

SHOPPIE

PROJECT

DOCUMENTATION

Table of Contents

1. ABSTRACT.....	2
TITLE OF THE PROJECT	2
BRIEF DESCRIPTION OF PROJECT	2
NAME AND HALL TICKET OF TEAM NUMBERS	2
2. ER DIAGRAM.....	3
3. SOFTWARE REQUIREMENTS	4
4. DATA BASE TABLES	4
5. CODE.....	11
conn.java	11
CustomerRegister.java.....	12
CustomerTab.java.....	17
Login.java	21
MyCart.java	26
Myinfo.java	32
MyOrders.java	36
MyProducts.java.....	42
RetailerRegister.java.....	46
RetailerTab.java.....	51
Sellitems.java.....	55
Shop.java.....	60
Test_Login.java.....	68
Wallet.java.....	69
Invalid_tf_shop.java.....	74
ProductNotSelected.java.....	75
6. SCREENSHOTS.....	76
7. CONCLUSION.....	89

SHOPPIE

A PROJECT Done By

Name	Roll Number
T.Karthik Reddy	1602-19-737-137
R.Shitij Reddy	1602-19-737-168
J.Siddharth	1602-19-737-172

ABSTRACT

The project objective is to provide a portal for buying and selling. Shoppie is a desktop based application which provides shopping services. It can have various brands, customers and merchants.

This application will provide an interactive and ease of use platform for users and merchants to buy and sell products at the comfort of home.

It would make searching, viewing and selection of a product easier. It contains a sophisticated search engine for users to search for products specific to their needs. The search engine provides an easy and convenient way to search for products.

Considering this pandemic time, it would be a helping tool for the ones who want to shop from their home.

PROJECT DESCRIPTION:

Shoppie is a desktop based E-Commerce application which provides the an intuitive and easy to use platform for the buyers and sellers.

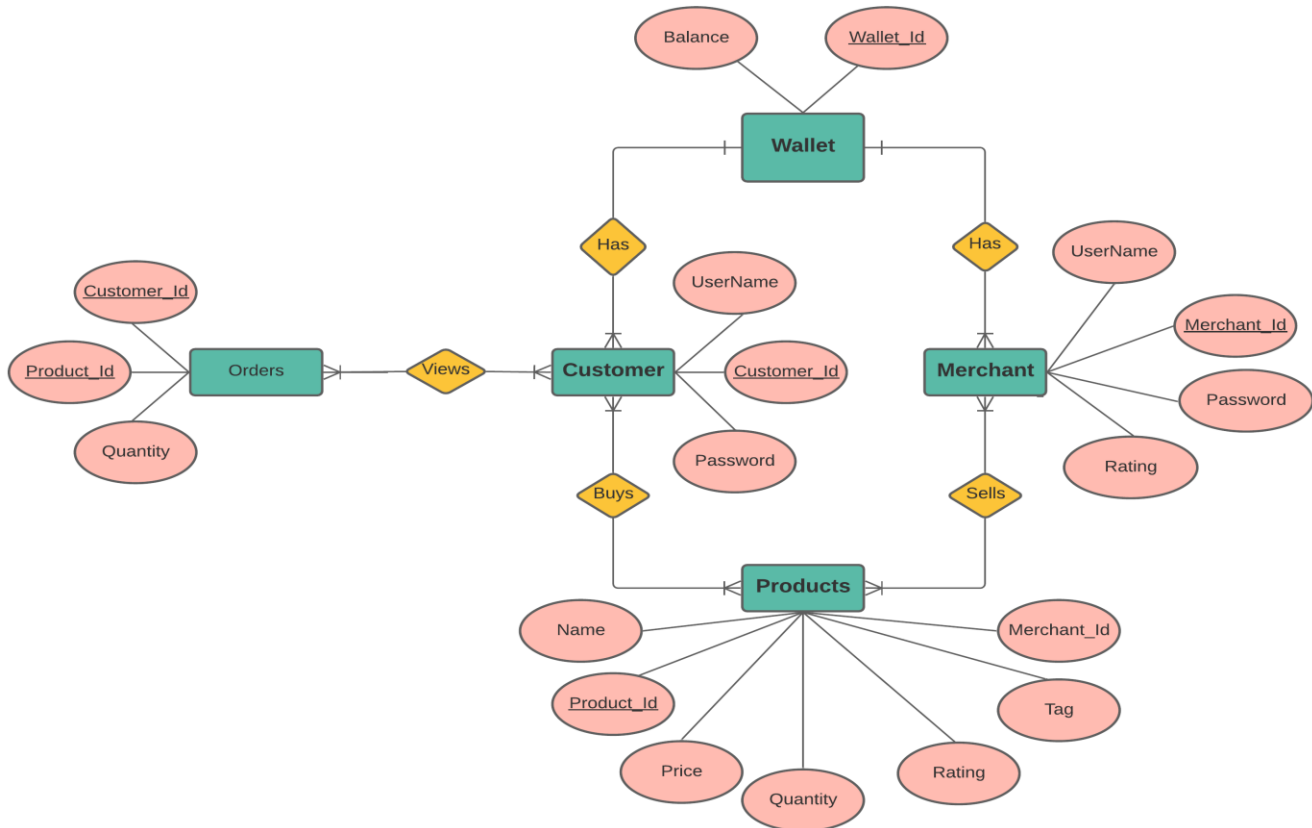
Shoppie is different from the already existing E-Commerce applications. With this application it is possible for both the buyers and sellers to buy/sell products.

Customers can give rating to the products listed by the merchants and to the merchants themselves based on the quality of products being sold.

Using this application, shops can provide a portal where their customers can enjoy easy shopping from anywhere. It helps buying the products in the shop anywhere by using a desktop application. Thus the customer will get the service of online shopping and home delivery from his favorite shop.

For newly started businesses, this application becomes very helpful by providing them with customers.

ER DIAGRAM:



SOFTWARE REQUIREMENTS:

- a) Operating System – Windows 7 and above
- b) JAVA Development Kit (JDK 8)
- c) MySQL Database version 5.7 and above with appropriate Driver Connectors

DATABASE TABLES:

LIST OF TABLES:

- CART
- CUSTOMER
- MERCHANT
- ORDERS
- PRODUCTS
- WALLET

ATTRIBUTES:

[illegible]

Primary key: Product_Id **Foreign Key :** Merchant_Id

	Product_Id	Name	Tag	Price	Rating	Quantity	Merchant_Id
p1	apple iphone X	phone/apple	80000	4	4	m4	
p2	samsung 7s phone	phone/samsung	60000	5	7	m2	
p3	hp pavillion x360 laptop	laptop/hp	85000	4	2	m2	
p4	realme buds	earphone/realme	1200	2	3	m2	
p5	axe intense perfume	perfume/axe	600	1	1	m5	
p6	dove daily care shampoo	shampoo/dove	500	3	2	m1	
	NULL	NULL	NULL	NULL	NULL	NULL	

ORDERS TABLE:

ATTRIBUTES:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Customer_Id	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Product_Id	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Quantity	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Primary key : Customer_Id, Product_Id **Foreign Key :** Customer_Id, Product_Id

Table :

	Customer_Id	Product_Id	Quantity
▶	c1	p2	2
	c1	p4	1
	c2	p3	1
	c3	p1	1
	c4	p5	4
	c5	p2	1
*	NULL	NULL	NULL

WALLET TABLE:

ATTRIBUTES:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Wallet_Id	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Balance	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	




Primary key : Wallet_Id

Table :

	Wallet_Id	Balance
▶	c1	80000
	c2	4500
	c3	25000
	m1	65000
	m2	95000
	m3	45000
*	NULL	NULL

CUSTOMERTABLE:

ATTRIBUTES:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 Customer_Id	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 UserName	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Password	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Primary key : Customer_Id

Table :

	Customer_Id	UserName	Password
▶	c1	siddarth	sidd
	c2	rahul	rahu
	c3	vijay	vija
	c4	vikram	vikr
	c5	pawan	pawa
*	NULL	NULL	NULL

MERCHANT TABLE:

ATTRIBUTES:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
🔑 Merchant_Id	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
💠 UserName	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 Password	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 Rating	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

PRIMARY KEY : Merchant_Id

TABLE:

	Merchant_Id	UserName	Password	Rating
▶	m1	rohith	rohi	4
	m2	deepak	deep	2
	m3	preetham	pree	5
	m4	shankar	shan	3
	m5	manikanta	mani	1
	m6	vikranth	vikr	4
*	NULL	NULL	NULL	NULL

CART TABLE:

ATTRIBUTES:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
🔦 Customer_Id	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
🔹 Products	VARCHAR(90)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

PRIMARY KEY: Customer_Id

TABLE:

	Customer_Id	Products
	c1	p1,p3
	c2	p2,p3,p4
	c5	p6
	c8	p2,p1
▶▶		NULL

CODE:

Conn.java

```
import java.sql.*;
import java.util.Scanner;
public class conn
{
    public Connection c;
    public Statement st;

    public conn()
    {
        String url = "jdbc:mysql://localhost:3306/shoppie_db";
        String userName = "root";
        String db_pass = "project";

        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            c = DriverManager.getConnection(url, userName, db_pass);
            st = c.createStatement();

        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

    }
}
```

CustomerRegister.java

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JTextArea;
import javax.swing.JButton;
import javax.swing.JTextPane;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.awt.event.ActionEvent;
import javax.swing.JPanel;
import java.awt.SystemColor;

public class CustomerRegister {

    private JFrame frame;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JTextField textField_4;
    private JTextField textField_6;
    public Login loginobj;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    CustomerRegister window = new CustomerRegister();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public CustomerRegister() {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
```

```

        try {
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 877, 504);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("Customer Registration");
    lblNewLabel.setForeground(Color.BLACK);
    lblNewLabel.setBackground(Color.BLACK);
    lblNewLabel.setFont(new Font("Verdana", Font.BOLD, 22));
    lblNewLabel.setBounds(291, 11, 292, 48);
    frame.getContentPane().add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Name :");
    lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1.setBounds(102, 104, 66, 34);
    frame.getContentPane().add(lblNewLabel_1);

    textField = new JTextField();
    textField.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField.setBounds(193, 107, 227, 28);
    frame.getContentPane().add(textField);
    textField.setColumns(10);

    textField_1 = new JTextField();
    textField_1.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField_1.setColumns(10);
    textField_1.setBounds(193, 164, 227, 28);
    frame.getContentPane().add(textField_1);

    textField_2 = new JTextField();
    textField_2.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField_2.setColumns(10);
    textField_2.setBounds(193, 228, 227, 28);
    frame.getContentPane().add(textField_2);

    textField_3 = new JTextField();
    textField_3.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField_3.setColumns(10);
    textField_3.setBounds(193, 288, 227, 28);
    frame.getContentPane().add(textField_3);

```

```
JLabel lblNewLabel_1_1 = new JLabel("Email :");
lblNewLabel_1_1.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_1.setBounds(102, 161, 66, 34);
frame.getContentPane().add(lblNewLabel_1_1);
```

```
JLabel lblNewLabel_1_2 = new JLabel("Phone :");
lblNewLabel_1_2.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_2.setBounds(91, 225, 77, 34);
frame.getContentPane().add(lblNewLabel_1_2);
```

```
JLabel lblNewLabel_1_3 = new JLabel("Password :");
lblNewLabel_1_3.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_3.setBounds(68, 285, 100, 34);
frame.getContentPane().add(lblNewLabel_1_3);
```

```
JLabel lblNewLabel_1_4 = new JLabel("Age :");
lblNewLabel_1_4.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_4.setBounds(494, 104, 66, 34);
frame.getContentPane().add(lblNewLabel_1_4);
```

```
JLabel lblNewLabel_1_5 = new JLabel("Address :");
lblNewLabel_1_5.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_5.setBounds(469, 164, 91, 34);
frame.getContentPane().add(lblNewLabel_1_5);
```

```
JLabel lblNewLabel_1_6 = new JLabel("PinCode :");
lblNewLabel_1_6.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_6.setBounds(469, 285, 91, 34);
frame.getContentPane().add(lblNewLabel_1_6);
```

```
textField_4 = new JTextField();
textField_4.setFont(new Font("Tahoma", Font.PLAIN, 18));
textField_4.setColumns(10);
textField_4.setBounds(570, 104, 227, 28);
frame.getContentPane().add(textField_4);
```

```
textField_6 = new JTextField();
textField_6.setFont(new Font("Tahoma", Font.PLAIN, 18));
textField_6.setColumns(10);
textField_6.setBounds(570, 288, 227, 28);
frame.getContentPane().add(textField_6);
```

```
JTextArea textArea = new JTextArea();
textArea.setBounds(570, 164, 227, 99);
frame.getContentPane().add(textArea);
```

```
JButton btnNewButton = new JButton("Register");
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
btnNewButton.setBounds(570, 393, 123, 41);
frame.getContentPane().add(btnNewButton);
```

```
JButton btnNewButton_1 = new JButton("Back");
btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD, 18));
btnNewButton_1.setBounds(193, 393, 123, 41);
frame.getContentPane().add(btnNewButton_1);
```

```

JPanel panel = new JPanel();
panel.setBackground(SystemColor.desktop);
panel.setBounds(0, 0, 863, 69);
frame.getContentPane().add(panel);

//Register Button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try
        {
            String username = textField.getText();
            String password = textField_3.getText();
            String phone = textField_2.getText();
            String address = textArea.getText();
            String pincode = textField_6.getText();

            conn cobj = new conn();

            String query = "select * from customer;";
            ResultSet rs = cobj.st.executeQuery(query);

            boolean flag = false;
            while(rs.next())
            {
                if(rs.getString("username").equals(username))
                {
                    JOptionPane.showMessageDialog(frame, "Username already
exists !", "Register", JOptionPane.ERROR_MESSAGE);
                    flag = true;
                    break;
                }
            }

            if(!flag)//user name does not exist
            {

                String query1 = "select count(*) as count from customer;";

                rs = cobj.st.executeQuery(query1);
                rs.next();
                int cnum = rs.getInt("count") + 1;
                String cust_id = "c" + Integer.toString(cnum);

                String query2 = "insert into customer values('"+cust_id+"',
 '"+username+"', '"+password+"', '"+phone+"', '"+address+"', '"+pincode+"')";

                cobj.st.executeUpdate(query2);

                Wallet.createWallet(cust_id);

                JOptionPane.showMessageDialog(frame, "Registration
successful", "Register", JOptionPane.INFORMATION_MESSAGE);

```

```
                btnNewButton_1.doClick();
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
});

//Back button
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.setVisible(false);
        Login.loginobj.frmShoppie.setVisible(true);
    }
});
}
}
```


CustomerTab.java

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JList;
import java.awt.Color;
import javax.swing.JTabbedPane;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.SystemColor;
import javax.swing.ImageIcon;
import javax.swing.JPanel;
import javax.swing.SwingConstants;

public class CustomerTab {

    public static JFrame frame;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    CustomerTab window = new CustomerTab();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public CustomerTab() {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });

        initialize();
    }
}
```

```

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.getContentPane().setBackground(Color.GRAY);
    frame.setBounds(100, 100, 941, 576);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    String uname = Login.username_ip.toUpperCase();
    JLabel lblNewLabel = new JLabel("Hello " + uname + " !");
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setBackground(Color.BLACK);
    lblNewLabel.setForeground(Color.BLACK);
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 28));
    lblNewLabel.setBounds(157, 11, 598, 54);
    frame.getContentPane().add(lblNewLabel);

    JButton btnNewButton = new JButton("Shop");
    btnNewButton.setBorderPainted(false);
    btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnNewButton.setForeground(Color.WHITE);
    btnNewButton.setBackground(Color.BLACK);
    btnNewButton.setBounds(354, 111, 171, 54);
    frame.getContentPane().add(btnNewButton);

    JButton btnMyCart = new JButton("My Cart");
    btnMyCart.setBorderPainted(false);
    btnMyCart.setForeground(Color.WHITE);
    btnMyCart.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnMyCart.setBackground(Color.BLACK);
    btnMyCart.setBounds(554, 226, 171, 54);
    frame.getContentPane().add(btnMyCart);

    JButton btnMyOrders = new JButton("My Orders");
    btnMyOrders.setBorderPainted(false);
    btnMyOrders.setForeground(Color.WHITE);
    btnMyOrders.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnMyOrders.setBackground(Color.BLACK);
    btnMyOrders.setBounds(157, 226, 171, 54);
    frame.getContentPane().add(btnMyOrders);

    JButton btnMyWallet = new JButton("My Wallet");
    btnMyWallet.setBorderPainted(false);
    btnMyWallet.setForeground(Color.WHITE);
    btnMyWallet.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnMyWallet.setBackground(Color.BLACK);
    btnMyWallet.setBounds(157, 330, 171, 54);
    frame.getContentPane().add(btnMyWallet);

    JButton btnBack = new JButton("Logout");
    btnBack.setBorderPainted(false);
    btnBack.setForeground(Color.WHITE);
    btnBack.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnBack.setBackground(Color.BLACK);

```

```

btnBack.setBounds(376, 471, 162, 41);
frame.getContentPane().add(btnBack);

JButton btnMyInfo = new JButton("My Info");
btnMyInfo.setBorderPainted(false);
btnMyInfo.setForeground(Color.WHITE);
btnMyInfo.setFont(new Font("Tahoma", Font.BOLD, 18));
btnMyInfo.setBackground(Color.BLACK);
btnMyInfo.setBounds(554, 330, 171, 54);
frame.getContentPane().add(btnMyInfo);

JPanel panel = new JPanel();
panel.setBackground(new Color(255, 99, 71));
panel.setBounds(0, 0, 927, 77);
frame.getContentPane().add(panel);

JLabel lblNewLabel_1 = new JLabel("New label");
lblNewLabel_1.setIcon(new ImageIcon(CustomerTab.class.getResource("/icons/ecom1.jpg")));
lblNewLabel_1.setBounds(0, 76, 927, 463);
frame.getContentPane().add(lblNewLabel_1);

//logout button
btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        Login.loginobj.frmShoppie.setVisible(true);

    }
});

//shop button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        new Shop();

    }
});

//Wallet button
btnMyWallet.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        new Wallet();

    }
});

//My orders button
btnMyOrders.addActionListener(new ActionListener() {

```

```
        public void actionPerformed(ActionEvent arg0) {

            frame.dispose();
            new MyOrders();
        }
    });

    //my cart button
    btnMyCart.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

            frame.dispose();
            new MyCart();
        }
    });

    //my info
    btnMyInfo.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

            frame.dispose();
            new MyInfo();
        }
    });
}
}
```

Login.java (MainClass)

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.awt.event.ActionEvent;
import java.awt.SystemColor;
import javax.swing.JPanel;
import javax.swing.ImageIcon;
import java.awt.Toolkit;
import javax.swing.JToggleButton;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;
import javax.swing.JSeparator;
import javax.swing.JRadioButton;
import javax.swing.ButtonGroup;
import javax.swing.AbstractAction;
import javax.swing.Action;
import javax.swing.SwingConstants;

public class Login {

    public JFrame frmShoppie;
    private JTextField txtField;
    private JPasswordField passwordField;
    private final ButtonGroup buttonGroup = new ButtonGroup();
    public String userType = "Customer";
    public static Login loginobj;
    public static String username_ip;
    public static String pass_ip;
    public static String userId;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Login window = new Login();
                    window.frmShoppie.setVisible(true);

                    try {

                        UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");
                        SwingUtilities.updateComponentTreeUI(window.frmShoppie);
```

```

        }
        catch (Exception ex) {
            JOptionPane.showMessageDialog(window.frmShoppie, "LookAndFeel
not set", "ERROR",
                                     JOptionPane.ERROR_MESSAGE);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

});
}

/**
 * Create the application.
 */
public Login() {
    loginobj = this;

    initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frmShoppie = new JFrame();
    frmShoppie.setTitle("Shoppie");
    frmShoppie.getContentPane().setBackground(SystemColor.scrollbar);
    frmShoppie.setBounds(100, 100, 510, 552);
    frmShoppie.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmShoppie.getContentPane().setLayout(null);

    JLabel lblNewLabel_3 = new JLabel("User Name :");
    lblNewLabel_3.setBackground(new Color(230, 230, 250));
    lblNewLabel_3.setFont(new Font("Tahoma", Font.BOLD, 20));
    lblNewLabel_3.setBounds(67, 239, 142, 32);
    frmShoppie.getContentPane().add(lblNewLabel_3);

    txtField = new JTextField();
    txtField.setFont(new Font("Tahoma", Font.PLAIN, 15));
    txtField.setBounds(219, 245, 157, 26);
    frmShoppie.getContentPane().add(txtField);
    txtField.setColumns(10);

    JLabel lblNewLabel_4 = new JLabel("Password :");
    lblNewLabel_4.setFont(new Font("Tahoma", Font.BOLD, 20));
    lblNewLabel_4.setBounds(67, 297, 119, 32);
    frmShoppie.getContentPane().add(lblNewLabel_4);

    passwordField = new JPasswordField();
    passwordField.setFont(new Font("Tahoma", Font.PLAIN, 15));
    passwordField.setBounds(219, 302, 157, 28);
    frmShoppie.getContentPane().add(passwordField);

```

```

JButton btnNewButton = new JButton("Login");
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 22));
btnNewButton.setForeground(Color.BLACK);
btnNewButton.setBackground(Color.WHITE);
btnNewButton.setBounds(219, 369, 119, 45);
frmShoppie.getContentPane().add(btnNewButton);

JLabel lblNewLabel_5 = new JLabel("New User ?");
lblNewLabel_5.setFont(new Font("Tahoma", Font.PLAIN, 18));
lblNewLabel_5.setBounds(106, 435, 103, 40);
frmShoppie.getContentPane().add(lblNewLabel_5);

JRadioButton rdbtnNewRadioButton = new JRadioButton("Customer");
buttonGroup.add(rdbtnNewRadioButton);
rdbtnNewRadioButton.setSelected(true);
rdbtnNewRadioButton.setFont(new Font("Tahoma", Font.PLAIN, 15));
rdbtnNewRadioButton.setBounds(219, 183, 111, 23);
frmShoppie.getContentPane().add(rdbtnNewRadioButton);

JRadioButton rdbtnNewRadioButton_1 = new JRadioButton("Retailer");
buttonGroup.add(rdbtnNewRadioButton_1);
rdbtnNewRadioButton_1.setFont(new Font("Tahoma", Font.PLAIN, 15));
rdbtnNewRadioButton_1.setBounds(354, 183, 111, 23);
frmShoppie.getContentPane().add(rdbtnNewRadioButton_1);

JLabel lblNewLabel_6 = new JLabel("Iam a :");
lblNewLabel_6.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_6.setForeground(new Color(0, 0, 0));
lblNewLabel_6.setBackground(Color.BLACK);
lblNewLabel_6.setFont(new Font("Tahoma", Font.BOLD, 20));
lblNewLabel_6.setBounds(67, 176, 87, 32);
frmShoppie.getContentPane().add(lblNewLabel_6);

JButton btnNewButton_1 = new JButton("Register");
btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD, 15));
btnNewButton_1.setBounds(219, 443, 111, 26);
frmShoppie.getContentPane().add(btnNewButton_1);

JLabel lblNewLabel_13 = new JLabel("");
lblNewLabel_13.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_13.setIcon(new ImageIcon(Login.class.getResource("/icons/shop6.gif")));
lblNewLabel_13.setBounds(0, 0, 496, 149);
frmShoppie.getContentPane().add(lblNewLabel_13);

rdbtnNewRadioButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        userType = "Customer";
    }
});

rdbtnNewRadioButton_1.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent arg0) {

            userType = "Retailer";

        }

    });

//Login Button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        conn cobj = new conn();

        String query;

        if(userType.equals("Customer"))
        {
            query = "SELECT * FROM customer;";
        }
        else
        {
            query = "SELECT * FROM merchant;";
        }

        System.out.println(userType);
        username_ip = txtField.getText();
        pass_ip = passwordField.getText();

        try
        {
            ResultSet rs = cobj.st.executeQuery(query);

            Boolean access = false;
            while(rs.next())
            {
                String uname = rs.getString("username");
                String pass = rs.getString("password");

                if(uname.equals(username_ip) && pass.equals(pass_ip))
                {
                    access = true;
                    break;
                }
            }
            if(access)
            {
                System.out.println("Access Granted!");
                JOptionPane.showMessageDialog(frmShoppie, "Login
Successful", "Login", JOptionPane.INFORMATION_MESSAGE);

                if(userType.equals("Customer"))
                {
                    userId = rs.getString("Customer_Id");

                    frmShoppie.setVisible(false);

```



```

        new CustomerTab();
    }
    else
    {
        userId = rs.getString("Merchant_Id");

        frmShoppie.setVisible(false);
        new RetailerTab();
    }

}
else
{
    System.out.println("Access Denied");
    JOptionPane.showMessageDialog(frmShoppie, "Username or Password
is incorrect !","Login",JOptionPane.ERROR_MESSAGE);
}

    txtField.setText("");
    passwordField.setText("");
}
catch(Exception e)
{
    System.out.println(e);
}
}

});

//Register button
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        if(userType.equals("Customer"))
        {
            System.out.println("Customer Registration");
            frmShoppie.setVisible(false);
            new CustomerRegister();

        }
        else
        {
            System.out.println("Retailer Registration");
            frmShoppie.setVisible(false);
            new RetailerRegister();
        }
    }
});

}
}

```

Mycart.java

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.JTableHeader;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;
import java.awt.event.ActionEvent;
import javax.swing.JTextField;
import java.awt.SystemColor;

public class MyCart {

    private JFrame frame;
    private JTable table;
    private DefaultTableModel model;
    private JTextField textField;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    MyCart window = new MyCart();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
```

```

* Create the application.
*/
public MyCart() {

    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });

    initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.getContentPane().setBackground(new Color(192, 192, 192));
    frame.setBounds(100, 100, 902, 566);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JPanel panel = new JPanel();
    panel.setBounds(0, 0, 888, 55);
    panel.setBackground(SystemColor.desktop);
    frame.getContentPane().add(panel);

    JLabel lblNewLabel = new JLabel("My Cart");
    panel.add(lblNewLabel);
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 28));

    //Table start
    table = new JTable();
    table.setRowHeight(20);
    table.setFont(new Font("Tahoma", Font.PLAIN, 15));
    model = new DefaultTableModel(
        new Object[][] {

            },
        new String[] {
            "SNo.", "Product Id", "Product Name", "Price", "Rating", "Quantity"
        }
    );

    table.setModel(model);
    table.getColumnModel().getColumn(0).setPreferredWidth(4);
    table.getColumnModel().getColumn(1).setPreferredWidth(20);

    table.getColumnModel().getColumn(2).setPreferredWidth(200);

    table.getColumnModel().getColumn(3).setPreferredWidth(50);

```

```

table.getColumnModel().getColumn(4).setPreferredWidth(50);
table.getColumnModel().getColumn(5).setPreferredWidth(50);
//table.setBounds(28, 77, 832, 364);
//frame.getContentPane().add(table);

JScrollPane scrollPane = new JScrollPane(table);
scrollPane.setBounds(28, 77, 832, 364);
scrollPane.setBorder(new LineBorder(new Color(130, 135, 144), 3, true));
scrollPane.setFont(new Font("Tahoma", Font.BOLD, 15));

JTableHeader Theader = table.getTableHeader();
Theader.setBackground(Color.GRAY);
Theader.setForeground(Color.BLACK);
Theader.setFont(new Font("Tahoma", Font.BOLD, 15));
frame.getContentPane().add(scrollPane);

loadCartTable();
// Table end

JButton btnNewButton = new JButton("Back");
btnNewButton.setBounds(64, 474, 95, 31);
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
frame.getContentPane().add(btnNewButton);

JButton btnBuy = new JButton("Buy");
btnBuy.setFont(new Font("Tahoma", Font.BOLD, 18));
btnBuy.setBounds(741, 474, 95, 31);
frame.getContentPane().add(btnBuy);

JLabel lblNewLabel_1 = new JLabel("Quantity :");
lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1.setBounds(373, 473, 95, 33);
frame.getContentPane().add(lblNewLabel_1);

textField = new JTextField();
textField.setFont(new Font("Tahoma", Font.PLAIN, 14));
textField.setBounds(478, 474, 76, 31);
frame.getContentPane().add(textField);
textField.setColumns(10);

//Back button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        new CustomerTab();
    }
});

//Buy button
btnBuy.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        int orderQuantity=0;

```

```

int prod_Price=0;
String prod_Id=null;
String prod_Name=null;
String prod_Pricestr=null;
int prod_Quantity=0;

try
{
    try
    {
        prod_Id = (String) model.getValueAt(table.getSelectedRow(), 1);
        prod_Name = (String) model.getValueAt(table.getSelectedRow(), 2);
        prod_Pricestr = (String) model.getValueAt(table.getSelectedRow(), 3);
        prod_Quantity = Integer.parseInt((String)
model.getValueAt(table.getSelectedRow(), 5));
        prod_Price = Integer.parseInt(prod_Pricestr);
    }
    catch(Exception e)
    {
        throw new MyException.ProductNotSelected("");
    }

    try{orderQuantity = Integer.parseInt(textField.getText());}
    catch(Exception e) {throw new MyException.Invalid_tf_shop("");}

    if(orderQuantity==0) {throw new MyException.Invalid_tf_shop("");}

    int totalPrice = prod_Price*orderQuantity;
    int currBalance = Wallet.checkBalance(Login.userId);

    //successful purchase
    if((prod_Quantity>=orderQuantity) && (currBalance >= totalPrice))
    {
        Wallet.updateBalance(currBalance - totalPrice);
        Wallet.addToRetailerBalance(Shop.getRetailerId(prod_Id), totalPrice);
        Shop.updateQuantity(prod_Id, prod_Quantity - orderQuantity);
        Shop.addToOrders(Login.userId, prod_Id, orderQuantity);
        removeFromCart(prod_Id);
        JOptionPane.showMessageDialog(frame, prod_Name + "
("+orderQuantity+") Successfully purchased", "Purchase",JOptionPane.INFORMATION_MESSAGE);
    }
    else if(currBalance < totalPrice)
    {
        JOptionPane.showMessageDialog(frame, "Insufficient funds in wallet
!nWallet balance : "+currBalance, "Purchase",JOptionPane.INFORMATION_MESSAGE);
    }
    else
    {
        JOptionPane.showMessageDialog(frame, "Products unavailable
currently !", "Purchase" ,JOptionPane.INFORMATION_MESSAGE);
    }

}
catch(MyException.Invalid_tf_shop e1)

```

```

        {
            JOptionPane.showMessageDialog(frame, "Enter valid Quantity
!", "Purchase", JOptionPane.INFORMATION_MESSAGE);
        }
        catch(MyException.ProductNotSelected e2)
        {
            JOptionPane.showMessageDialog(frame, "First select a product to buy !",
"Purchase" ,JOptionPane.INFORMATION_MESSAGE);
        }
    }
});

}

private void loadCartTable()
{
    conn cobj = new conn();

    String query1 = "SELECT products FROM cart where Customer_Id='" + Login.userId + "'";

    String prodstr;
    String[] products;
    try
    {
        ResultSet rs = cobj.st.executeQuery(query1);
        rs.next();
        prodstr = rs.getString("products");
        products = prodstr.split(",");

        int index = 1;

        for(String prodId : products)
        {
            String query2 = "select name,price,rating,quantity from products where Product_Id='" +
prodId + "'";

            rs = cobj.st.executeQuery(query2);
            rs.next();

            String prodName = rs.getString("Name");
            String prodPrice = rs.getString("Price");
            String prodRating = rs.getString("Rating");
            String prodQuantity = rs.getString("Quantity");

            Object[] newRow = {index, prodId, prodName, prodPrice, prodRating, prodQuantity};
            model.addRow(newRow);

            index++;
        }
    }
    catch (SQLException e)
    {
        //empty cart
        e.printStackTrace();
    }
}

```

```

    }

    private void removeFromCart(String prodId)
    {
        conn cobj = new conn();

        String query1 = "SELECT products FROM cart where Customer_Id=" + Login.userId + " ";

        try
        {
            ResultSet rs = cobj.st.executeQuery(query1);
            rs.next();
            String prodStr = rs.getString("products");
            List<String> products = new LinkedList(Arrays.asList(prodStr.split(",")));

            products.remove(prodId);
            String newProdStr = String.join(",", products);

            String query2 = "update cart set products=" + newProdStr + " where Customer_Id=" +
Login.userId + " ";

            cobj.st.executeUpdate(query2);
        }
        catch(SQLException e)
        {
            e.printStackTrace();
        }

        model.removeRow(table.getSelectedRow());
    }
}

```

MyInfo.java

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JTextPane;
import javax.swing.JSeparator;
import javax.swing.SwingConstants;
import javax.swing.JPanel;
import java.awt.SystemColor;

public class MyInfo {

    private JFrame frame;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JTextPane textPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    MyInfo window = new MyInfo();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public MyInfo()
    {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
```



```

        try {
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 951, 469);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("My Info");
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 25));
    lblNewLabel.setBounds(390, 11, 114, 45);
    frame.getContentPane().add(lblNewLabel);

    String userName = Login.username_ip;
    JLabel lblNewLabel_1 = new JLabel("Name : ");
    lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1.setBounds(29, 101, 81, 36);
    frame.getContentPane().add(lblNewLabel_1);

    JLabel lblNewLabel_1_1 = new JLabel("Password : ");
    lblNewLabel_1_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1_1.setBounds(521, 101, 105, 36);
    frame.getContentPane().add(lblNewLabel_1_1);

    JLabel lblNewLabel_1_2 = new JLabel("Phone : ");
    lblNewLabel_1_2.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1_2.setBounds(29, 180, 81, 36);
    frame.getContentPane().add(lblNewLabel_1_2);

    JButton btnNewButton_1 = new JButton("Back");
    btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnNewButton_1.setBounds(61, 354, 94, 36);
    frame.getContentPane().add(btnNewButton_1);

    JLabel lblNewLabel_1_2_1 = new JLabel("Address :");
    lblNewLabel_1_2_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1_2_1.setBounds(521, 195, 99, 36);
    frame.getContentPane().add(lblNewLabel_1_2_1);

    JLabel lblNewLabel_1_2_2 = new JLabel("Pincode : ");
    lblNewLabel_1_2_2.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1_2_2.setBounds(29, 246, 94, 36);

```

```

frame.getContentPane().add(lblNewLabel_1_2_2);

textPane = new JTextPane();
textPane.setFont(new Font("Tahoma", Font.PLAIN, 15));
textPane.setEditable(false);
textPane.setBounds(630, 195, 272, 150);
frame.getContentPane().add(textPane);

textField = new JTextField();
textField.setFont(new Font("Tahoma", Font.PLAIN, 15));
textField.setEditable(false);
textField.setBounds(120, 104, 232, 36);
frame.getContentPane().add(textField);
textField.setColumns(10);

textField_1 = new JTextField();
textField_1.setFont(new Font("Tahoma", Font.PLAIN, 15));
textField_1.setEditable(false);
textField_1.setColumns(10);
textField_1.setBounds(630, 104, 232, 36);
frame.getContentPane().add(textField_1);

textField_2 = new JTextField();
textField_2.setFont(new Font("Tahoma", Font.PLAIN, 15));
textField_2.setEditable(false);
textField_2.setColumns(10);
textField_2.setBounds(120, 183, 232, 36);
frame.getContentPane().add(textField_2);

textField_3 = new JTextField();
textField_3.setFont(new Font("Tahoma", Font.PLAIN, 15));
textField_3.setEditable(false);
textField_3.setColumns(10);
textField_3.setBounds(120, 249, 232, 36);
frame.getContentPane().add(textField_3);

JSeparator separator = new JSeparator();
separator.setOrientation(SwingConstants.VERTICAL);
separator.setBounds(444, 84, 27, 306);
frame.getContentPane().add(separator);

JPanel panel = new JPanel();
panel.setBackground(SystemColor.desktop);
panel.setBounds(0, 0, 937, 68);
frame.getContentPane().add(panel);

loadData();

//back button
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        if(Login.userId.charAt(0)=='c')
        {

```

```

        new CustomerTab();
    }
    else
    {
        new RetailerTab();
    }
}

});

}

private void loadData()
{
    conn cobj = new conn();
    String query;

    if(Login.userId.charAt(0)=='c')
    {
        query = "select * from customer where Customer_Id='" + Login.userId + "';";
    }
    else
    {
        query = "select * from merchant where Merchant_Id='" + Login.userId + "';";
    }

    try
    {
        ResultSet rs = cobj.st.executeQuery(query);
        rs.next();

        String name = rs.getString("username");
        String password = rs.getString("password");
        String phone = rs.getString("phone");
        String address = rs.getString("address");
        String pincode = rs.getString("pincode");

        textField.setText(name);
        textField_1.setText(password);
        textField_2.setText(phone);
        textField_3.setText(pincode);
        textPane.setText(address);
    }
    catch(SQLException se)
    {
        se.printStackTrace();
    }
}
}

```

MyOrders.java

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.JTableHeader;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.awt.event.ActionEvent;
import java.awt.SystemColor;

public class MyOrders {

    private JFrame frame;
    private JTable table;
    private DefaultTableModel model;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    MyOrders window = new MyOrders();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public MyOrders() {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
```

```

        frame.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

});

initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.getContentPane().setBackground(SystemColor.scrollbar);
    frame.setBounds(100, 100, 902, 566);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JPanel panel = new JPanel();
    panel.setBackground(SystemColor.desktop);
    panel.setBounds(0, 0, 888, 55);
    frame.getContentPane().add(panel);

    JLabel lblNewLabel = new JLabel("My Orders");
    panel.add(lblNewLabel);
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 28));

    //Table start
    table = new JTable();
    table.setRowHeight(20);
    table.setFont(new Font("Tahoma", Font.PLAIN, 15));
    model = new DefaultTableModel(
        new Object[][] {

            },
        new String[] {
            "SNo.", "Order Number", "Product Id", "Product Name", "Price", "Quantity"
        }
    );

    table.setModel(model);
    table.getColumnModel().getColumn(0).setPreferredWidth(2);
    table.getColumnModel().getColumn(1).setPreferredWidth(80);

    table.getColumnModel().getColumn(2).setPreferredWidth(20);

    table.getColumnModel().getColumn(3).setPreferredWidth(237);
    table.getColumnModel().getColumn(4).setPreferredWidth(70);
    table.getColumnModel().getColumn(5).setPreferredWidth(109);
    //table.setBounds(28, 77, 832, 364);
    //frame.getContentPane().add(table);

    JScrollPane scrollPane = new JScrollPane(table);
    scrollPane.setBorder(new LineBorder(new Color(130, 135, 144), 3, true));

```

```

scrollPane.setFont(new Font("Tahoma", Font.BOLD, 15));
scrollPane.setBounds(28, 77, 832, 364);

JTableHeader Theader = table.getTableHeader();
Theader.setBackground(Color.GRAY);
Theader.setForeground(Color.BLACK);
Theader.setFont(new Font("Tahoma", Font.BOLD, 15));
frame.getContentPane().add(scrollPane);

loadOrderTable();
// Table end

JButton btnNewButton = new JButton("Back");
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
btnNewButton.setBounds(64, 474, 95, 31);
frame.getContentPane().add(btnNewButton);

JComboBox comboBox = new JComboBox();
comboBox.setModel(new DefaultComboBoxModel(new String[] { "1", "2", "3", "4", "5" }));
comboBox.setFont(new Font("Tahoma", Font.BOLD, 20));
comboBox.setBounds(587, 470, 50, 37);
frame.getContentPane().add(comboBox);

JLabel lblNewLabel_1 = new JLabel("Rating :");
lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 20));
lblNewLabel_1.setBounds(495, 474, 95, 28);
frame.getContentPane().add(lblNewLabel_1);

JButton btnSubmit = new JButton("Submit");
btnSubmit.setFont(new Font("Tahoma", Font.BOLD, 18));
btnSubmit.setBounds(680, 474, 116, 31);
frame.getContentPane().add(btnSubmit);

//Back button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        CustomerTab.frame.setVisible(true);
    }
});

//submit button
btnSubmit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        int orderId;
        String prodId = "";
        int ratingIp = 2;

        try
        {
            try

```

```

        {
            orderId = Integer.parseInt((String)
model.getValueAt(table.getSelectedRow(), 1));
            prodId = (String) model.getValueAt(table.getSelectedRow(), 2);
            ratingIp = Integer.parseInt(comboBox.getSelectedItem().toString());
        }
        catch(Exception e)
        {
            e.printStackTrace();
            throw new MyException.ProductNotSelected("");
        }

        try
        {
            updateProductRating(prodId, ratingIp);
            updateRetailerRating(Shop.getRetailerId(prodId), ratingIp);

            model.removeRow(table.getSelectedRow());

            deleteOrder(orderId);
            JOptionPane.showMessageDialog(frame, "Thanks you for your feedback
!", "Rating" ,JOptionPane.INFORMATION_MESSAGE);
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
    }
    catch(MyException.ProductNotSelected e2)
    {
        JOptionPane.showMessageDialog(frame, "First select a product !", "Rating"
,JOptionPane.INFORMATION_MESSAGE);
    }

    });
}

private void loadOrderTable()
{
    conn cobj = new conn();

    String query = "SELECT * FROM orders where Customer_Id='" + Login.userId + "' order by Order_Id
desc;";

    try
    {
        ResultSet rs = cobj.st.executeQuery(query);

        int index = 1;
        while(rs.next())
        {
            String orderId = rs.getString("Order_Id");
            String prodId = rs.getString("Product_Id");
            String prodQuantity = rs.getString("Quantity");

```

```

        String prodName = getProdName(prodId);
        String prodPrice = getProdPrice(prodId);

        Object[] newRow = {index, orderId, prodId, prodName, prodPrice, prodQuantity};
        model.addRow(newRow);

        index++;
    }
}
catch (SQLException e)
{
    e.printStackTrace();
}
}

```

```

private String getProdPrice(String prodId) throws SQLException {

    conn cobj = new conn();

    String query = "SELECT price from products where Product_Id=" + prodId + """;

    ResultSet rs = cobj.st.executeQuery(query);
    rs.next();

    return rs.getString("price");
}

```

```

private String getProdName(String prodId) throws SQLException
{

    conn cobj = new conn();

    String query = "SELECT name from products where Product_Id=" + prodId + """;

    ResultSet rs = cobj.st.executeQuery(query);
    rs.next();

    return rs.getString("name");
}

```

```

public void updateProductRating(String prodId, int ratingIp) throws SQLException
{

    conn cobj = new conn();

    String query1 = "SELECT rating, reviews FROM products where Product_Id=" + prodId + """;

    ResultSet rs = cobj.st.executeQuery(query1);
    rs.next();

    double rating = rs.getDouble("rating");
    int reviews = rs.getInt("reviews");
}

```



```

        double newRating;
        if(reviews==0) {newRating = ratingIp;}
        else {newRating = ((reviews*rating)+ratingIp)/(reviews+1);}

        int newReviews = reviews + 1;

        String query2 = "update products set rating=" + newRating + ", reviews=" + newReviews + " where
product_Id=" + prodId + """;

        cobj.st.executeUpdate(query2);
    }

public void updateRetailerRating(String merchantId, int ratingIp) throws SQLException
{
    conn cobj = new conn();

    String query1 = "SELECT rating, reviews FROM merchant where Merchant_Id=" + merchantId + """;

    ResultSet rs = cobj.st.executeQuery(query1);
    rs.next();

    double rating = rs.getDouble("rating");
    int reviews = rs.getInt("reviews");

    double newRating;
    if(reviews==0) {newRating = ratingIp;}
    else {newRating = ((reviews*rating)+ratingIp)/(reviews+1);}

    int newReviews = reviews + 1;

    String query2 = "update merchant set rating=" + newRating + ", reviews=" + newReviews + " where
merchant_Id=" + merchantId + """;

    cobj.st.executeUpdate(query2);
}

public void deleteOrder(int orderId) throws SQLException
{
    conn cobj = new conn();

    String query = "DELETE FROM orders WHERE (`Order_Id` = " + orderId + ")";

    cobj.st.executeUpdate(query);
}
}

```

MyProducts.java

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.JTableHeader;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.awt.event.ActionEvent;
import java.awt.SystemColor;

public class MyProducts {

    private JFrame frame;
    private JTable table;
    private DefaultTableModel model;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    MyProducts window = new MyProducts();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public MyProducts() {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
```

```

        try {
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.getContentPane().setBackground(new Color(192, 192, 192));
    frame.setBounds(100, 100, 902, 566);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JPanel panel = new JPanel();
    panel.setBackground(SystemColor.desktop);
    panel.setBounds(0, 0, 888, 55);
    frame.getContentPane().add(panel);

    JLabel lblNewLabel = new JLabel("My Products");
    panel.add(lblNewLabel);
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 28));

    //Table start
    table = new JTable();
    table.setRowHeight(20);
    table.setFont(new Font("Tahoma", Font.PLAIN, 15));
    model = new DefaultTableModel(
        new Object[][] {

            },
        new String[] {
            "SNo.", "Product Id", "Product Name", "Price", "Rating", "Quantity"
        }
    );

    table.setModel(model);
    table.getColumnModel().getColumn(0).setPreferredWidth(4);
    table.getColumnModel().getColumn(1).setPreferredWidth(20);

    table.getColumnModel().getColumn(2).setPreferredWidth(200);

    table.getColumnModel().getColumn(3).setPreferredWidth(50);
    table.getColumnModel().getColumn(4).setPreferredWidth(50);
    table.getColumnModel().getColumn(5).setPreferredWidth(50);
    //table.setBounds(28, 77, 832, 364);
    //frame.getContentPane().add(table);

    JScrollPane scrollPane = new JScrollPane(table);

```

```
scrollPane.setBorder(new LineBorder(new Color(130, 135, 144), 3, true));
scrollPane.setFont(new Font("Tahoma", Font.BOLD, 15));
scrollPane.setBounds(28, 82, 832, 364);
```

```
JTableHeader Theader = table.getTableHeader();
Theader.setBackground(Color.GRAY);
Theader.setForeground(Color.BLACK);
Theader.setFont(new Font("Tahoma", Font.BOLD, 15));
frame.getContentPane().add(scrollPane);
```

```
loadProductsTable();
// Table end
```

```
JButton btnNewButton = new JButton("Back");
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
btnNewButton.setBounds(51, 468, 95, 31);
frame.getContentPane().add(btnNewButton);
```

```
//Back button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        new RetailerTab();
    }
});
```

```
}
```

```
private void loadProductsTable()
{
```

```
    conn cobj = new conn();
```

```
    String query = "SELECT * FROM products where Merchant_Id='" + Login.userId + "';";
    try
    {
```

```
        ResultSet rs = cobj.st.executeQuery(query);
```

```
        int index = 1;
        while(rs.next())
        {
```

```
            String prodId = rs.getString("Product_Id");
            String prodName = rs.getString("Name");
            String prodPrice = rs.getString("Price");
            String prodRating = rs.getString("Rating");
            String prodQuantity = rs.getString("Quantity");
```

```
            Object[] newRow = {index, prodId, prodName, prodPrice, prodRating, prodQuantity};
            model.addRow(newRow);
```

```
            index++;
```

```
        }
```

```
    }
```

```
    catch (SQLException e)
```

```
        {  
            e.printStackTrace();  
        }  
    }  
}
```

RetailerRegister.java

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JTextArea;
import javax.swing.JButton;
import javax.swing.JTextPane;
import javax.swing.JComboBox;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.awt.event.ActionEvent;
import javax.swing.JPanel;
import java.awt.SystemColor;

public class RetailerRegister {

    private JFrame frame;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JTextField textField_4;
    private JTextField textField_6;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    RetailerRegister window = new RetailerRegister();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public RetailerRegister() {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
```

```

        frame.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

});

initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 877, 504);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("Retailer Registration");
    lblNewLabel.setForeground(Color.BLACK);
    lblNewLabel.setBackground(Color.BLACK);
    lblNewLabel.setFont(new Font("Verdana", Font.BOLD, 22));
    lblNewLabel.setBounds(316, 11, 292, 48);
    frame.getContentPane().add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Name :");
    lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1.setBounds(102, 104, 66, 34);
    frame.getContentPane().add(lblNewLabel_1);

    textField = new JTextField();
    textField.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField.setBounds(193, 107, 227, 28);
    frame.getContentPane().add(textField);
    textField.setColumns(10);

    textField_1 = new JTextField();
    textField_1.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField_1.setColumns(10);
    textField_1.setBounds(193, 164, 227, 28);
    frame.getContentPane().add(textField_1);

    textField_2 = new JTextField();
    textField_2.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField_2.setColumns(10);
    textField_2.setBounds(193, 228, 227, 28);
    frame.getContentPane().add(textField_2);

    textField_3 = new JTextField();
    textField_3.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField_3.setColumns(10);
    textField_3.setBounds(193, 288, 227, 28);
    frame.getContentPane().add(textField_3);

    JLabel lblNewLabel_1_1 = new JLabel("Email :");

```

```
lblNewLabel_1_1.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_1.setBounds(102, 161, 66, 34);
frame.getContentPane().add(lblNewLabel_1_1);

JLabel lblNewLabel_1_2 = new JLabel("Phone :");
lblNewLabel_1_2.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_2.setBounds(91, 225, 77, 34);
frame.getContentPane().add(lblNewLabel_1_2);

JLabel lblNewLabel_1_3 = new JLabel("Password :");
lblNewLabel_1_3.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_3.setBounds(68, 285, 100, 34);
frame.getContentPane().add(lblNewLabel_1_3);

JLabel lblNewLabel_1_4 = new JLabel("Age :");
lblNewLabel_1_4.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_4.setBounds(494, 104, 66, 34);
frame.getContentPane().add(lblNewLabel_1_4);

JLabel lblNewLabel_1_5 = new JLabel("Address :");
lblNewLabel_1_5.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_5.setBounds(469, 164, 91, 34);
frame.getContentPane().add(lblNewLabel_1_5);

JLabel lblNewLabel_1_6 = new JLabel("PinCode :");
lblNewLabel_1_6.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1_6.setBounds(469, 285, 91, 34);
frame.getContentPane().add(lblNewLabel_1_6);

textField_4 = new JTextField();
textField_4.setFont(new Font("Tahoma", Font.PLAIN, 18));
textField_4.setColumns(10);
textField_4.setBounds(570, 104, 227, 28);
frame.getContentPane().add(textField_4);

textField_6 = new JTextField();
textField_6.setFont(new Font("Tahoma", Font.PLAIN, 18));
textField_6.setColumns(10);
textField_6.setBounds(570, 288, 227, 28);
frame.getContentPane().add(textField_6);

JTextArea textArea = new JTextArea();
textArea.setBounds(570, 164, 227, 99);
frame.getContentPane().add(textArea);

JButton btnNewButton = new JButton("Register");
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
btnNewButton.setBounds(570, 387, 123, 41);
frame.getContentPane().add(btnNewButton);

JButton btnBack = new JButton("Back");
btnBack.setFont(new Font("Tahoma", Font.BOLD, 18));
btnBack.setBounds(194, 387, 123, 41);
frame.getContentPane().add(btnBack);

JPanel panel = new JPanel();
```



```

panel.setBackground(SystemColor.desktop);
panel.setBounds(0, 0, 863, 74);
frame.getContentPane().add(panel);

//register
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try
        {
            String username = textField.getText();
            String password = textField_3.getText();
            String phone = textField_2.getText();
            String address = textArea.getText();
            String pincode = textField_6.getText();

            conn cobj = new conn();

            String query = "select * from merchant;";
            ResultSet rs = cobj.st.executeQuery(query);

            boolean flag = false;
            while(rs.next())
            {
                if(rs.getString("username").equals(username))
                {
                    JOptionPane.showMessageDialog(frame, "Username already
exists !", "Register", JOptionPane.ERROR_MESSAGE);
                    flag = true;
                    break;
                }
            }

            if(!flag)
            {
                String query1 = "select count(*) as count from merchant;";

                rs = cobj.st.executeQuery(query1);
                rs.next();
                int cnum = rs.getInt("count") + 1;
                String cust_id = "m" + Integer.toString(cnum);

                String query2 = "insert into merchant values('"+cust_id+"',
 '"+username+"', '"+password+"', 0,0,'"+ phone+"', '"+address+"', '"+pincode+"');";

                cobj.st.executeUpdate(query2);

                Wallet.createWallet(cust_id);

                JOptionPane.showMessageDialog(frame, "Registration
successful", "Register", JOptionPane.INFORMATION_MESSAGE);

                btnBack.doClick();
            }
        }
    }
}

```

```
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }

});

//back
btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.setVisible(false);
        Login.loginobj.frmShoppie.setVisible(true);
    }
});

}

}
```

RetailerTab.java

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JList;
import java.awt.Color;
import javax.swing.JTabbedPane;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.SystemColor;
import javax.swing.JPanel;
import javax.swing.ImageIcon;

public class RetailerTab {

    private JFrame frame;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    RetailerTab window = new RetailerTab();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public RetailerTab() {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });

        initialize();
    }
}
```

```

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.getContentPane().setBackground(new Color(128, 128, 128));
    frame.setBounds(100, 100, 941, 576);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    String uname = Login.username_ip.toUpperCase();
    JLabel lblNewLabel = new JLabel("Hello " + uname + " !");
    lblNewLabel.setBackground(Color.BLACK);
    lblNewLabel.setForeground(Color.BLACK);
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 28));
    lblNewLabel.setBounds(361, 11, 263, 54);
    frame.getContentPane().add(lblNewLabel);

    JButton btnNewButton = new JButton("Sell Items");
    btnNewButton.setBorderPainted(false);
    btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnNewButton.setForeground(Color.WHITE);
    btnNewButton.setBackground(Color.BLACK);
    btnNewButton.setBounds(382, 160, 171, 54);
    frame.getContentPane().add(btnNewButton);

    JButton btnMyOrders = new JButton("My Products");
    btnMyOrders.setBorderPainted(false);
    btnMyOrders.setForeground(Color.WHITE);
    btnMyOrders.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnMyOrders.setBackground(Color.BLACK);
    btnMyOrders.setBounds(133, 303, 171, 54);
    frame.getContentPane().add(btnMyOrders);

    JButton btnMyWallet = new JButton("My Wallet");
    btnMyWallet.setBorderPainted(false);
    btnMyWallet.setForeground(Color.WHITE);
    btnMyWallet.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnMyWallet.setBackground(Color.BLACK);
    btnMyWallet.setBounds(382, 303, 171, 54);
    frame.getContentPane().add(btnMyWallet);

    JButton btnBack = new JButton("Logout");
    btnBack.setBorderPainted(false);
    btnBack.setForeground(Color.WHITE);
    btnBack.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnBack.setBackground(Color.BLACK);
    btnBack.setBounds(391, 451, 162, 41);
    frame.getContentPane().add(btnBack);

    JButton btnMyInfo = new JButton("My Info");
    btnMyInfo.setBorderPainted(false);
    btnMyInfo.setForeground(Color.WHITE);
    btnMyInfo.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnMyInfo.setBackground(Color.BLACK);
    btnMyInfo.setBounds(630, 303, 171, 54);

```

```

frame.getContentPane().add(btnMyInfo);

JPanel panel = new JPanel();
panel.setBackground(new Color(255, 127, 80));
panel.setBounds(0, 0, 927, 83);
frame.getContentPane().add(panel);

JLabel lblNewLabel_1 = new JLabel("New label");
lblNewLabel_1.setIcon(new ImageIcon(RetailerTab.class.getResource("/icons/ecom.jpg")));
lblNewLabel_1.setBounds(0, 82, 927, 457);
frame.getContentPane().add(lblNewLabel_1);


//Logout button
btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        Login.loginobj.frmShoppie.setVisible(true);
    }
});

//My wallet Button
btnMyWallet.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        new Wallet();
    }
});

//Sell Items button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        new SellItems();
    }
});

//My products button
btnMyOrders.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        new MyProducts();
    }
});

//My Info button
btnMyInfo.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

```

```
frame.dispose();  
new MyInfo();
```

```
}
```

```
});
```

```
}
```

```
}
```

SellItems.java:

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.awt.event.ActionEvent;
import javax.swing.JSeparator;
import java.awt.Color;
import javax.swing.SwingConstants;
import java.awt.SystemColor;
import javax.swing.JPanel;

public class SellItems {

    private JFrame frame;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JTextField textField_4;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    SellItems window = new SellItems();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public SellItems() {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame.setVisible(true);
```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 955, 530);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("Sell Your Products");
    lblNewLabel.setBounds(368, 11, 243, 42);
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 25));
    frame.getContentPane().add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Product Name :");
    lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 19));
    lblNewLabel_1.setBounds(26, 121, 161, 31);
    frame.getContentPane().add(lblNewLabel_1);

    JLabel lblNewLabel_1_1 = new JLabel("Product Tags :");
    lblNewLabel_1_1.setFont(new Font("Tahoma", Font.BOLD, 19));
    lblNewLabel_1_1.setBounds(26, 198, 161, 31);
    frame.getContentPane().add(lblNewLabel_1_1);

    JLabel lblNewLabel_1_2 = new JLabel("Product Price :");
    lblNewLabel_1_2.setFont(new Font("Tahoma", Font.BOLD, 19));
    lblNewLabel_1_2.setBounds(587, 121, 150, 31);
    frame.getContentPane().add(lblNewLabel_1_2);

    JLabel lblNewLabel_1_3 = new JLabel("Product Quantity :");
    lblNewLabel_1_3.setFont(new Font("Tahoma", Font.BOLD, 19));
    lblNewLabel_1_3.setBounds(587, 217, 187, 31);
    frame.getContentPane().add(lblNewLabel_1_3);

    textField = new JTextField();
    textField.setFont(new Font("Tahoma", Font.PLAIN, 17));
    textField.setBounds(191, 122, 302, 31);
    frame.getContentPane().add(textField);
    textField.setColumns(10);

    textField_1 = new JTextField();
    textField_1.setFont(new Font("Tahoma", Font.PLAIN, 17));
    textField_1.setColumns(10);
    textField_1.setBounds(747, 122, 135, 31);
    frame.getContentPane().add(textField_1);

    textField_2 = new JTextField();

```



```
textField_2.setFont(new Font("Tahoma", Font.PLAIN, 17));
textField_2.setColumns(10);
textField_2.setBounds(191, 199, 214, 31);
frame.getContentPane().add(textField_2);
```

```
textField_3 = new JTextField();
textField_3.setFont(new Font("Tahoma", Font.PLAIN, 17));
textField_3.setColumns(10);
textField_3.setBounds(786, 218, 96, 31);
frame.getContentPane().add(textField_3);
```

```
JButton btnNewButton = new JButton("Add");
btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
btnNewButton.setBounds(426, 199, 67, 31);
frame.getContentPane().add(btnNewButton);
```

```
JButton btnBack = new JButton("Back");
btnBack.setFont(new Font("Tahoma", Font.BOLD, 20));
btnBack.setBounds(60, 413, 108, 42);
frame.getContentPane().add(btnBack);
```

```
JButton btnListMyProduct = new JButton("List my Product");
btnListMyProduct.setFont(new Font("Tahoma", Font.BOLD, 20));
btnListMyProduct.setBounds(657, 413, 208, 42);
frame.getContentPane().add(btnListMyProduct);
```

```
JSeparator separator = new JSeparator();
separator.setOrientation(SwingConstants.VERTICAL);
separator.setBounds(542, 103, 35, 260);
frame.getContentPane().add(separator);
```

```
textField_4 = new JTextField();
textField_4.setFont(new Font("Tahoma", Font.ITALIC, 15));
textField_4.setBackground(SystemColor.inactiveCaption);
textField_4.setEditable(false);
textField_4.setBounds(26, 255, 467, 31);
frame.getContentPane().add(textField_4);
textField_4.setColumns(10);
```

```
JButton btnClear = new JButton("Clear");
btnClear.setFont(new Font("Tahoma", Font.BOLD, 17));
btnClear.setBounds(229, 297, 82, 29);
frame.getContentPane().add(btnClear);
```

```
JPanel panel = new JPanel();
panel.setBackground(SystemColor.desktop);
panel.setBounds(0, 1, 941, 73);
frame.getContentPane().add(panel);
```

```
//back button
btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
    }
});
```

```

        new RetailerTab();
    }
});

//add tag button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        String newTag = textField_2.getText();
        String Tags = textField_4.getText();
        if(newTag.equals(""))
        {
            JOptionPane.showMessageDialog(frame, "Add a valid tag !", "Sell"
,JOptionPane.INFORMATION_MESSAGE);
        }
        else
        {
            if(Tags.equals("")) {textField_4.setText(newTag);}

            else {textField_4.setText(Tags + "/" + newTag);}
        }

        textField_2.setText("");
    }
});

//list my product button
btnListMyProduct.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try
        {
            String prodName = textField.getText();
            int prodPrice = Integer.parseInt(textField_1.getText());
            String prodTags = textField_4.getText();
            int prodQuantity = Integer.parseInt(textField_3.getText());

            addToProducts(prodName, prodPrice, prodTags, prodQuantity);

            JOptionPane.showMessageDialog(frame, prodName + " (" +prodQuantity+")
successfully listed","Sell",JOptionPane.INFORMATION_MESSAGE);
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(frame, "Invalid Text Fields !", "Sell"
,JOptionPane.INFORMATION_MESSAGE);
        }
    }
});

```

```

//clear button
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        textField_4.setText("");
    }
});
}

```

```

public void addToProducts(String prodName, int prodPrice, String prodTags, int prodQuantity)
{
    try
    {
        conn cobj = new conn();
        String query1, query2;

        query1 = "select count(*) as count from products;";
        ResultSet rs ;

        rs = cobj.st.executeQuery(query1);
        rs.next();
        int count = rs.getInt("count") + 1;
        String prodId = "p" + Integer.toString(count);

        double initialRating = 0;
        int reviews = 0;
        String retailerId = Login.userId;

        query2 = "insert into products values('" + prodId + "', '" + prodName + "', '" + prodTags + "', '" +
prodPrice
        + "', '" + initialRating + "', '" + prodQuantity + "', '" + retailerId + "', '" + reviews + "');";

        cobj.st.executeUpdate(query2);
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}
}

```

Shop.java:

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JSpinner;
import java.awt.BorderLayout;
import java.awt.Color;

import javax.swing.JTable;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;

import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.JTableHeader;

import javax.swing.border.SoftBevelBorder;
import javax.swing.border.BevelBorder;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;
import java.awt.event.ActionEvent;
import javax.swing.border.LineBorder;
import java.lang.Exception;
import javax.swing.JPanel;
import java.awt.SystemColor;

public class Shop {

    private JFrame frame;
    private JTextField textField;
    private JTable table;
    private JTextField textField_1;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Shop window = new Shop();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        })
    }
}
```

```

        });
    }

    /**
     * Create the application.
     */
    public Shop()
    {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });

        initialize();
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        frame = new JFrame();
        frame.getContentPane().setFont(new Font("Tahoma", Font.PLAIN, 15));
        frame.setBounds(100, 100, 788, 635);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);

        JLabel lblNewLabel = new JLabel("Search :");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 20));
        lblNewLabel.setBounds(35, 85, 94, 35);
        frame.getContentPane().add(lblNewLabel);

        textField = new JTextField();
        textField.setFont(new Font("Tahoma", Font.PLAIN, 15));
        textField.setBounds(132, 89, 412, 31);
        frame.getContentPane().add(textField);
        textField.setColumns(10);

        JButton btnNewButton = new JButton("Search");
        btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 15));
        btnNewButton.setBounds(575, 89, 89, 31);
        frame.getContentPane().add(btnNewButton);

        table = new JTable();
        table.setRowHeight(20);
        table.setFont(new Font("Tahoma", Font.PLAIN, 15));
        DefaultTableModel model = new DefaultTableModel(
            new Object[][] {

            },
            new String[] {
                "Product_Id", "Name", "Price", "Rating", "Quantity"
            }
        );
    }

```

```

    }
);
table.setModel(model);
table.getColumnModel().getColumn(1).setPreferredWidth(189);
table.getColumnModel().getColumn(2).setPreferredWidth(67);
table.getColumnModel().getColumn(3).setPreferredWidth(49);
table.getColumnModel().getColumn(4).setPreferredWidth(56);
//table.setBounds(35, 94, 693, 353);
//frame.getContentPane().add(table);

JScrollPane scrollPane = new JScrollPane(table);
scrollPane.setBorder(new LineBorder(new Color(130, 135, 144), 3, true));
scrollPane.setFont(new Font("Tahoma", Font.BOLD, 15));
scrollPane.setBounds(35, 141, 695, 386);

JTableHeader Theader = table.getTableHeader();
Theader.setBackground(Color.GRAY);
Theader.setForeground(Color.BLACK);
Theader.setFont(new Font("Tahoma", Font.BOLD, 15));
frame.getContentPane().add(scrollPane);

JButton btnNewButton_1 = new JButton("Buy");
btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD, 20));
btnNewButton_1.setBounds(641, 551, 89, 35);
frame.getContentPane().add(btnNewButton_1);

JLabel lblNewLabel_1 = new JLabel("Quantity :");
lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel_1.setBounds(202, 552, 100, 35);
frame.getContentPane().add(lblNewLabel_1);

textField_1 = new JTextField();
textField_1.setFont(new Font("Tahoma", Font.PLAIN, 18));
textField_1.setBounds(300, 554, 55, 31);
frame.getContentPane().add(textField_1);
textField_1.setColumns(10);

JButton btnNewButton_2 = new JButton("Back");
btnNewButton_2.setFont(new Font("Tahoma", Font.BOLD, 18));
btnNewButton_2.setBounds(35, 552, 100, 35);
frame.getContentPane().add(btnNewButton_2);

JButton btnNewButton_1_1 = new JButton("Add To Cart");
btnNewButton_1_1.setFont(new Font("Tahoma", Font.BOLD, 17));
btnNewButton_1_1.setBounds(479, 552, 135, 35);
frame.getContentPane().add(btnNewButton_1_1);

JLabel lblShop = new JLabel("Shop");
lblShop.setFont(new Font("Tahoma", Font.BOLD, 25));
lblShop.setBounds(337, 11, 94, 35);
frame.getContentPane().add(lblShop);

JPanel panel = new JPanel();
panel.setBackground(SystemColor.desktop);
panel.setBounds(0, 0, 774, 63);

```

```
frame.getContentPane().add(panel);
```

```
//Search button
```

```
btnNewButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {
```

```
        int numRows = model.getRowCount();  
        for(int i=numRows-1; i>=0; i--)  
        {  
            model.removeRow(i);  
        }
```

```
        String search = textField.getText().toLowerCase();  
        String[] searchTags = search.split(" ");
```

```
        conn cobj = new conn();
```

```
        String query = "SELECT * FROM products;";  
        try  
        {
```

```
            ResultSet rs = cobj.st.executeQuery(query);
```

```
            while(rs.next())  
            {
```

```
                String products = rs.getString("Tag").toLowerCase();  
                String[] productTags = products.split("/");
```

```
                int flag = 0;  
                for(String s1: searchTags)  
                {  
                    for(String s2: productTags)
```

```
                    {  
                        if(s1.equals(s2))
```

```
                        {  
                            String prod_Id = rs.getString("Product_Id");  
                            String prod_Name = rs.getString("Name");  
                            String prod_Price = rs.getString("Price");  
                            String prod_Rating = rs.getString("Rating");  
                            String prod_Quantity = rs.getString("Quantity");
```

```
                            Object[] newRow = {prod_Id, prod_Name,
```

```
                                prod_Price, prod_Rating, prod_Quantity};  
                            model.addRow(newRow);
```

```
                            flag = 1; break;
```

```
                        }
```

```
                    }
```

```
                }  
                if(flag == 1) break;
```

```
            }
```

```
        }
```

```
    }  
    catch (SQLException e)  
    {
```

```
        e.printStackTrace();
```

```
prod_Price, prod_Rating, prod_Quantity};
```

```

    }

    }

});

//Buy button
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        int orderQuantity=0;
        int prod_Price=0;
        String prod_Id=null;
        String prod_Name=null;
        String prod_Pricestr=null;
        int prod_Quantity=0;

        try
        {
            try
            {
                prod_Id = (String) model.getValueAt(table.getSelectedRow(), 0);
                prod_Name = (String) model.getValueAt(table.getSelectedRow(), 1);
                prod_Pricestr = (String) model.getValueAt(table.getSelectedRow(), 2);
                prod_Quantity = Integer.parseInt((String)
model.getValueAt(table.getSelectedRow(), 4));
                prod_Price = Integer.parseInt(prod_Pricestr);
            }
            catch(Exception e)
            {
                throw new MyException.ProductNotSelected("");
            }

            try{orderQuantity = Integer.parseInt(textField_1.getText());}
            catch(Exception e) {throw new MyException.Invalid_tf_shop("");}

            if(orderQuantity==0) {throw new MyException.Invalid_tf_shop("");}

            int totalPrice = prod_Price*orderQuantity;
            int currBalance = Wallet.checkBalance(Login.userId);

            //successful purchase
            if((prod_Quantity>=orderQuantity) && (currBalance >= totalPrice))
            {
                Wallet.updateBalance(currBalance - totalPrice);
                Wallet.addToRetailerBalance(getRetailerId(prod_Id), totalPrice);
                updateQuantity(prod_Id, prod_Quantity - orderQuantity);
                addToOrders(Login.userId, prod_Id, orderQuantity);
                JOptionPane.showMessageDialog(frame, prod_Name + "
("+orderQuantity+") Successfully purchased", "Purchase", JOptionPane.INFORMATION_MESSAGE);
                btnNewButton.doClick();          //refreshing content
            }
        }
    }
});

```



```

        else if(currBalance < totalPrice)
        {
            JOptionPane.showMessageDialog(frame, "Insufficient funds in wallet
!\nWallet balance : "+currBalance, "Purchase",JOptionPane.INFORMATION_MESSAGE);
        }
        else
        {
            JOptionPane.showMessageDialog(frame, "Products unavailable
currently !", "Purchase" ,JOptionPane.INFORMATION_MESSAGE);
        }

    }
    catch(MyException.Invalid_tf_shop e1)
    {
        JOptionPane.showMessageDialog(frame, "Enter valid Quantity
!", "Purchase",JOptionPane.INFORMATION_MESSAGE);
    }
    catch(MyException.ProductNotSelected e2)
    {
        JOptionPane.showMessageDialog(frame, "First select a product to buy !",
"Purchase" ,JOptionPane.INFORMATION_MESSAGE);
    }

    }

});

//back button
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        new CustomerTab();
    }
});

//add to cart button
btnNewButton_1_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        String prodId = (String) model.getValueAt(table.getSelectedRow(), 0);
        String prodName = (String) model.getValueAt(table.getSelectedRow(), 1);

        conn cobj = new conn();

        String query1 = "SELECT products FROM cart where Customer_Id=" + Login.userId +
"";

        try
        {
            ResultSet rs = cobj.st.executeQuery(query1);
            rs.next();

            String prodStr = rs.getString("products");

```

```

        List<String> products = new LinkedList(Arrays.asList(prodStr.split(",")));

        products.add(prodId);
        String newProdStr = String.join(",", products);

        String query2 = "update cart set products='" + newProdStr + "' where
Customer_Id='" + Login.userId + "';";

        cobj.st.executeUpdate(query2);
    }
    catch(SQLException e)
    {
        e.printStackTrace();
        String query3 = "insert into cart values('" + Login.userId + "','" + prodId + "');";

        try{cobj.st.executeUpdate(query3);}
        catch(SQLException se) {se.printStackTrace();}
    }

    JOptionPane.showMessageDialog(frame, prodName + " added to cart !","Add to
Cart",JOptionPane.INFORMATION_MESSAGE);
    }

    });
}

public static void updateQuantity(String prod_Id, int newQuantity)
{

    conn cobj = new conn();
    String query;

    query = "update products set quantity=" + newQuantity + " where Product_Id='" + prod_Id + "';";

    try
    {
        cobj.st.executeUpdate(query);
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}

public static void addToOrders(String custId, String prodId, int quantity)
{
    try
    {
        conn cobj = new conn();
        String query1, query2;

        query1 = "select count(*) as count from orders;";
        ResultSet rs ;

        rs = cobj.st.executeQuery(query1);
    }
}

```

```

        rs.next();
        int cnum = rs.getInt("count") + 1;
        //String order_Id = "o" + Integer.toString(cnum);
        int order_Id = cnum;

        query2 = "insert into orders values(" + order_Id + "," + custId + "," + prodId + "," + quantity +
");";

        cobj.st.executeUpdate(query2);
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}

public static String getRetailerId(String prodId)
{
    String RetailerId = "";
    try
    {
        conn cobj = new conn();

        String query = "select merchant_Id from products where Product_Id=" + prodId + ";";

        ResultSet rs = cobj.st.executeQuery(query);
        rs.next();

        RetailerId = rs.getString("merchant_Id");

    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }

    return RetailerId;
}
}

```

Test Login.java:

```
import java.sql.*;
import java.util.Scanner;

public class Test_Login
{
    Test_Login() throws SQLException
    {
        conn cobj = new conn();

        String query1 = "select UserName from customer where customer_Id = \"c1\"";
        String query2 = "insert into customer values(\"c45\", \"suhail\", \"suha\")";
        String query3 = "SELECT * FROM customer";

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Username : ");
        String uname_ip = sc.nextLine();
        System.out.print("Enter Password : ");
        String pass_ip = sc.nextLine();

        ResultSet rs = cobj.st.executeQuery(query3);

        Boolean access = false;
        while(rs.next())
        {
            String uname = rs.getString("username");
            String pass = rs.getString("password");
            if(uname.equals(uname_ip) && pass.equals(pass_ip))
            {
                access = true;
                break;
            }
        }

        if(access)
        {
            System.out.println("Access Granted!");
        }
        else
        {
            System.out.println("Access Denied");
        }
    }

    public static void main(String[] args) throws SQLException
    {
        new Test_Login();
    }
}
```

Wallet.java:

```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JPanel;
import java.awt.SystemColor;

public class Wallet {

    private JFrame frame;
    private JTextField textField;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Wallet window = new Wallet();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public Wallet()
    {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

        initialize();
    }

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 548, 353);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("My Wallet");
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 25));
    lblNewLabel.setBounds(185, 11, 139, 45);
    frame.getContentPane().add(lblNewLabel);

    String userName = Login.username_ip;
    JLabel lblNewLabel_1 = new JLabel("User Name : " + userName);
    lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1.setBounds(59, 79, 372, 36);
    frame.getContentPane().add(lblNewLabel_1);

    //Balance label
    String balance = checkBalance(Login.userId) + "";
    JLabel lblNewLabel_1_1 = new JLabel("Wallet Amount : " + balance);
    lblNewLabel_1_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1_1.setBounds(57, 136, 374, 36);
    frame.getContentPane().add(lblNewLabel_1_1);

    JLabel lblNewLabel_1_2 = new JLabel("Add Amount :");
    lblNewLabel_1_2.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel_1_2.setBounds(59, 195, 139, 36);
    frame.getContentPane().add(lblNewLabel_1_2);

    textField = new JTextField();
    textField.setFont(new Font("Tahoma", Font.PLAIN, 18));
    textField.setBounds(218, 195, 123, 36);
    frame.getContentPane().add(textField);
    textField.setColumns(10);

    JButton btnNewButton = new JButton("ADD");
    btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnNewButton.setBounds(377, 196, 85, 36);
    frame.getContentPane().add(btnNewButton);

    JButton btnNewButton_1 = new JButton("Back");
    btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD, 18));
    btnNewButton_1.setBounds(59, 264, 94, 36);
    frame.getContentPane().add(btnNewButton_1);

    JPanel panel = new JPanel();
    panel.setBackground(SystemColor.desktop);
    panel.setBounds(0, 0, 534, 66);

```

```

frame.getContentPane().add(panel);

//back button
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        frame.dispose();
        if(Login.userId.charAt(0)=='c')
        {
            new CustomerTab();
        }
        else
        {
            new RetailerTab();
        }
    }
});

//add balance button
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        String userId = Login.userId;
        int balance = checkBalance(userId);
        int newBalance = 0;

        try
        {
            int addAmount = Integer.parseInt(textField.getText());
            newBalance = balance + addAmount;
            conn cobj = new conn();
            String query = "update wallet set balance = " + newBalance + " where
wallet_Id='"+userId+"'";

            try { cobj.st.executeUpdate(query);}
            catch (SQLException e) {e.printStackTrace();}

            lblNewLabel_1_1.setText("Wallet Amount : " + newBalance);

            JOptionPane.showMessageDialog(frame, addAmount + " Successfully
added","Balance",JOptionPane.INFORMATION_MESSAGE);
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(frame, "Enter a valid amount
!", "Balance",JOptionPane.ERROR_MESSAGE);
        }
    }
});
}

```

```

public static void createWallet(String walletId)
{
    conn cobj = new conn();

    String query = "insert into wallet values('" + walletId + "'," + 0 + ")";

    try
    {
        cobj.st.executeUpdate(query);
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}

```

```

public static int checkBalance(String userId)
{
    conn cobj = new conn();

    String query = "SELECT * FROM Wallet;";
    //String userId = Login.userId;
    int balance = 0;
    try
    {
        ResultSet rs = cobj.st.executeQuery(query);

        while(rs.next())
        {
            String Id = rs.getString("Wallet_Id");

            if(userId.equalsIgnoreCase(Id))
            {
                balance = rs.getInt("Balance");
                return balance;
            }
        }

    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }

    return balance;
}

```

```

//update balance
public static void updateBalance(int newBalance)
{

```



```

        conn cobj = new conn();

        String userId = Login.userId;
        String query = "update wallet set Balance=" + newBalance + " where Wallet_Id = " + userId + """;

        try
        {
            cobj.st.executeUpdate(query);
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
    }

```

//update retailer wallet

```

public static void addToRetailerBalance(String userId, int balance)

```

```

{
    conn cobj = new conn();

    int newBalance = balance + checkBalance(userId);
    String query = "update wallet set Balance=" + newBalance + " where Wallet_Id = " + userId + """;

    try
    {
        cobj.st.executeUpdate(query);
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}
}

```

Invalid_tf_shop.java:

```
package MyException;

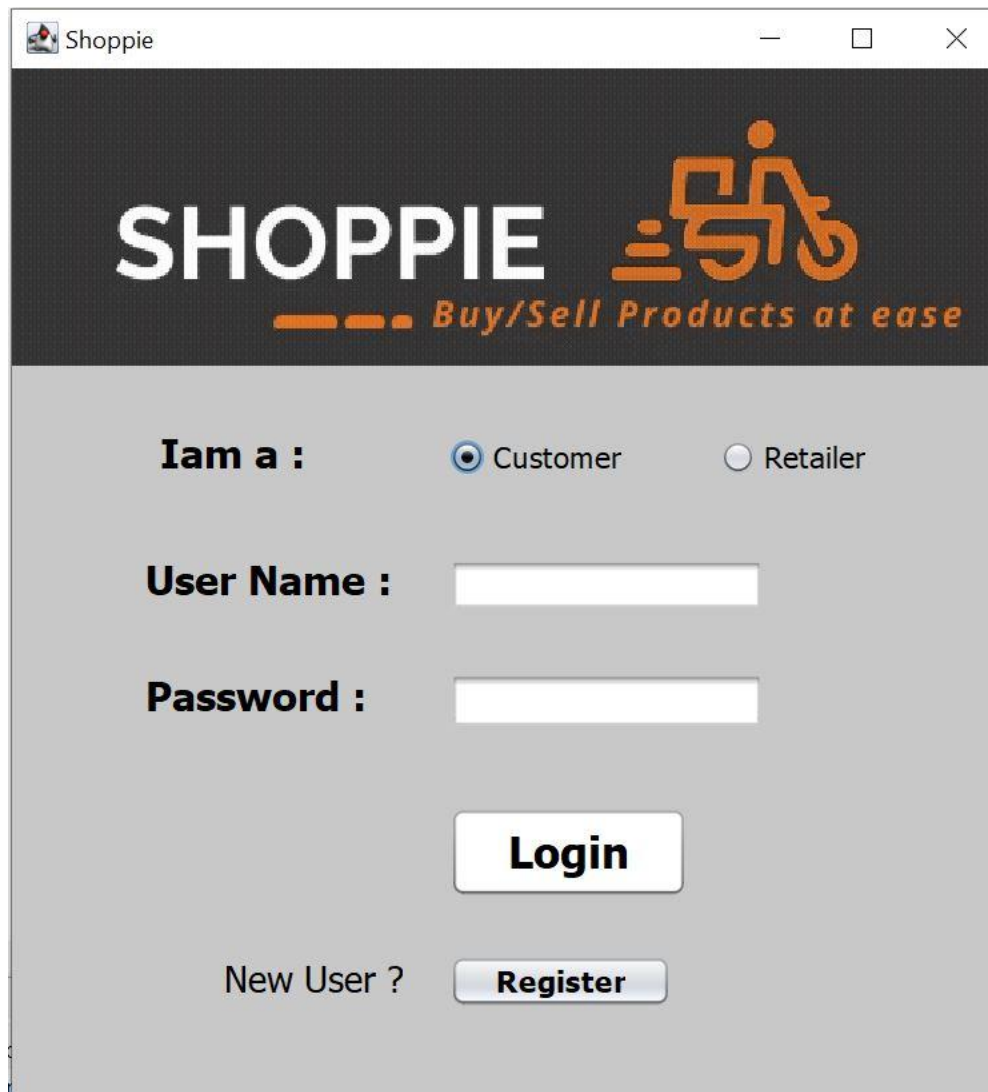
public class Invalid_tf_shop extends Exception
{
    public Invalid_tf_shop(String s)
    {
        super(s);
    }
}
```

ProductNotSelected.java:


```
package MyException;

public class ProductNotSelected extends Exception
{
    public ProductNotSelected(String s)
    {
        super(s);
    }
}
```

SCREENSHOTS:



The screenshot shows a web browser window titled "Shoppie". The header features the "SHOPPIE" logo in white on a dark background, accompanied by an orange icon of a person on a motorcycle. Below the logo is the tagline "Buy/Sell Products at ease" in orange. The main content area is light gray and contains a login and registration form. The form includes a "I am a :" label with two radio buttons: "Customer" (selected) and "Retailer". Below this are input fields for "User Name :" and "Password :". A "Login" button is positioned below the password field. At the bottom, there is a "New User ?" label and a "Register" button.

SHOPPIE 
Buy/Sell Products at ease

I am a : ☒ Customer ☐ Retailer

User Name :

Password :

Login

New User ? **Register**



Customer Registration

Name : **Age :**

Email : **Address :**

Phone :

Password : **PinCode :**

Back

Register



Retailer Registration

Name : **Age :**

Email : **Address :**

Phone :

Password : **PinCode :**

Back

Register



Hello SIDDHARTH !

Shop

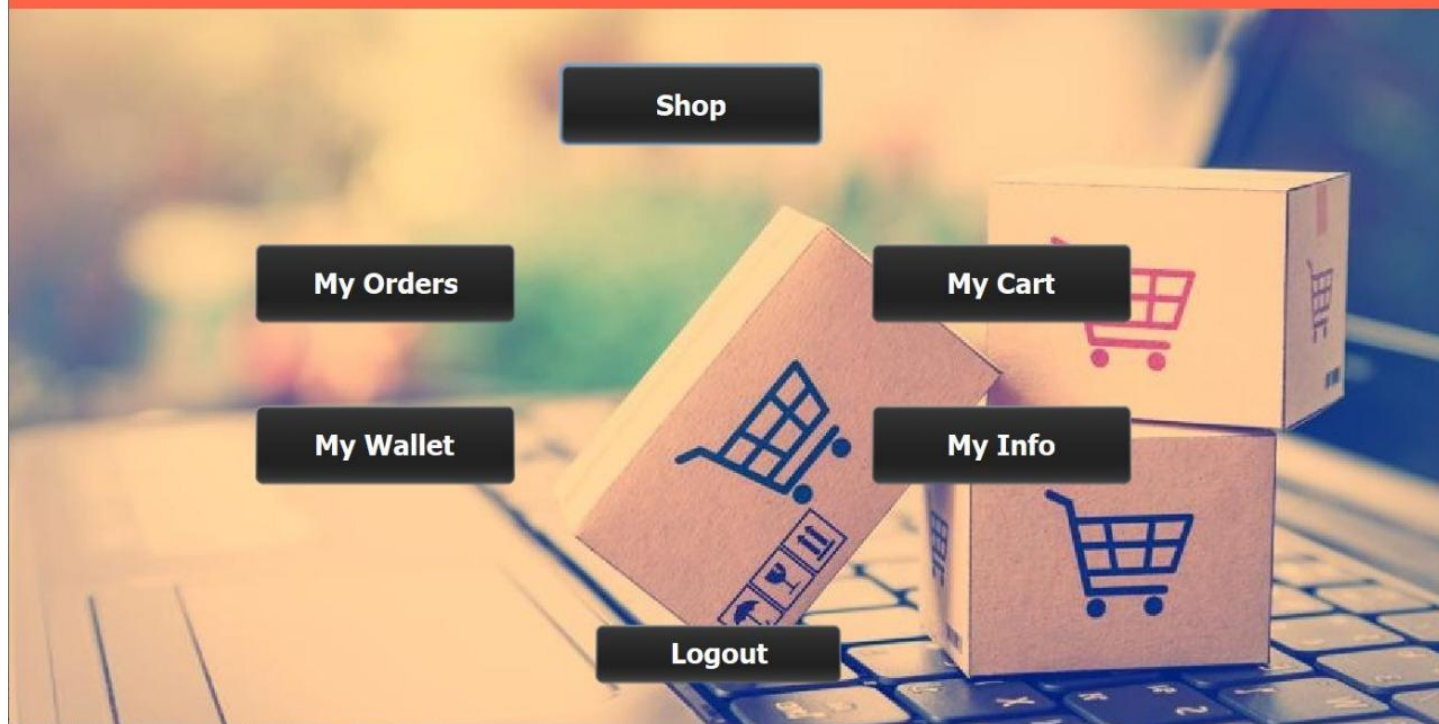
My Orders

My Cart

My Wallet

My Info

Logout





Shop

Search :

Search

Product_Id	Name	Price	Rating	Quantity
p1	apple iphone X	80000	4.6	4
p2	samsung 7s phone	60000	5.0	7
p9	google pixel v2	90000	2.0	1

Back

Quantity :

Add To Cart

Buy

Shop

Search :

Search

Product_Id	Name	Price	Rating	Quantity
p1	apple iphone X	80000	4.6	4
p2	samsung 7s phone	60000	5.0	7
p9	google pixel v2	90000	2.0	1

Add to Cart



apple iphone X added to cart !

OK

Back

Quantity :

Add To Cart

Buy



Shop

Search :

Search

Product_Id	Name	Price	Rating	Quantity
p3	hp pavillion x360 laptop	85000	4.0	2
p8	lenovo q45 laptop	45000	2.0	5

Purchase



Insufficient funds in wallet !

Wallet balance : 10000

OK

Back

Quantity :

Add To Cart

Buy



Shop

Search :

Search

Product_Id	Name	Price	Rating	Quantity
p3	hp pavillion x360 laptop	85000	4.0	2
p8	lenovo q45 laptop	45000	2.0	5

Purchase



lenovo q45 laptop (1) Successfully purchased

OK

Back

Quantity :

Add To Cart

Buy



My Orders

SNo.	Order Number	Produ...	Product Name	Price	Quantity
1	4	p9	google pixel v2	90000	1
2	3	p6	logitech mouse a200	650	1
3	2	p7	american tourister a20 bag	2200	1
4	1	p8	lenovo q45 laptop	45000	1

Back

Rating : 1

Submit



My Orders

SNo.	Order Number	Produ...	Product Name	Price	Quantity
2	3	p6	logitech mouse a200	650	1
3	2	p7	american tourister a20 bag	2200	1
4	1	p8	lenovo q45 laptop	45000	1

Rating



Thanks you for your feedback !

OK

Back

Rating : 4

Submit



My Cart

SNo.	Product Id	Product Name	Price	Rating	Quantity
1	p1	apple iphone X	80000	4.6	4
2	p7	american tourister a20 bag	2200	5.0	0
3	p6	logitech mouse a200	650	3.333333333...	22
4	p9	google pixel v2	90000	4.0	0
5	p5	axe intense perfume	600	4.0	0

Back

Quantity :

Buy



My Cart

SNo.	Product Id	Product Name	Price	Rating	Quantity
1	p1	apple iphone X	80000	4.6	4
2	p7	american tourister a20 bag	2200	5.0	0
4	p9	google pixel v2	90000	4.0	0
5	p5	axe intense perfume	600	4.0	0

Purchase



logitech mouse a200 (1) Successfully purchased

OK

Back

Quantity :

Buy



My Wallet

User Name : siddharth

Wallet Amount : 11500

Add Amount :

ADD

Back



My Info

Name :

siddharth

Password :

sidd

Phone :

9377365245

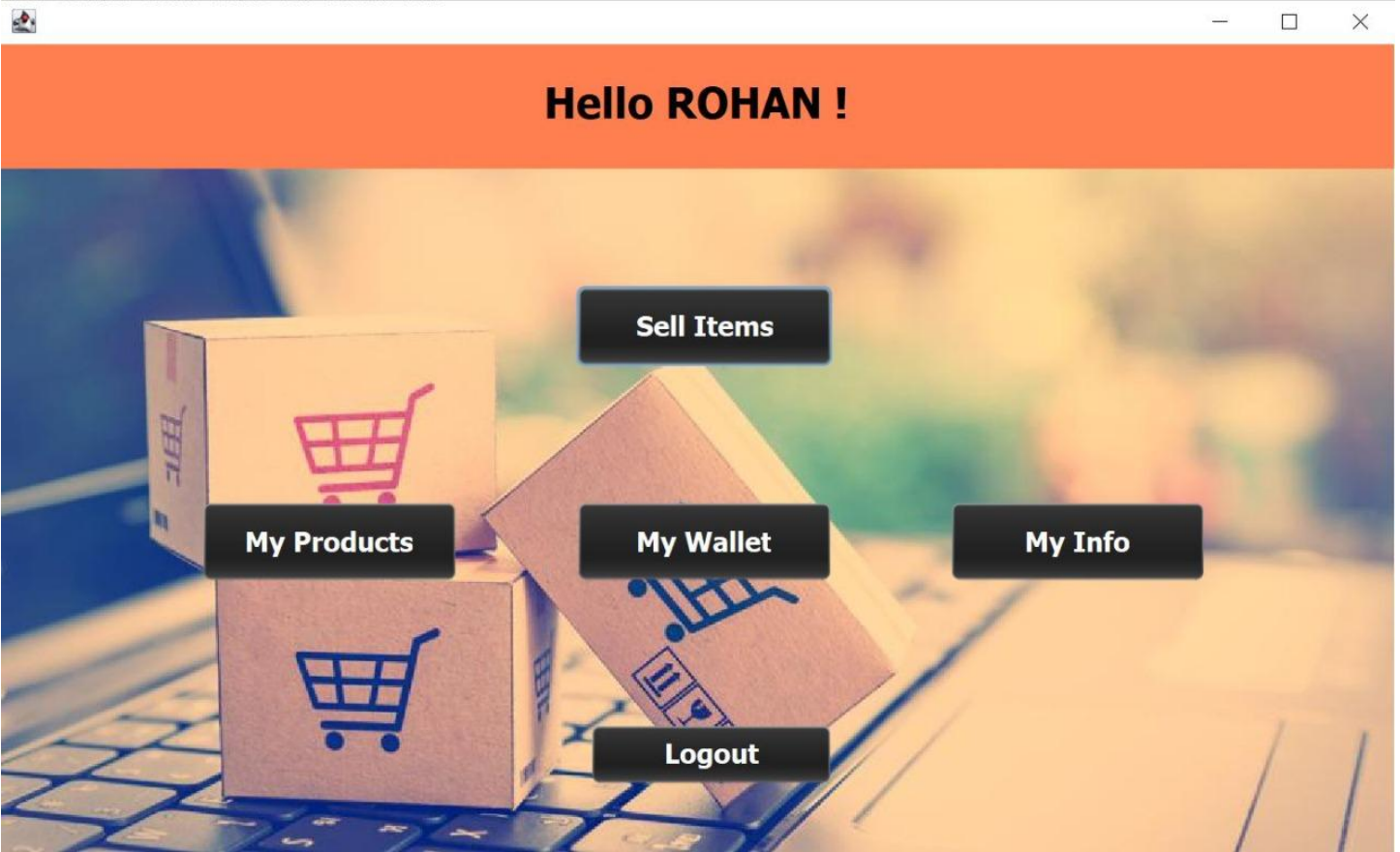
Address :

hno 1-838-38, mes colony, alwal,
hyderabad

Pincode :

500015

Back

A form titled "Sell Your Products" with a teal header. The form is divided into two main sections by a vertical line. The left section contains a "Product Name" field with the value "bluestar a400 air conditioner", a "Product Tags" field with the value "appliances" and an "Add" button, and a text area containing "air conditioner/bluestar/electronics" with a "Clear" button below it. The right section contains a "Product Price" field with the value "45000" and a "Product Quantity" field with the value "15". At the bottom of the form are two buttons: "Back" on the left and "List my Product" on the right.

Sell Your Products

Product Name :

bluestar a400 air conditioner

Product Price :

45000

Product Tags :

Add

air conditioner/bluestar/electronics/appliances

Clear

Back

Sell

i

bluestar a400 air conditioner (15) successfully listed

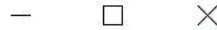
OK

List my Product

My Products

SNo.	Product Id	Product Name	Price	Rating	Quantity
1	p12	sony bravia kwl500	120000	4.0	4
2	p13	bluestar k400 AC	45000	0.0	15

Back



My Wallet

User Name : rohan

Wallet Amount : 120000

Add Amount :

ADD

Back



My Info

Name :

rohan

Password :

roha

Phone :

9374324922

Pincode :

937498

Address : hno 30-90-98, papaipet, warangal

Back

CONCLUSION:

In conclusion, shoppie is an E-commerce application which serves two purposes ie. buying and selling. With this application, Customers can buy and sell their products online in an easy way.

Shoppie is presetly a desktop application which uses MySQL as the database.

Additional changes to the application in the future versions will be as follows:

1. The application will be completely hosted onto the web so that it serves its real purpose.
2. The transactions will become more secure to gain the trust of customers.
3. Many more internal features will be incorporated.