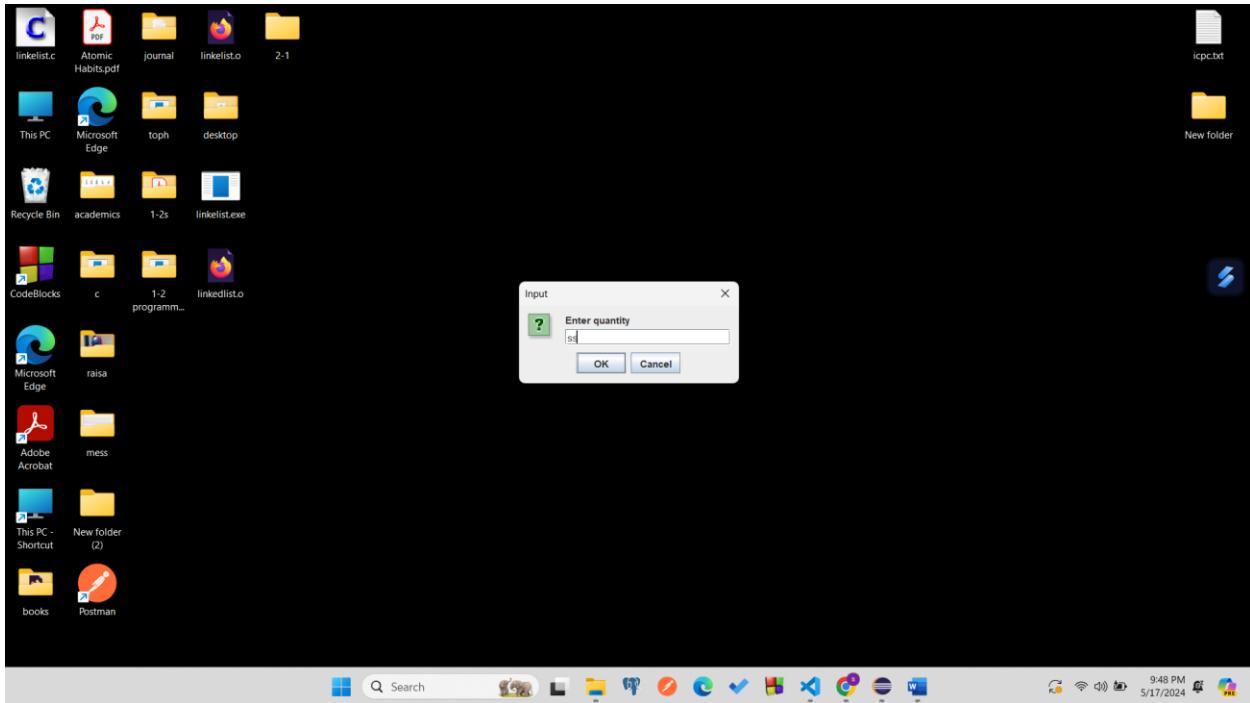


If customer password is wrong:





if instead of number, string input is given:



Exiting from the system:

