

# **SHOPPING MANAGEMENT SYSTEM**

## **Description about module->**

Our project aims on developing an application for a shopping forum. Our project contains 12 modules.

The front module(first.java) allows the user to select whether he/she is the customer or admin.

In the admin module(admin\_login.java), he/she has to enter username and password which is initially hard-coded in the database.

The admin option module(a\_op.java) allows the admin to view orders placed and manage customers.

The remove customer module(remove.java) removes the customer selected by the admin from the shopping forum.

In the customer module(cust\_login.java), the customer can enter his/her username and password to begin his shopping experience.

In the customer register module(cust\_reg.java), he/she can register themselves to become a member of the shopping forum.

In the customer option module(c\_op.java), he/she can navigate between various modules that allow customers to select the category and shop in that category, view and manage cart, view orders and make payment.

The mens module(men.java) displays various products and its details in the men category and allows the customer to add a particular product to his/her cart.

The women module(women.java) displays various products and its details in the women category and allows the customer to add a particular product to his/her cart.

The kids module(kids.java) displays various products and its details in the kids category and allows the customer to add a particular product to his/her cart.

The cart module(cart.java) displays the products added to cart by the customers and total number of products and total price. The cart can be modified by the customer where products can be removed and can place orders.

The orders module(order.java) displays the orders placed by the customer and allows the customer to pay by entering the account number.

Our project is useful for easy user interface and can implement the same.

**SHOPPINGRAM(** username, pw, fname, lname, ph\_no, email, addr, country, district, state, Orderno, no\_of\_pdt, total\_pdt, date, status, pid, pname, avail, category, price, aid, apw)**)**

### **Functional Dependencies:**

Fd1: Username  $\rightarrow$  pw , fname, Lname, Phno, E\_mail , Address , Country ,District , state ,pid

Fd2: pid  $\rightarrow$  pname, avail, category, price

Fd3: Orderno  $\rightarrow$  User, no\_of\_products, totalprice , date, status

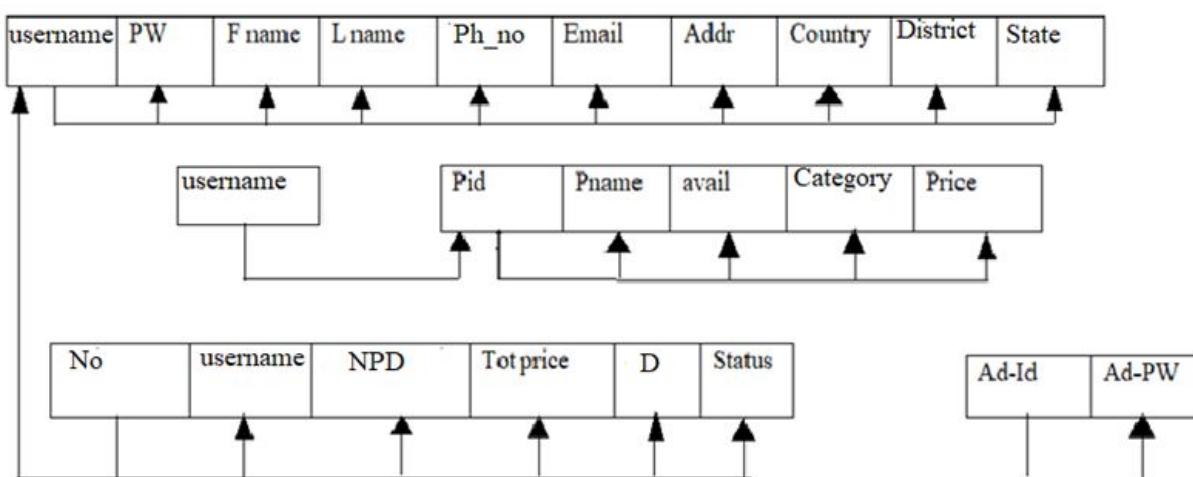
Fd4: ad\_id  $\rightarrow$  ad\_pw

{username,ad\_id} is the superkey.

### **Normalisation:**

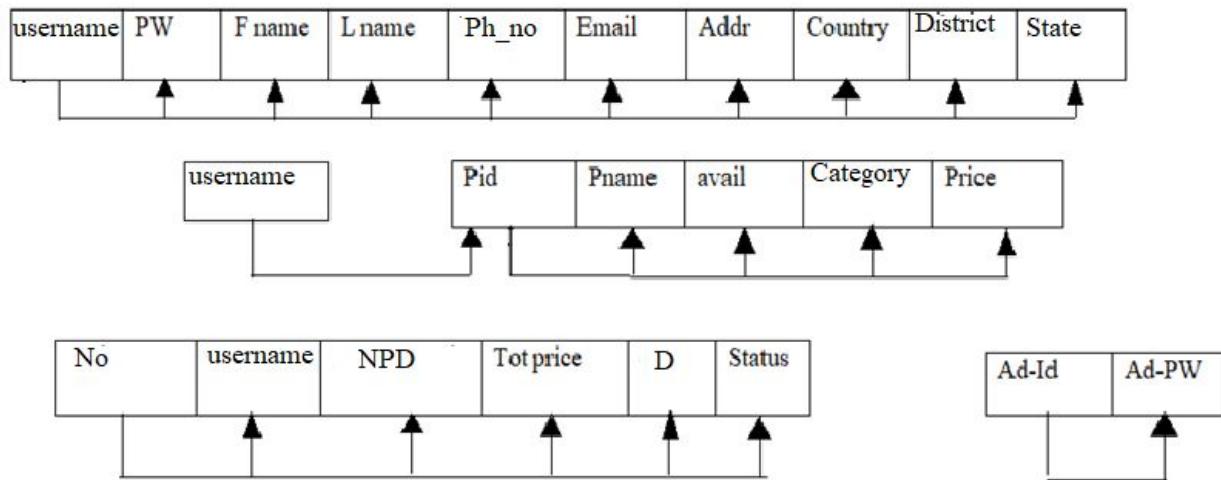
Decomposing the above relation into their first normal form

#### **1NF:**



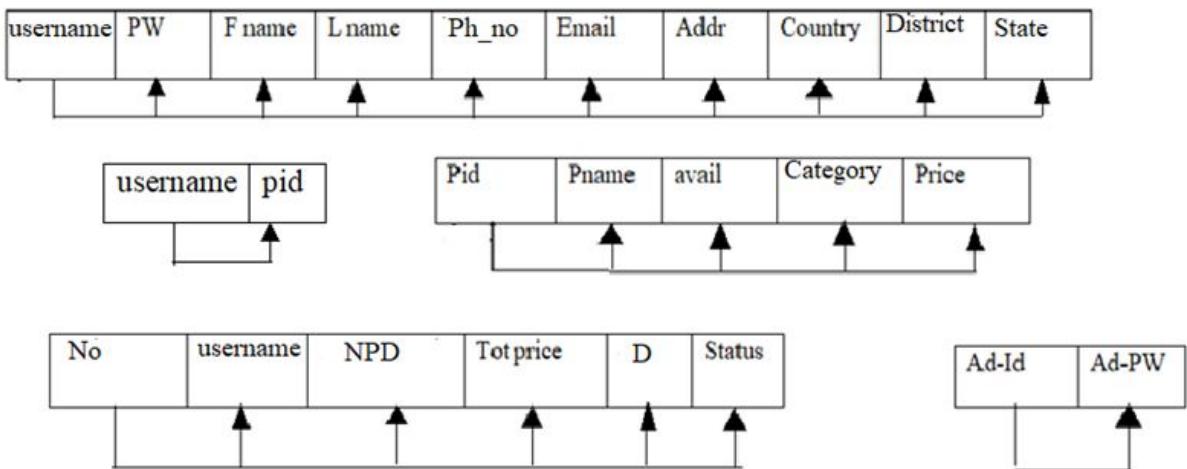
**Decomposing the above relation into their second normal form**

**2NF:**

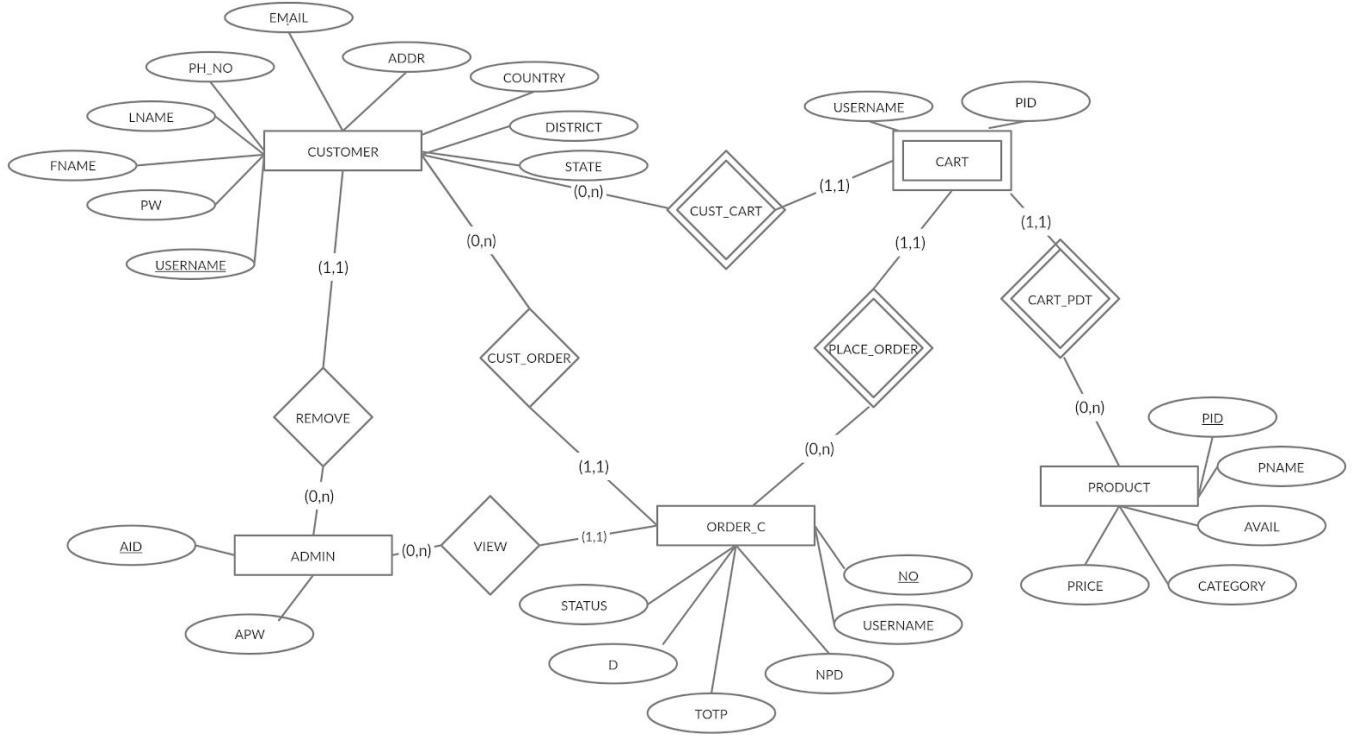


**Decomposing the above relation into their third normal form**

**3NF:**



## ER DIAGRAM



The relation remove, place\_order and view cannot be converted into relation as they have no common attributes and do not have the foreign key approach.

## ER TO RELATION DIAGRAM:

### Customer:

Username	PW	F name	L name	Ph_no	Email	Addr	Country	District	State
----------	----	--------	--------	-------	-------	------	---------	----------	-------

### Product:

Pid	Pname	avail	Category	Price
-----	-------	-------	----------	-------

### Cart:

Username	Pid
----------	-----

### Order c:

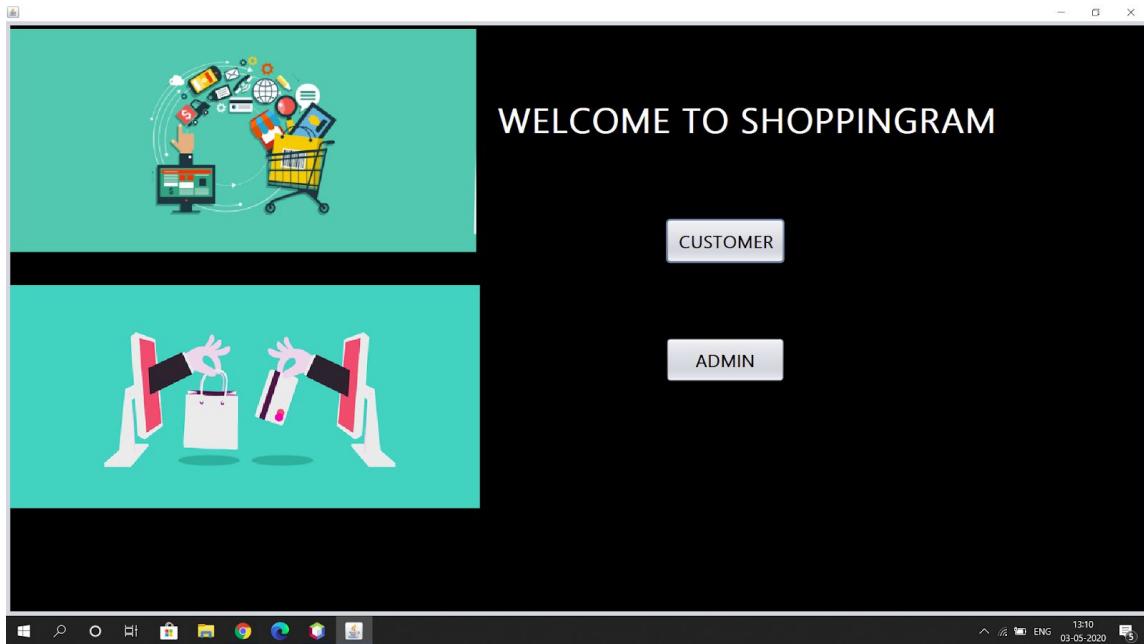
No	Username	NPD	TotP	D	Status
----	----------	-----	------	---	--------

### Admin:

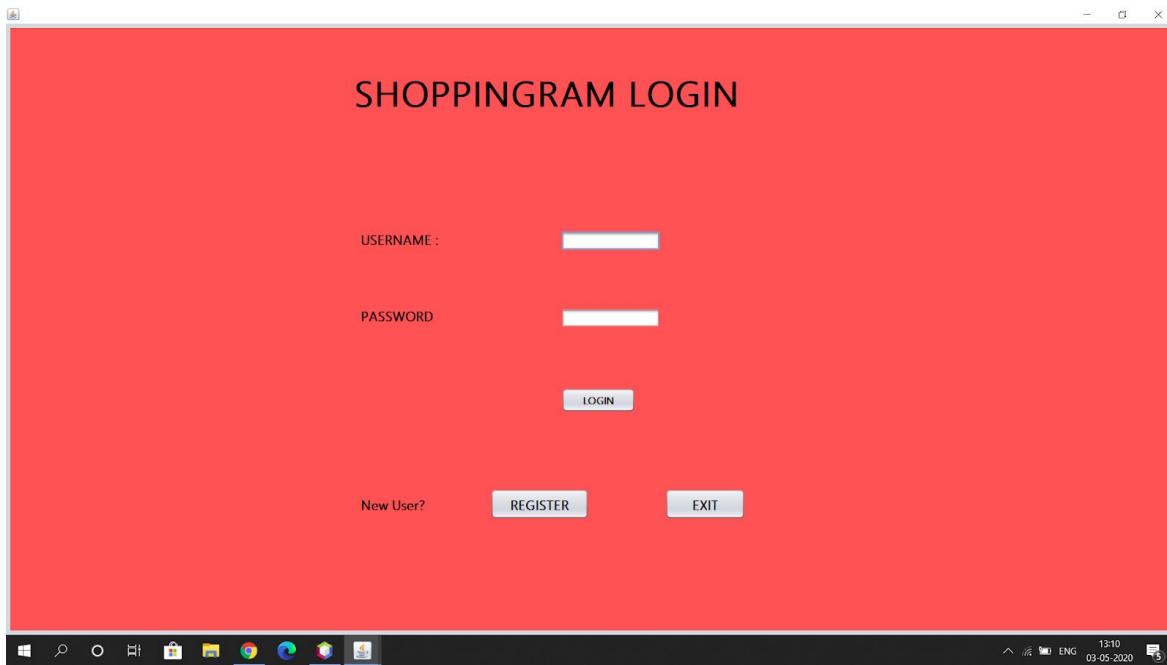
Alid	APW
------	-----

## **MODULE SCREENSHOTS(with explanation)->**

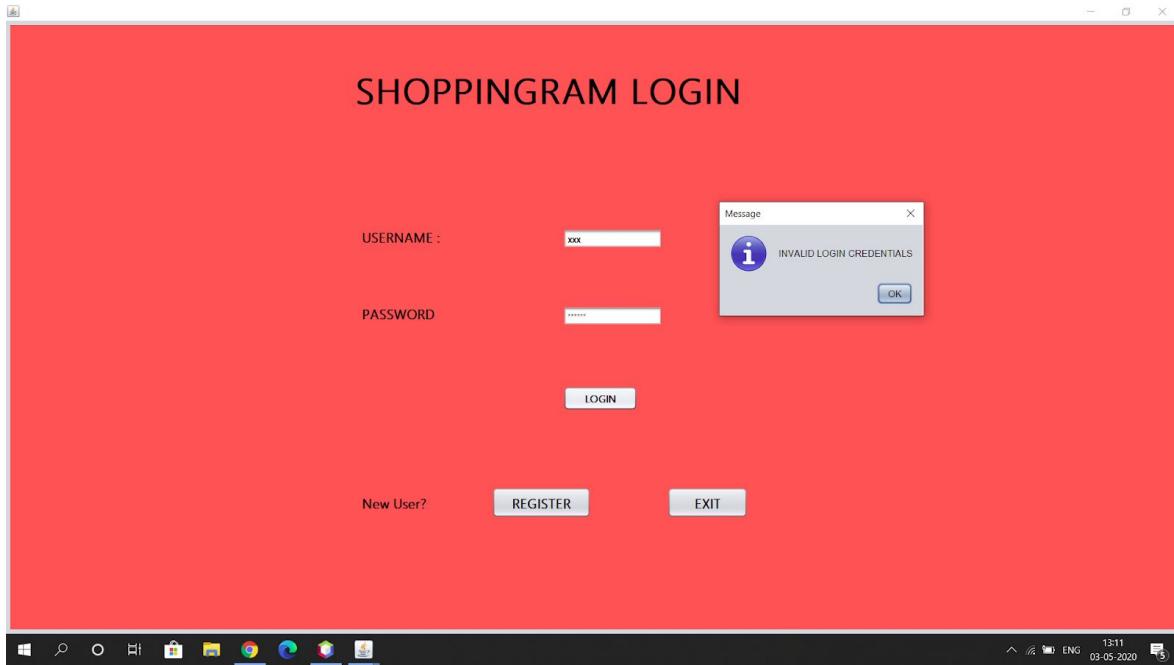
The first page of the front end application is shown below. This allows the user to choose whether he/she is the customer or admin.



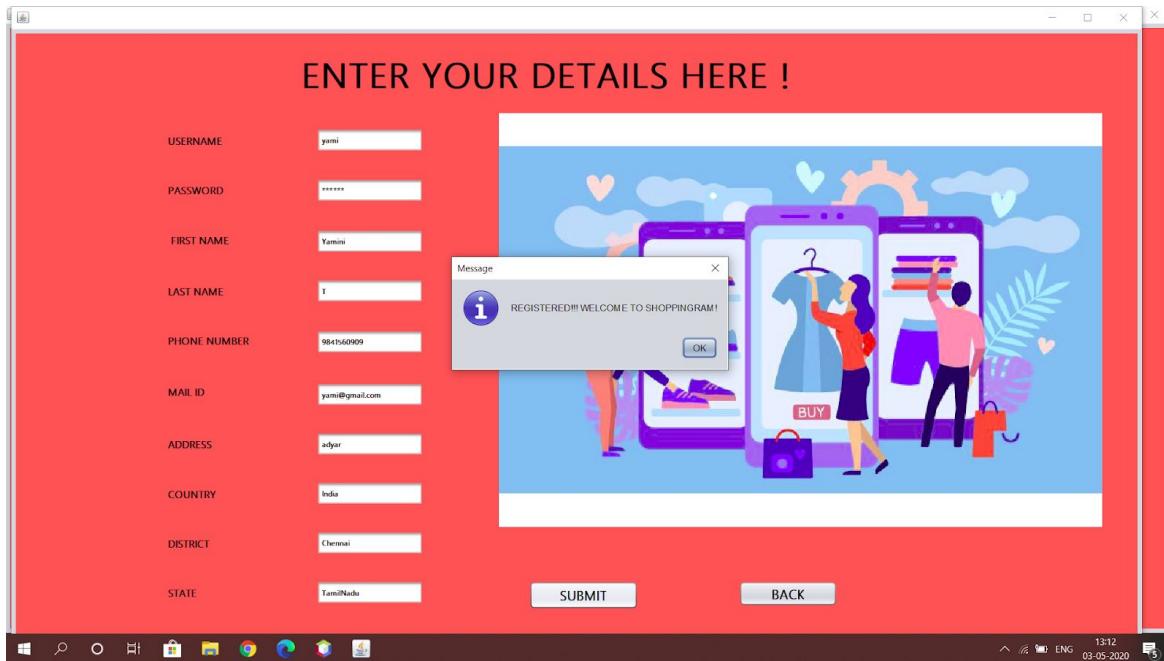
Clicking on customer in the above module transfers us to the customer login window where the customer can enter his/her details if he/she is already registered or register himself if he/she is a new user. Clicking on exit, exits the module.



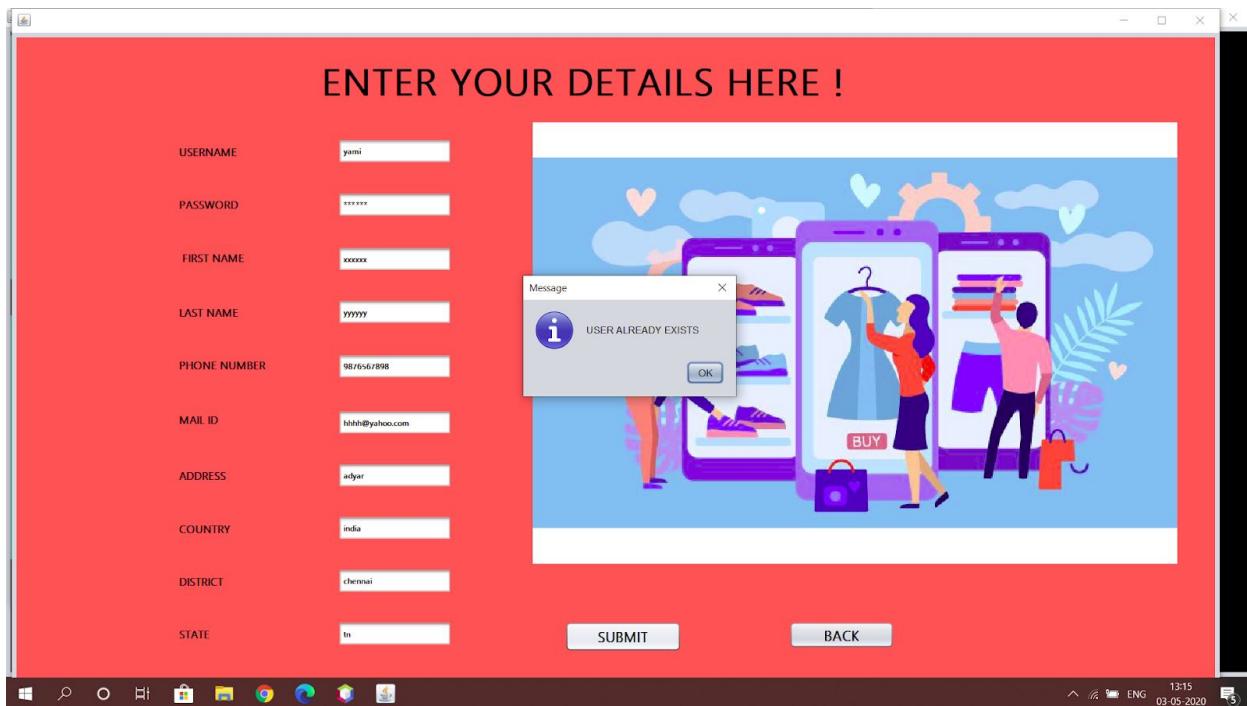
If invalid login details are entered by the customer and the LOGIN button is clicked,a dialog box stating "INVALID LOGIN CREDENTIALS' is displayed.



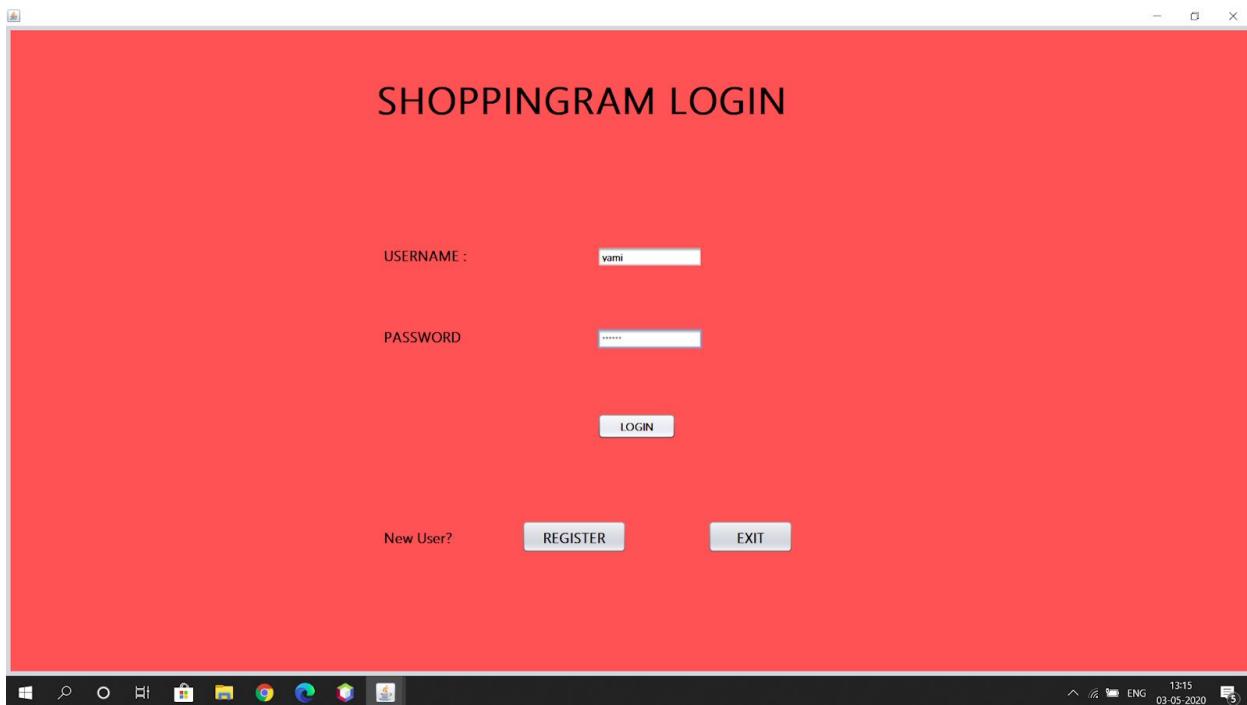
When the REGISTER button is clicked,the user is transferred to the customer register module where the user can enter his/her details and click on SUBMIT.Successful registration displays a dialog box stating “REGISTERED!!!WELCOME TO SHOPPINGRAM!”.



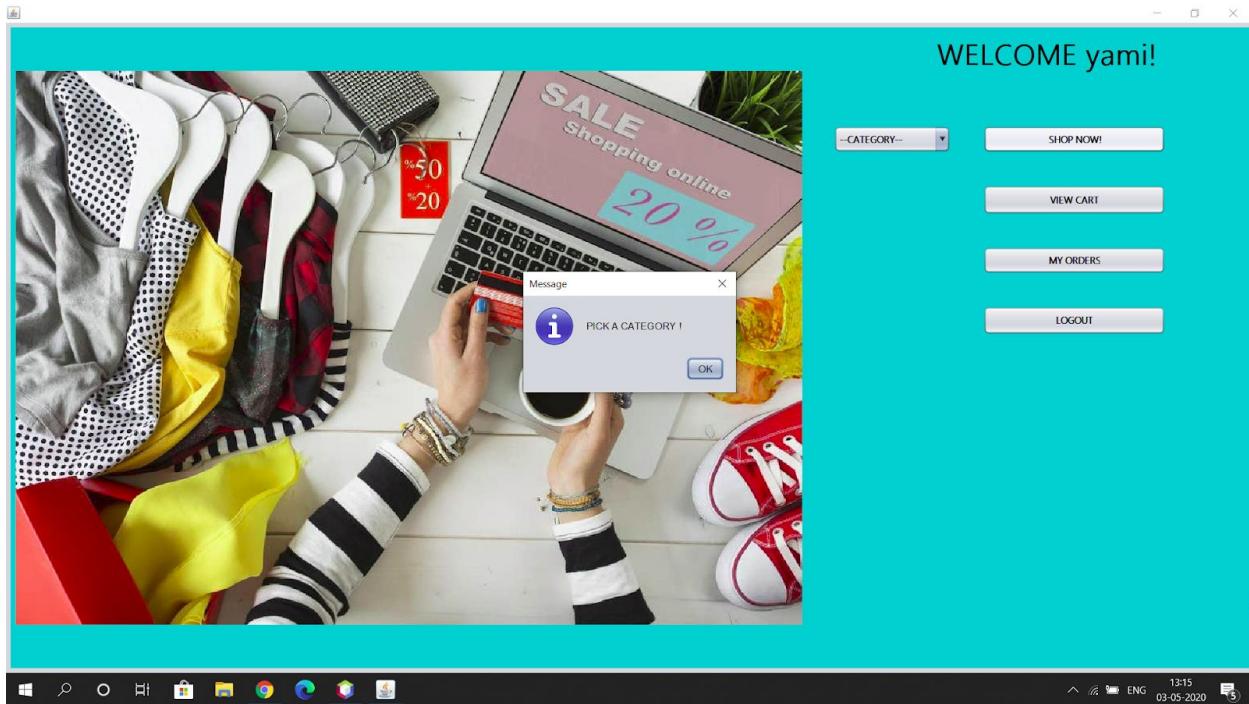
If a username that already exists is entered for registration a dialog box stating 'USER ALREADY EXISTS' is displayed.



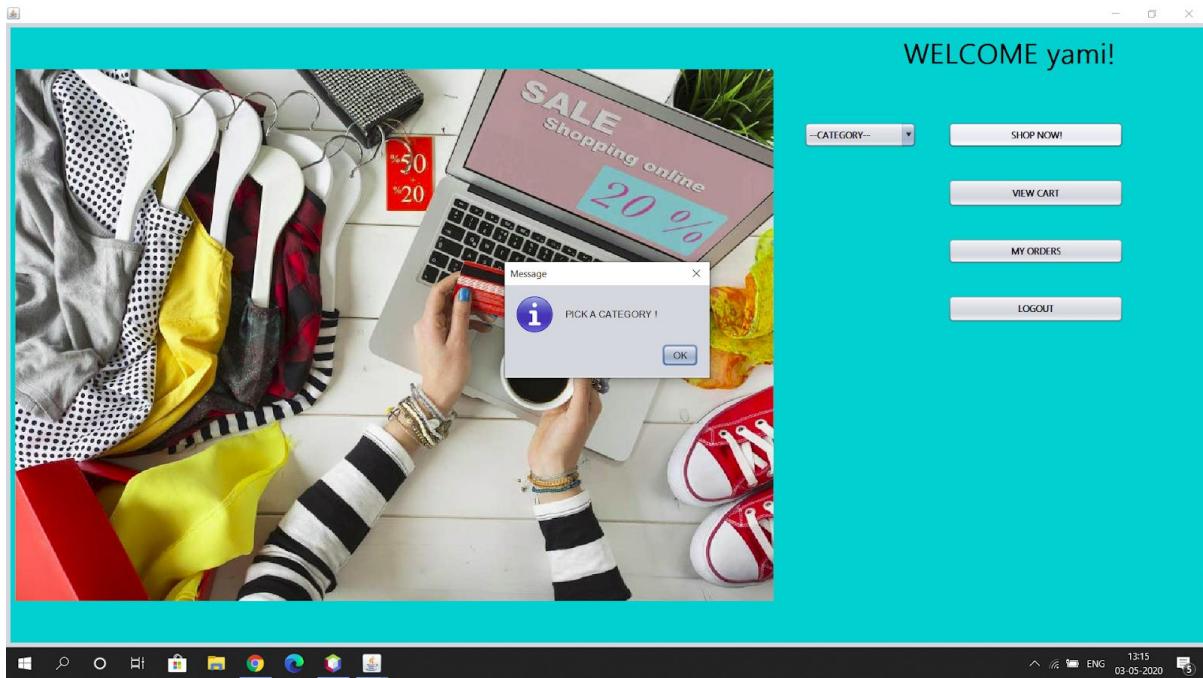
The below picture shows login information entered by the customer.



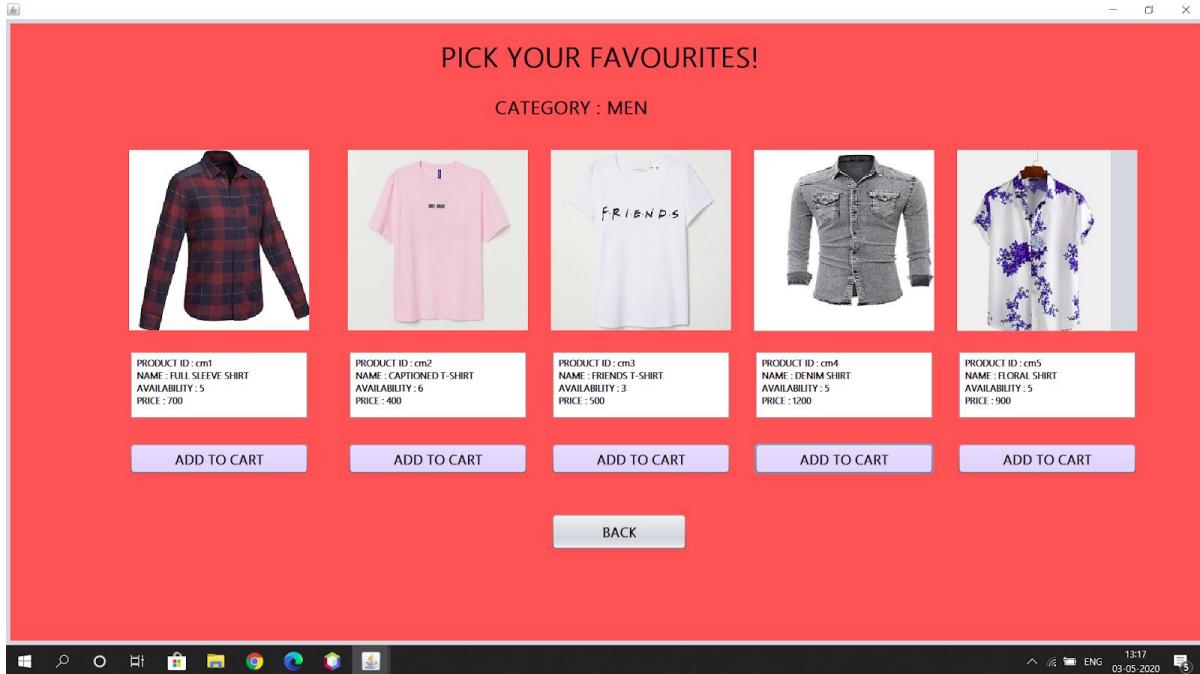
On clicking the SUBMIT button after entering correct login details, the customer is transferred to the customer options module.



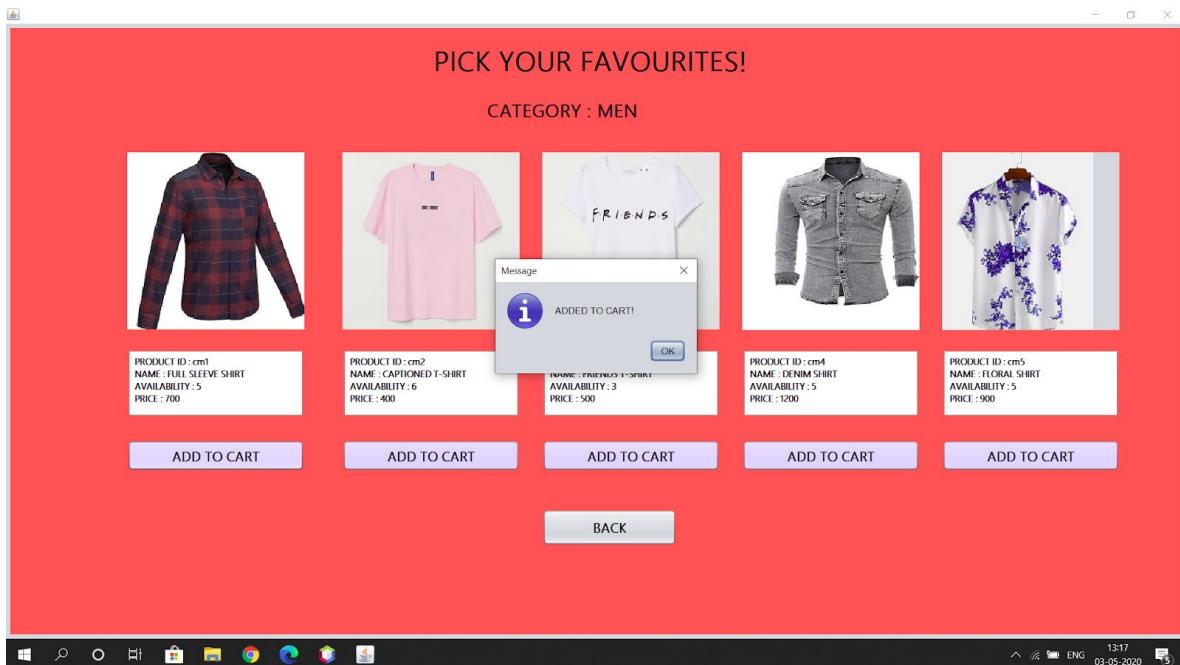
If SHOP NOW is clicked without selecting a category, it shows a dialog that asks you to pick a category.



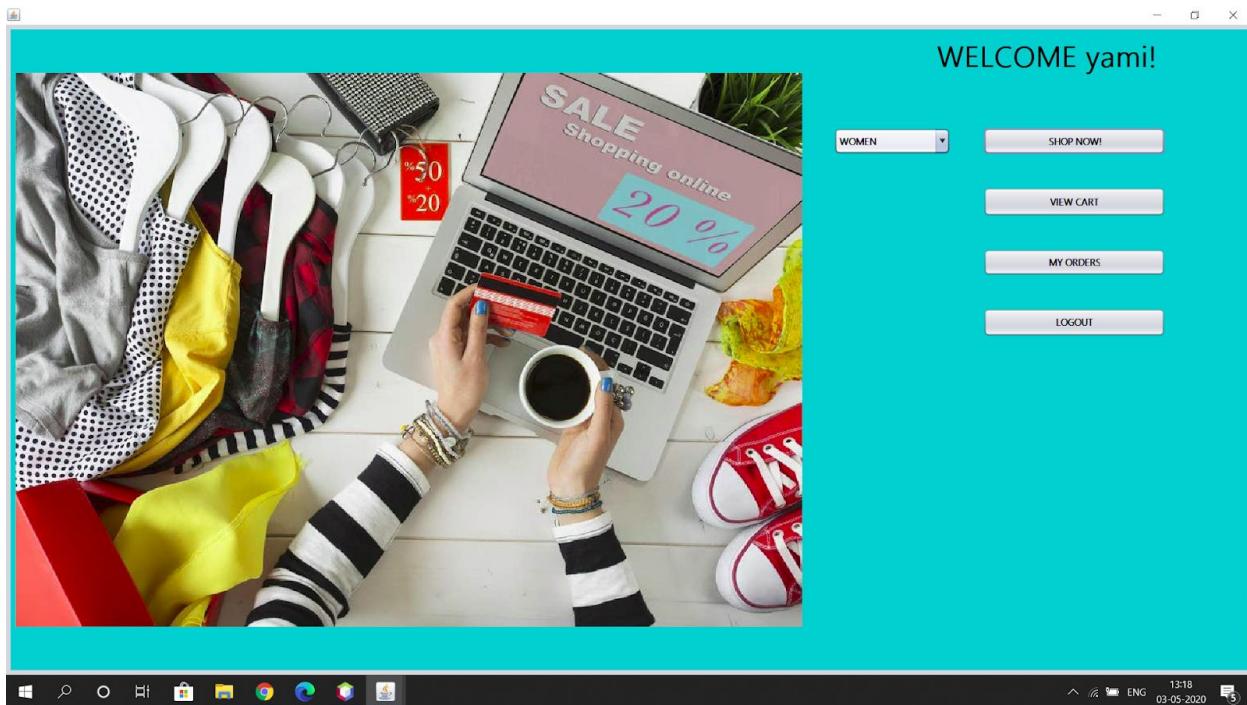
Clicking on 'SHOP NOW' in the customer options module after selecting the desired category transfers the customer to the shopping forum where the customer can select interested products by clicking on the 'ADD TO CART' button.



The 'ADD TO CART' button is clicked and the dialog box stating 'ADDED TO CART' is displayed.

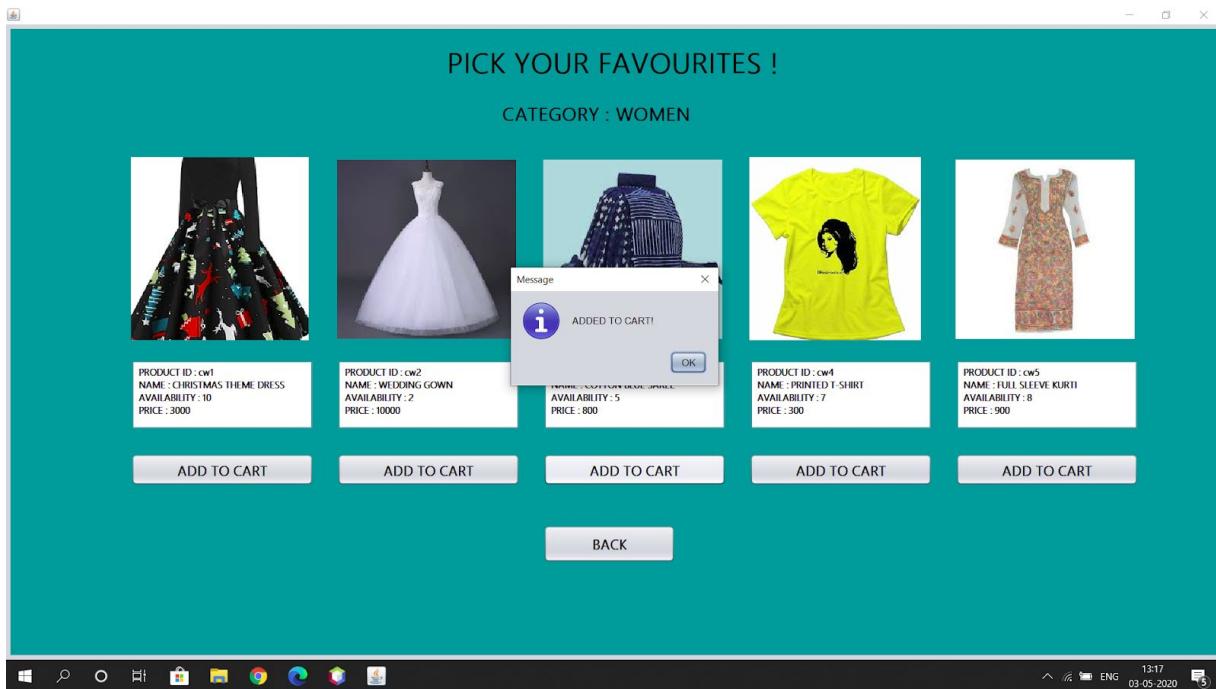


The WOMEN category is chosen in the customer login module.

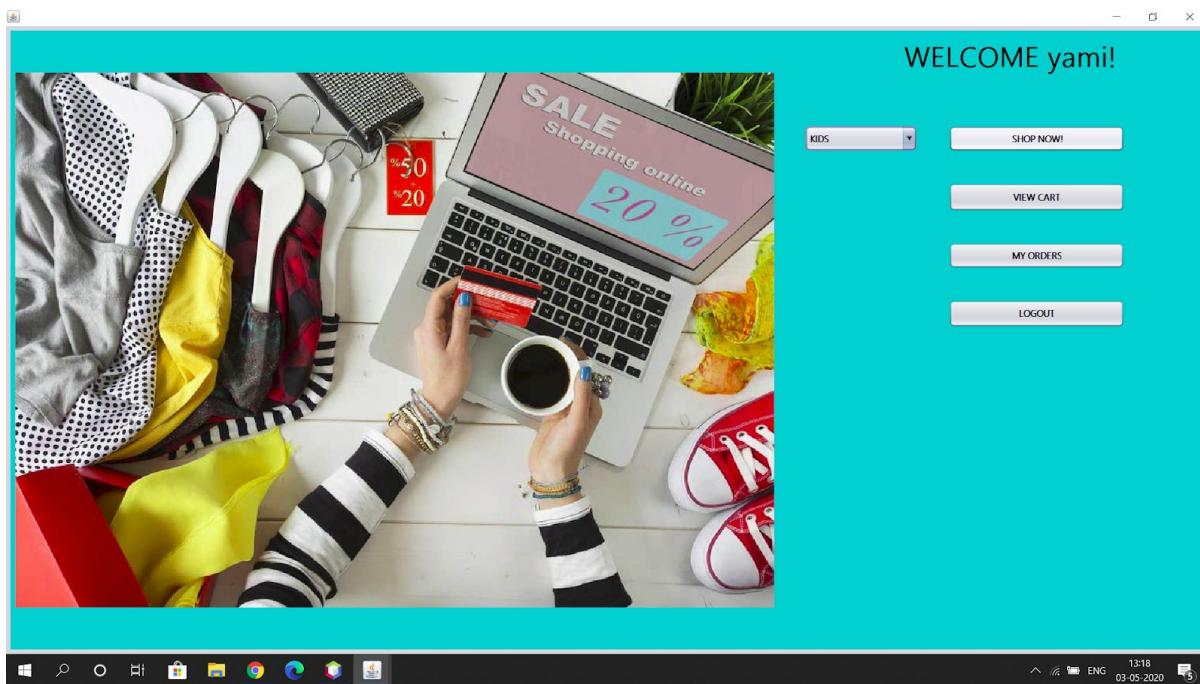


The womens shopping forum is displayed.

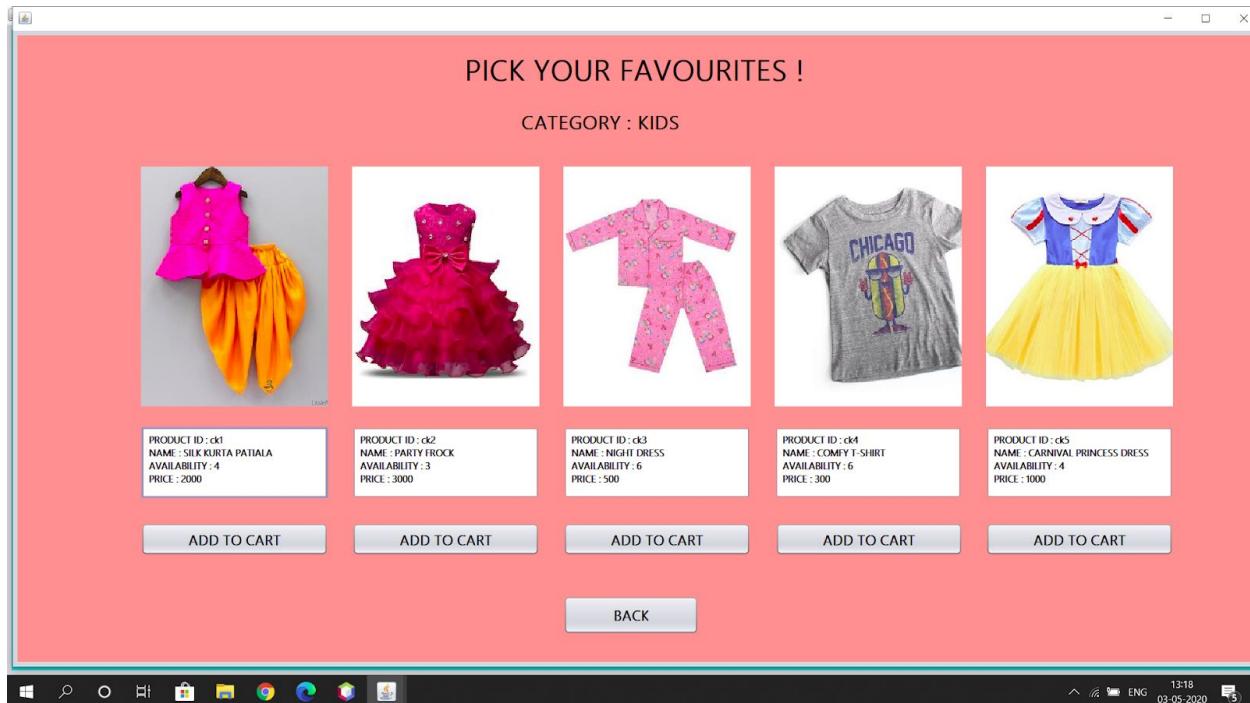
Clicking on the 'ADD TO CART' button is clicked and the dialog box stating 'ADDED TO CART' is displayed.



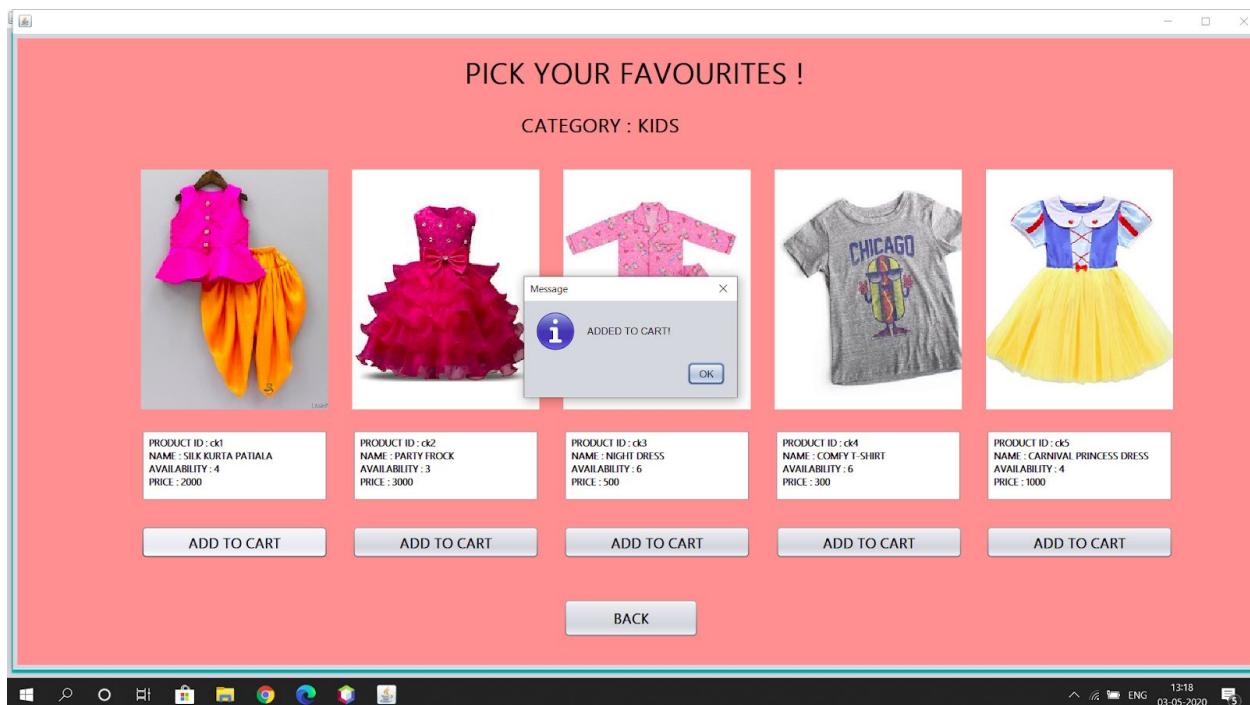
The KIDS category is chosen in the customer login module.



The kids shopping forum is displayed.

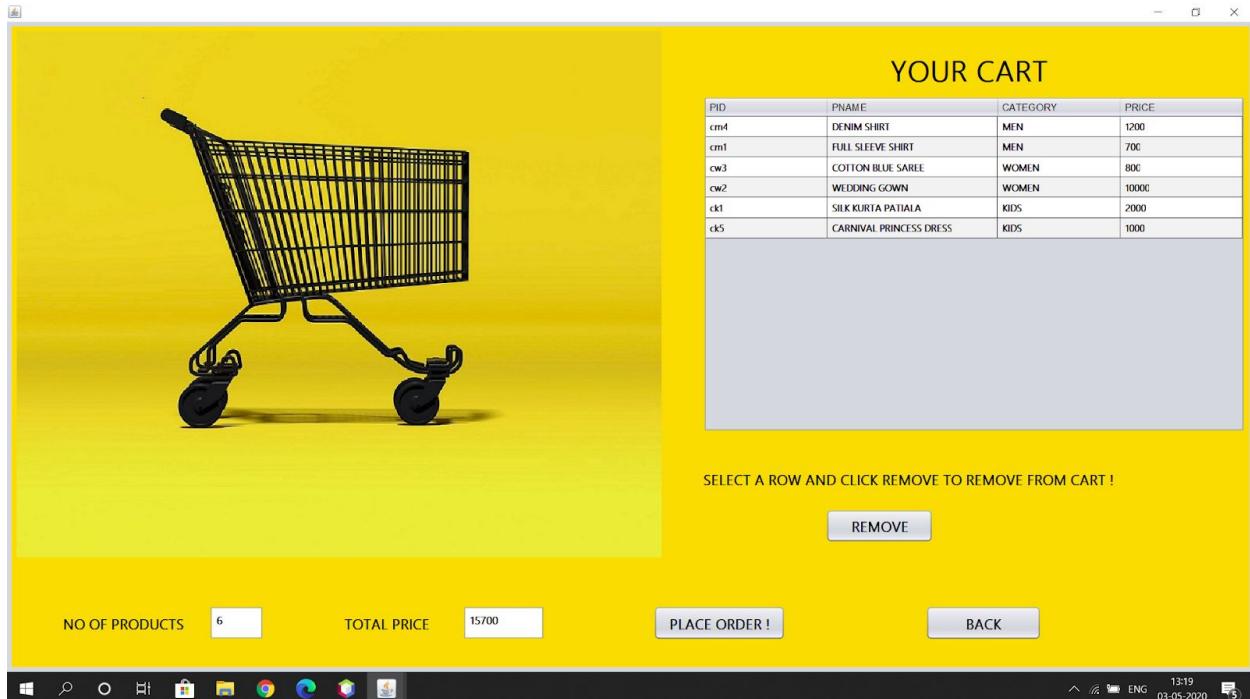


Clicking on the 'ADD TO CART' button is clicked and the dialog box stating 'ADDED TO CART' is displayed.

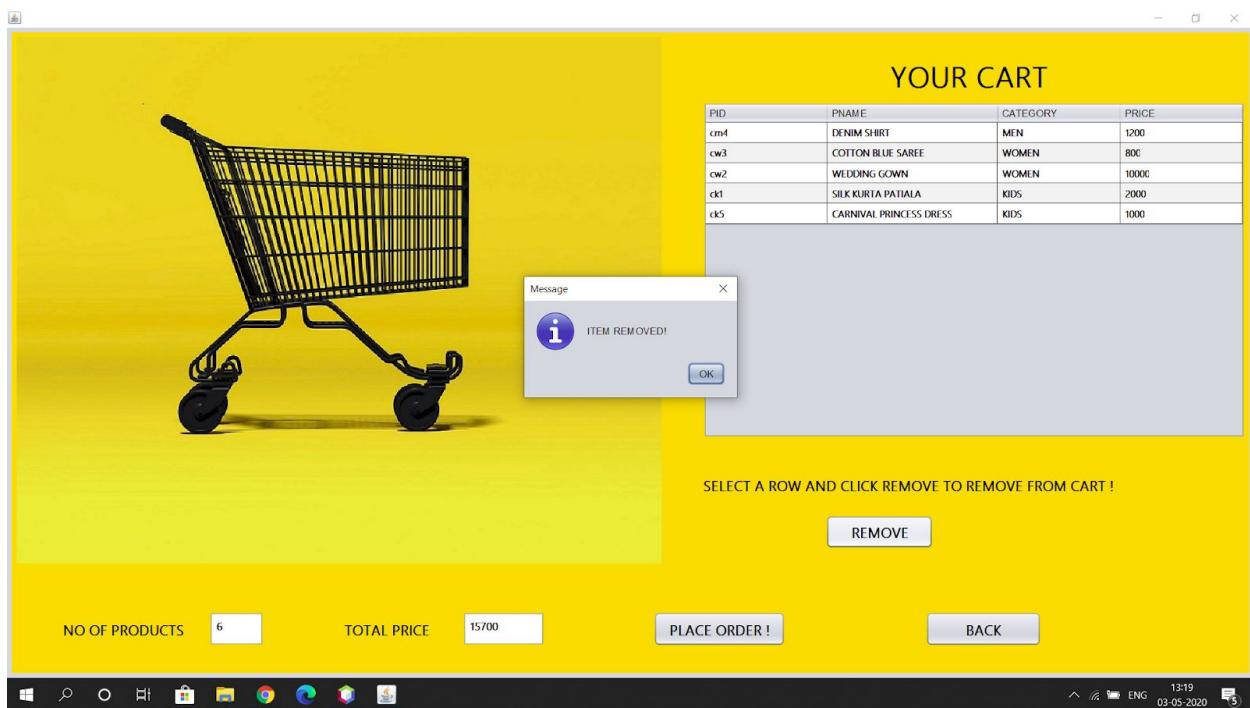


BACK button takes you back to the customer options module.

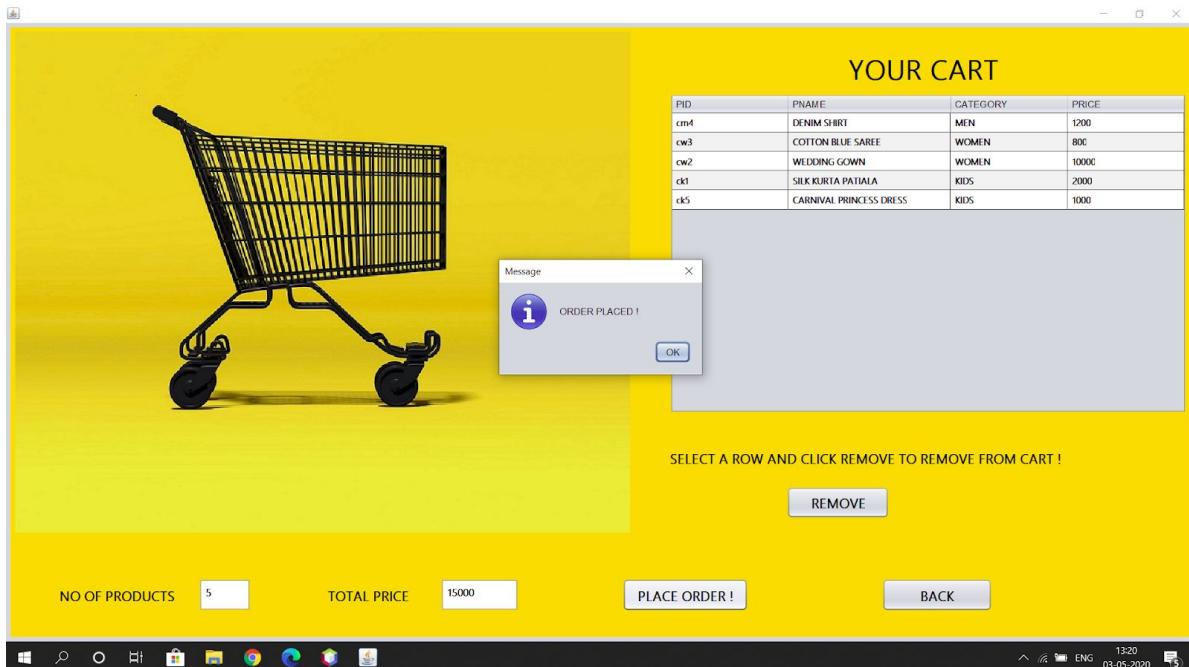
Clicking on 'VIEW CART' in the customer options module transferring user to cart module which displays the items selected for purchase by the customer and also no of products and total price is shown.



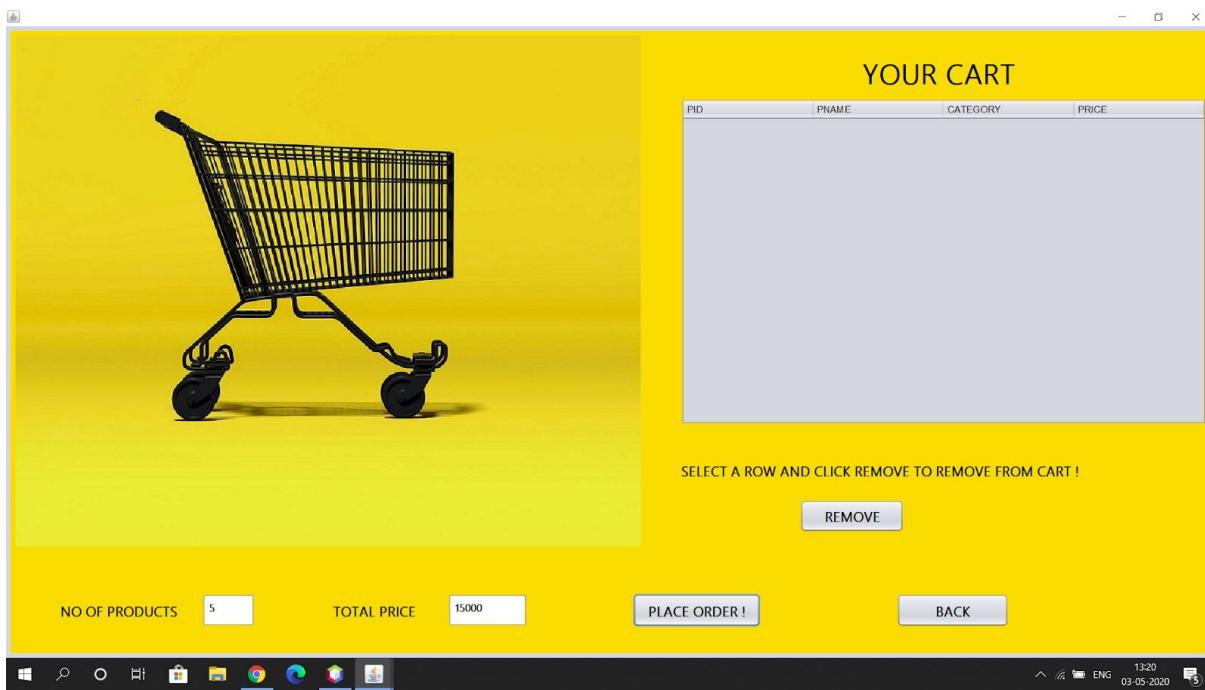
Clicking on a particular row and clicking on the REMOVE button removes the particular item from the cart and a dialog box showing 'ITEM REMOVED' is displayed.



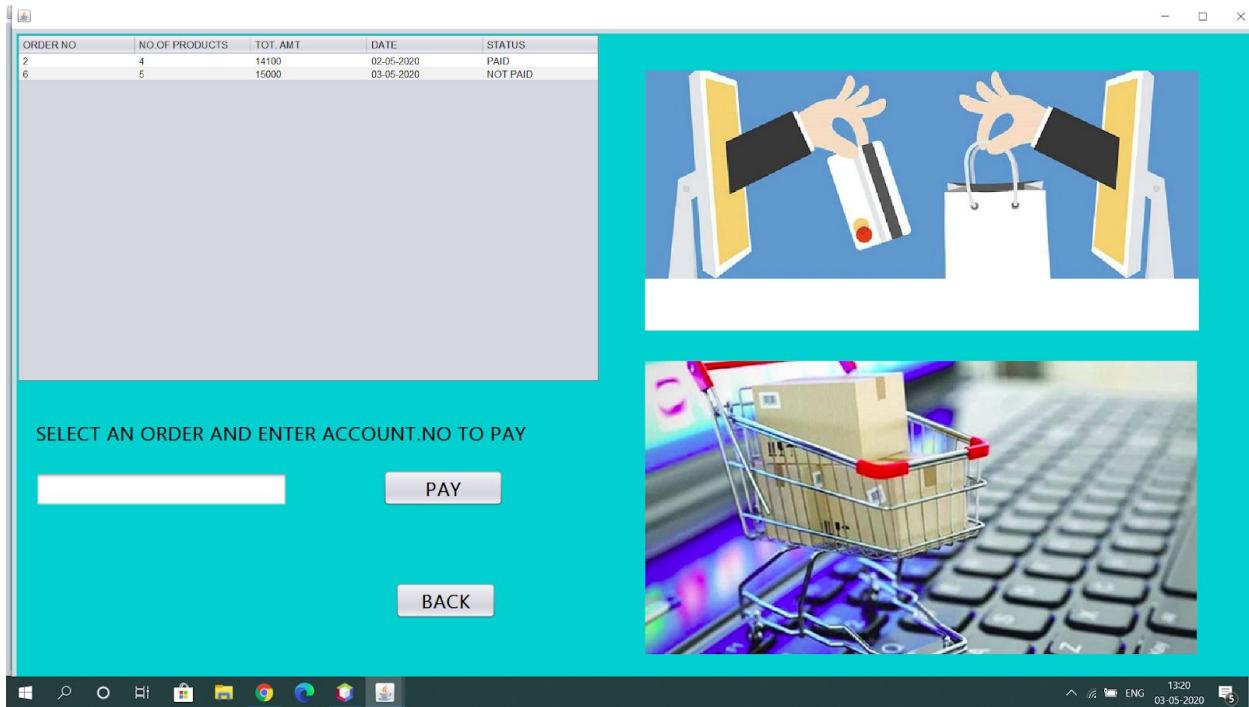
Clicking on 'PLACE ORDER!' gets the order placed and displays a dialog box stating 'ORDER PLACED'.



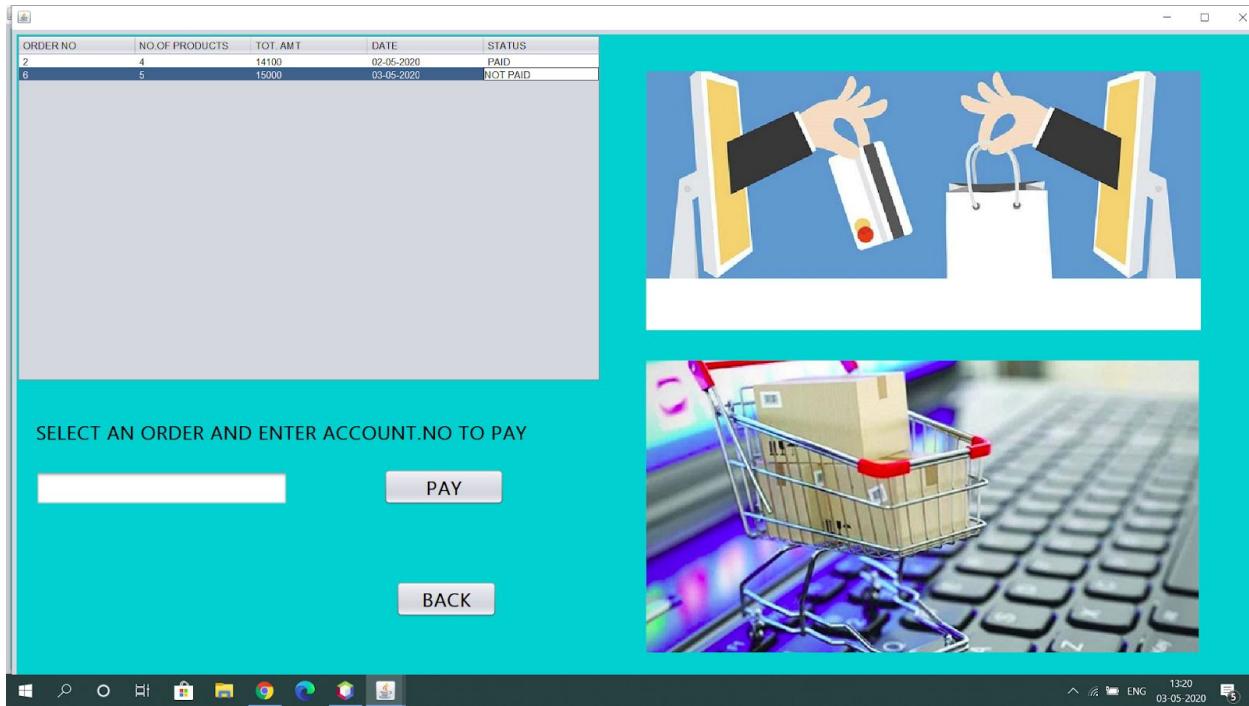
After the order is placed the cart becomes empty.



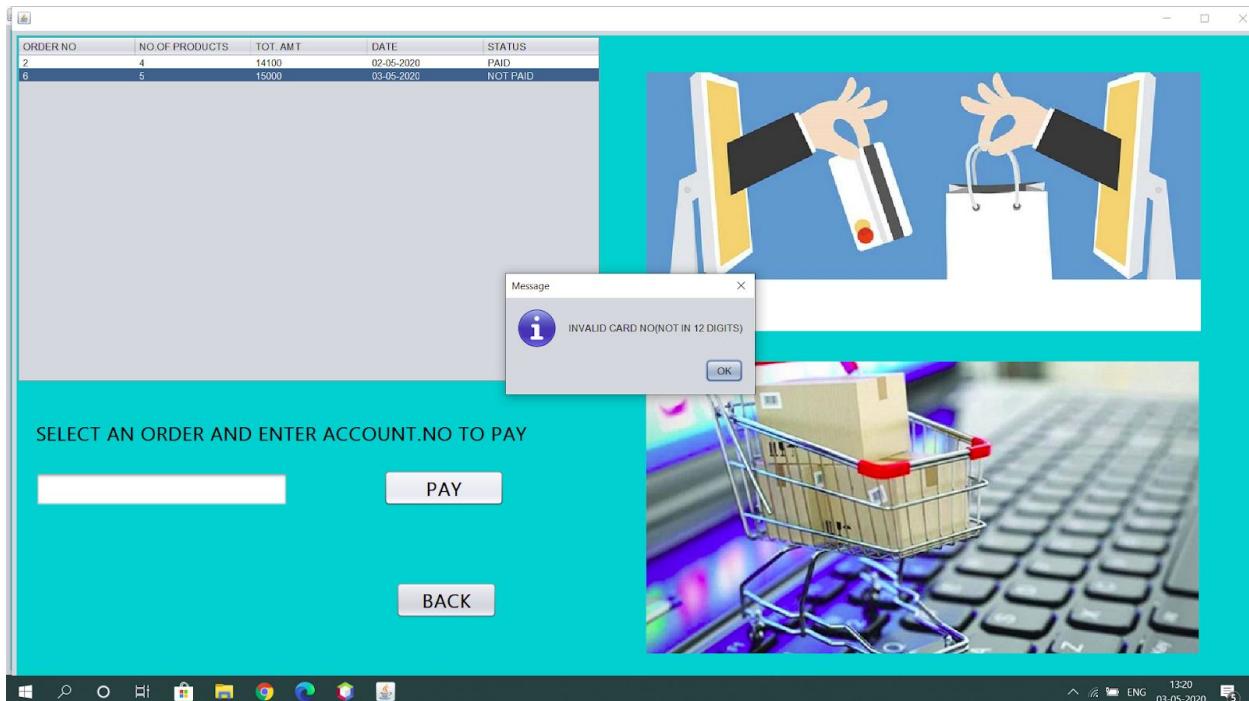
In the customer options module, when the 'MY ORDERS' button is clicked, the order module is displayed which contains all the orders that have been placed along with a text field to enter the card number to proceed for payment.



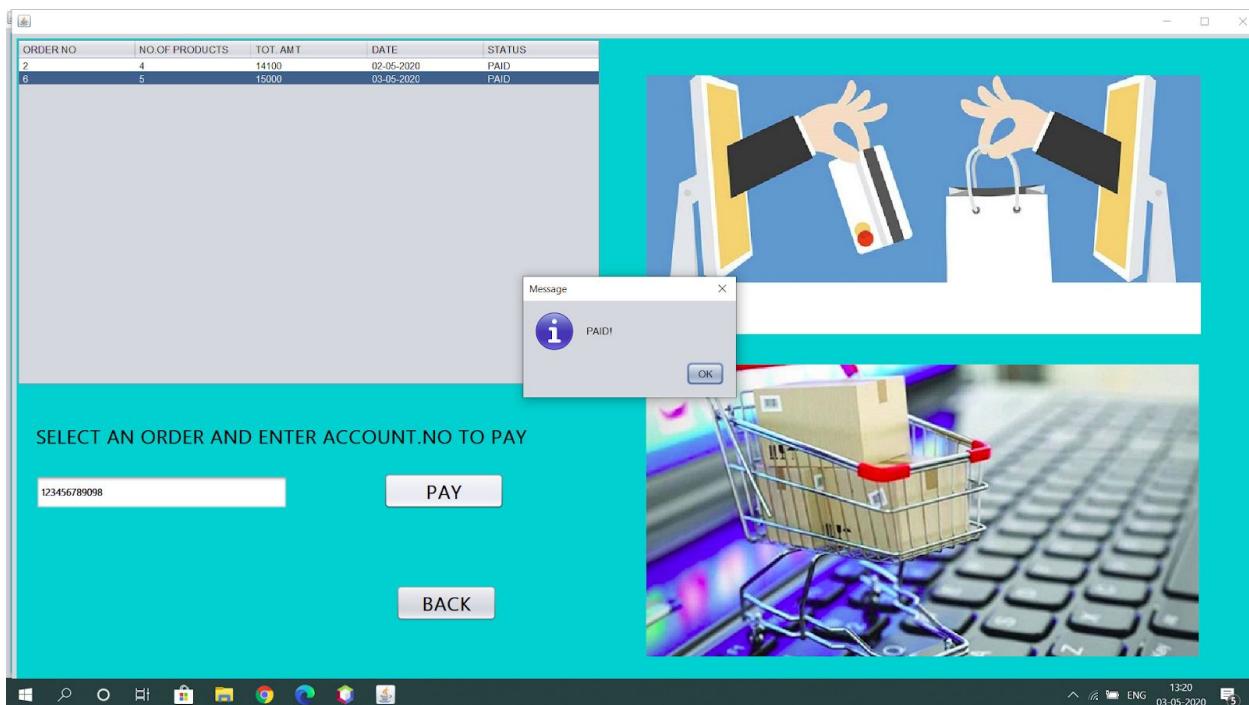
Selecting an order for which the customer hasn't paid yet from the table is done below.



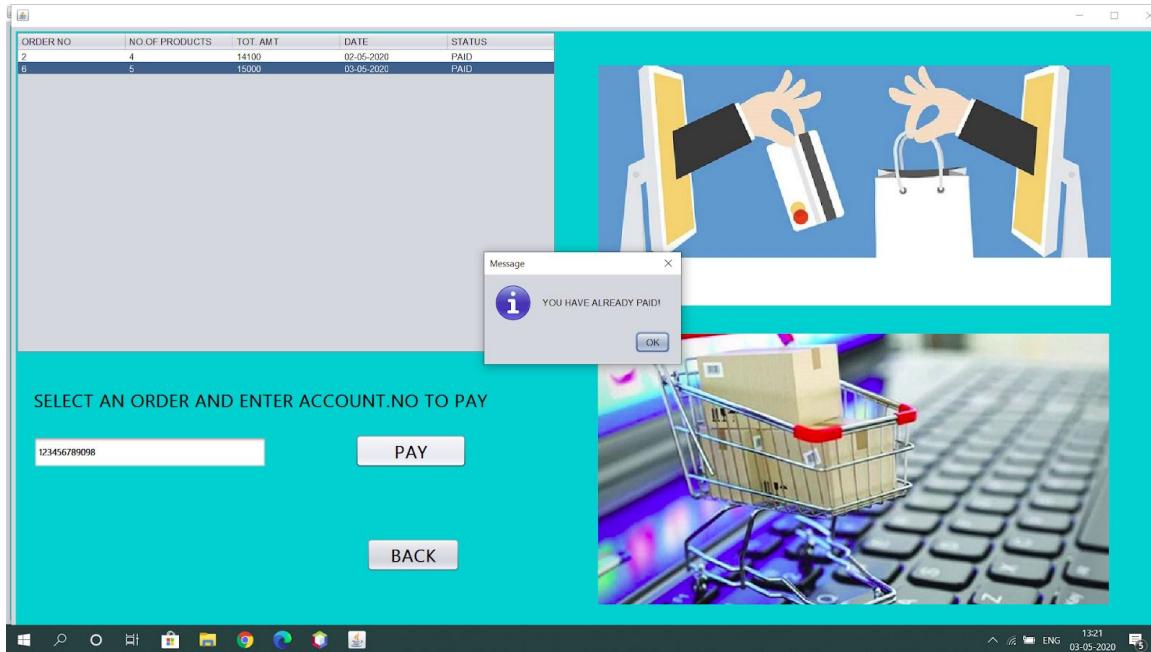
If a card number that does not consist of 12 digits is entered and 'PAY' button is clicked, the dialog box stating 'INVALID CARD NO(NOT IN 12 DIGITS)' is displayed.



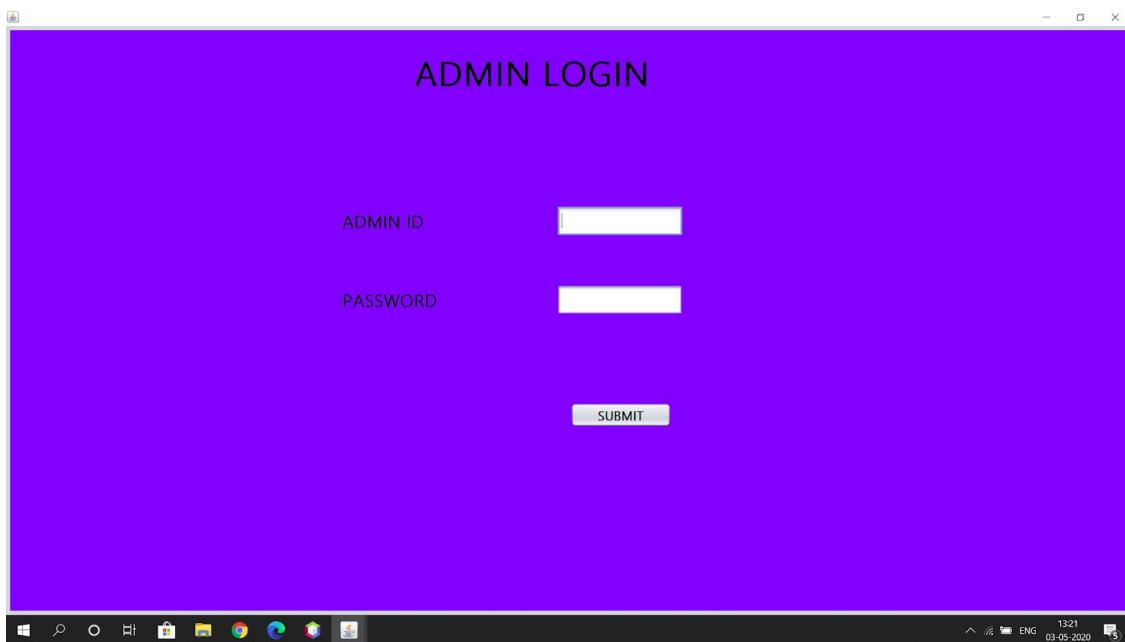
On successful payment, 'PAID' dialog box is displayed.



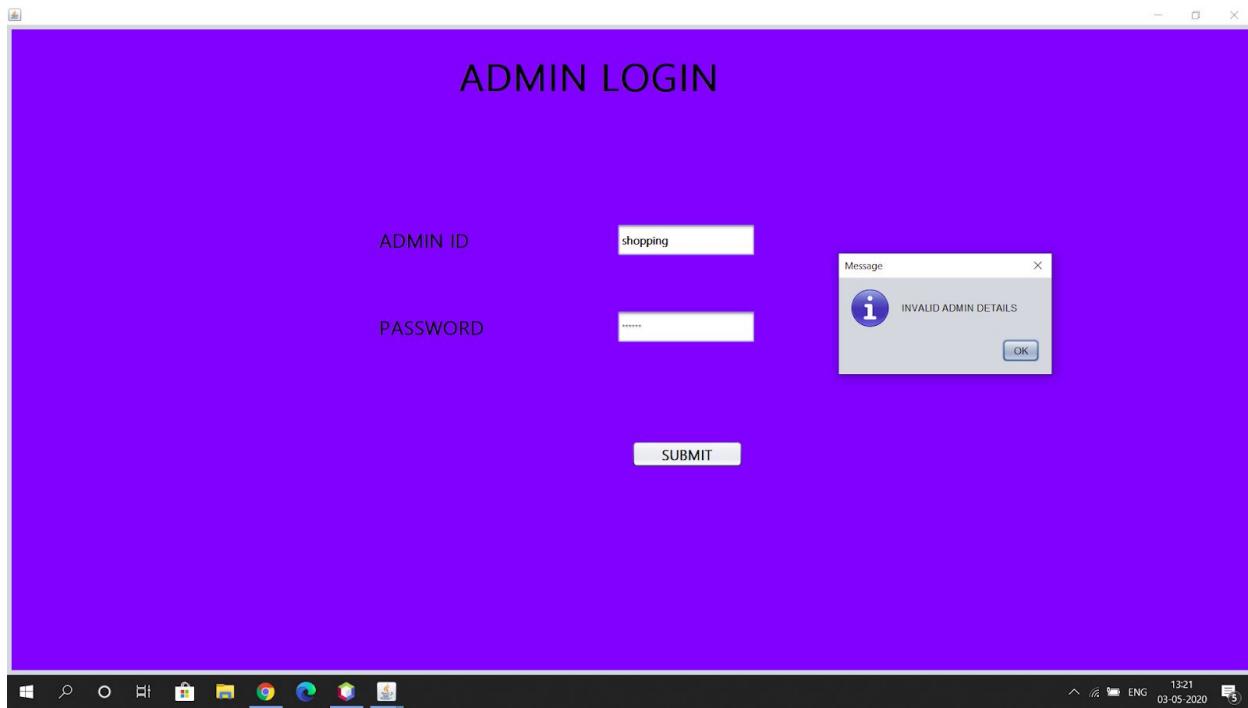
Selecting an order that is already paid and paying for such an order will result in a dialog box displaying 'YOU HAVE ALREADY PAID!'



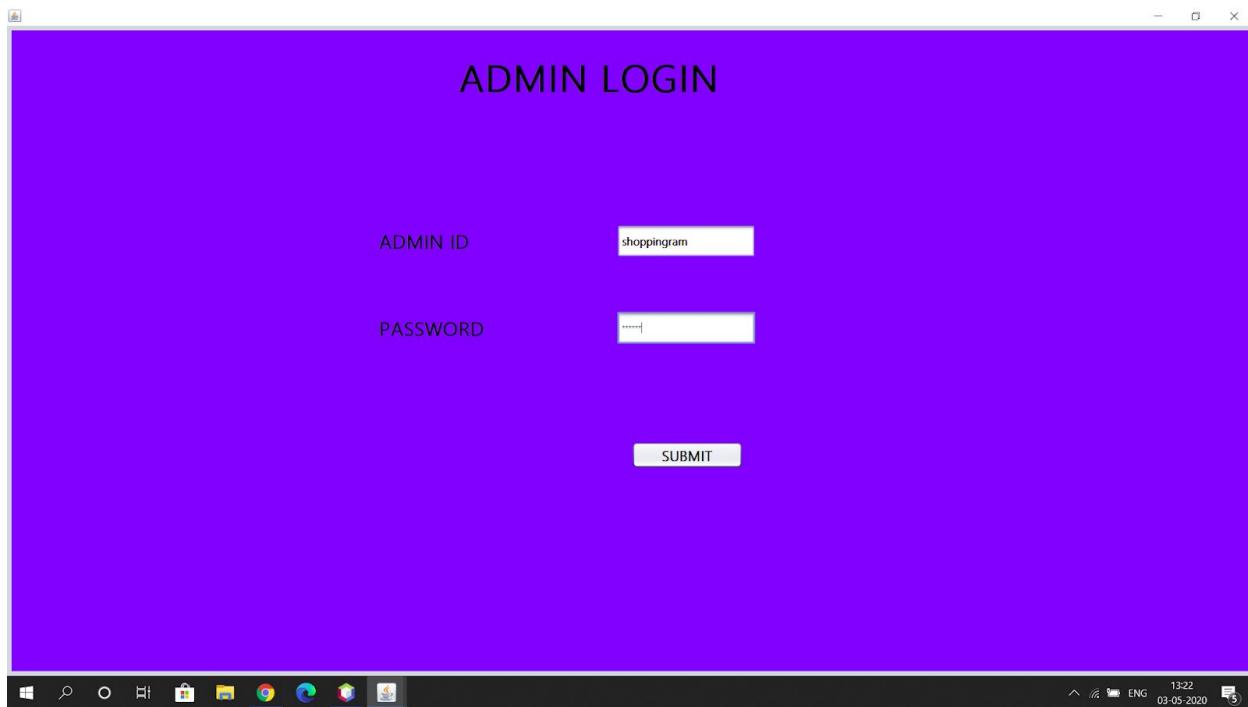
In the customer options module, clicking on the ADMIN button transfers to the admin login module.



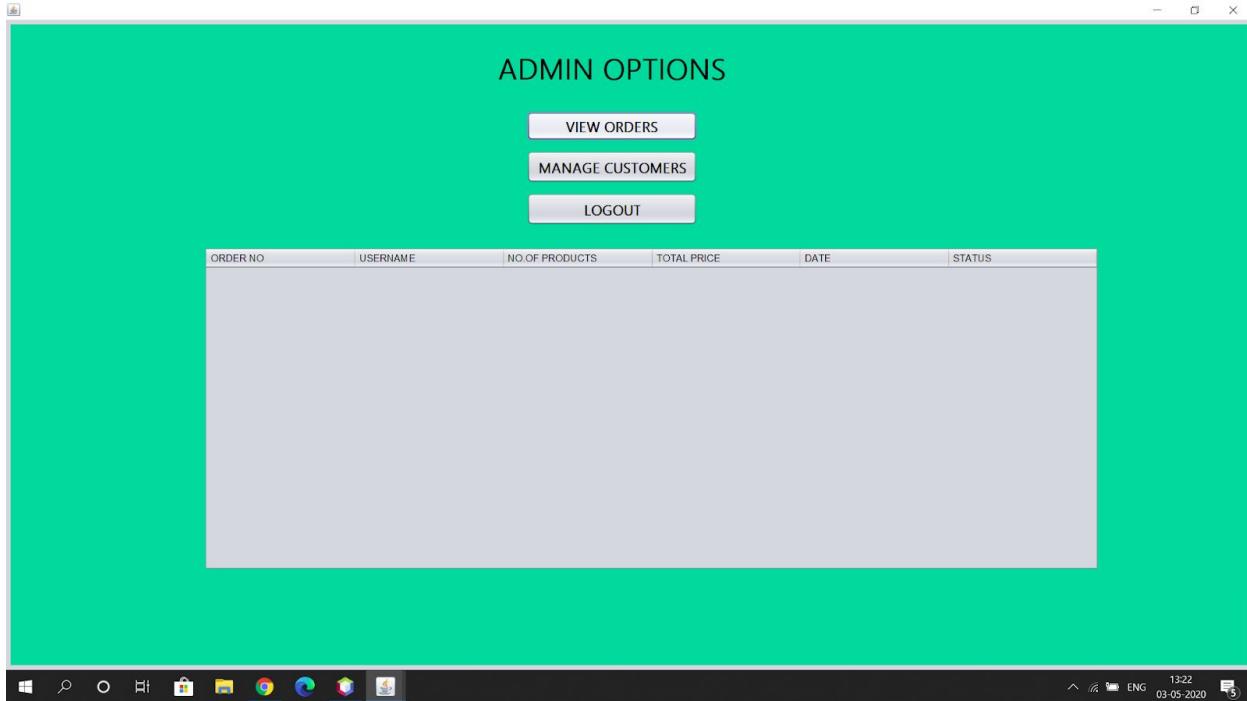
The admin login details are hardcoded initially. If wrong login details are entered dialog box stating 'INVALID ADMIN DETAILS' is displayed.



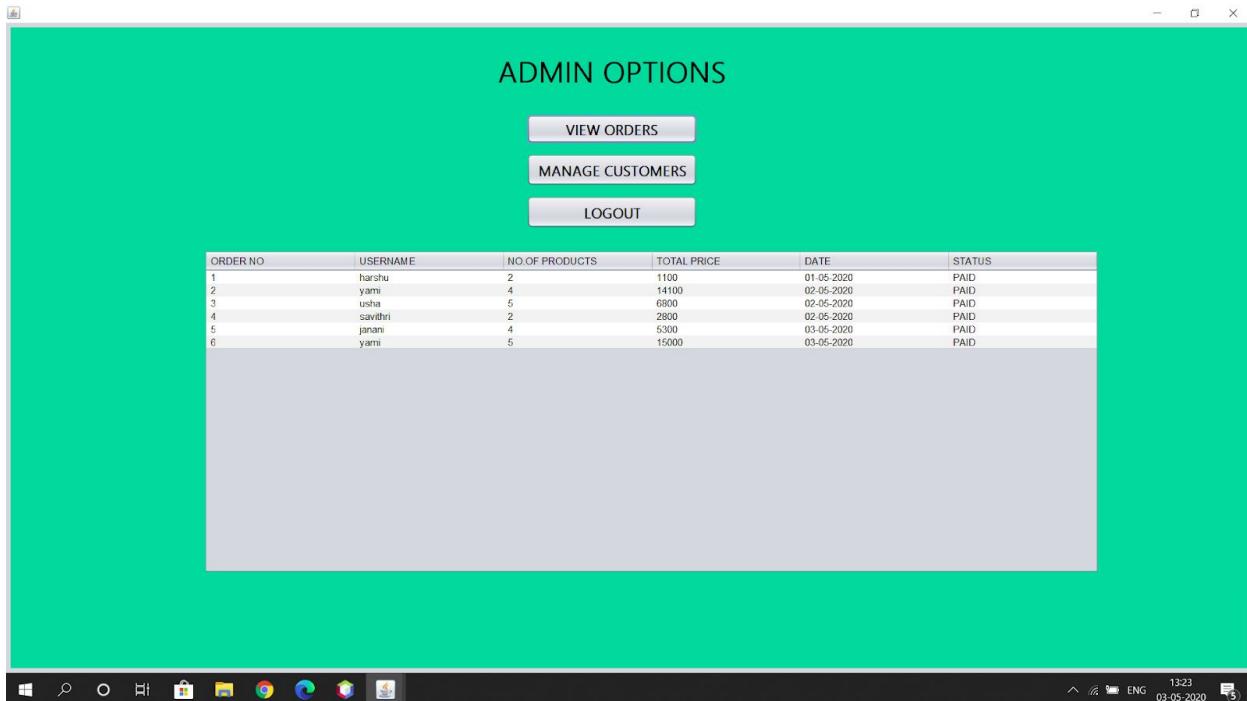
The correct login details are entered and the SUBMIT button is clicked.



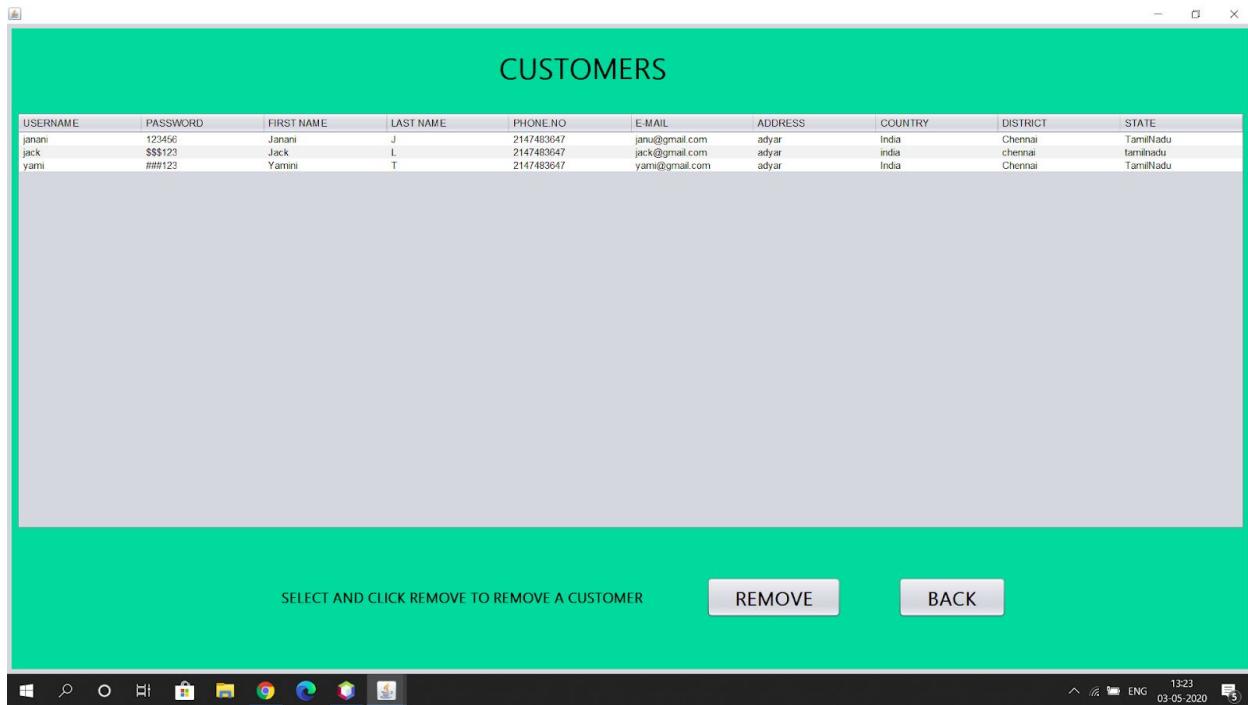
On successful login, the admin options module is displayed.



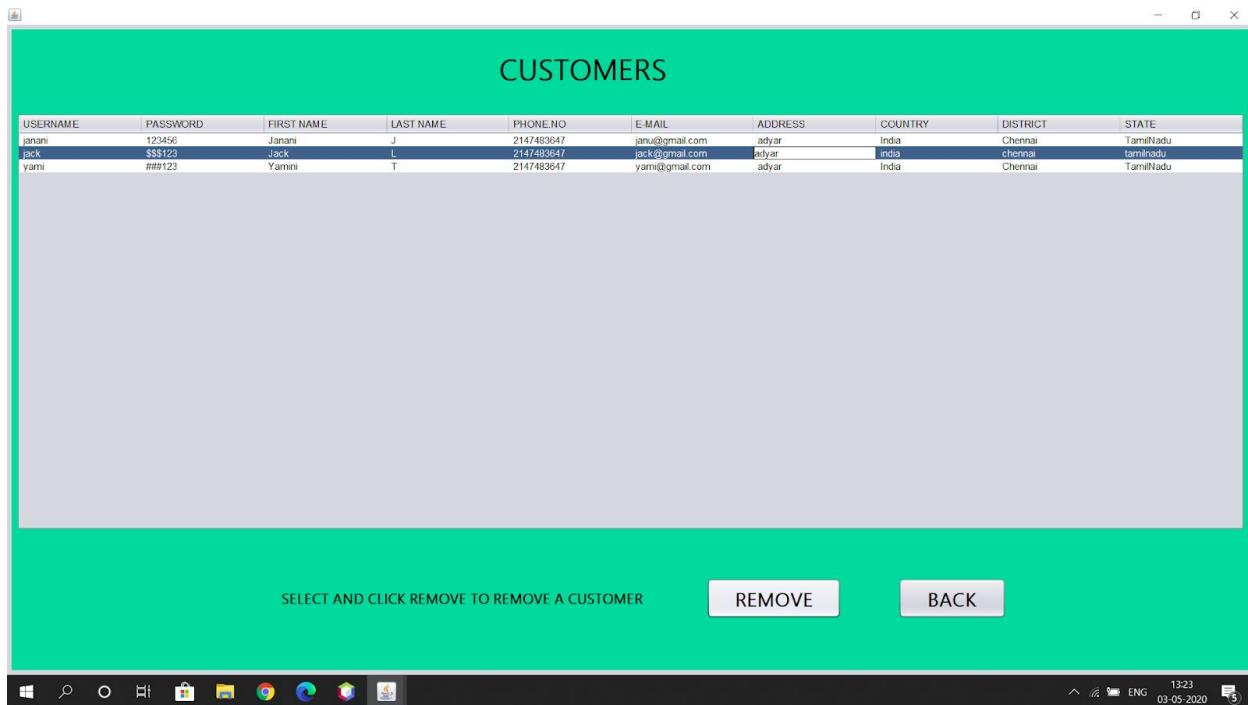
Clicking on 'VIEW ORDERS' button displays the orders that have been placed by the customers.



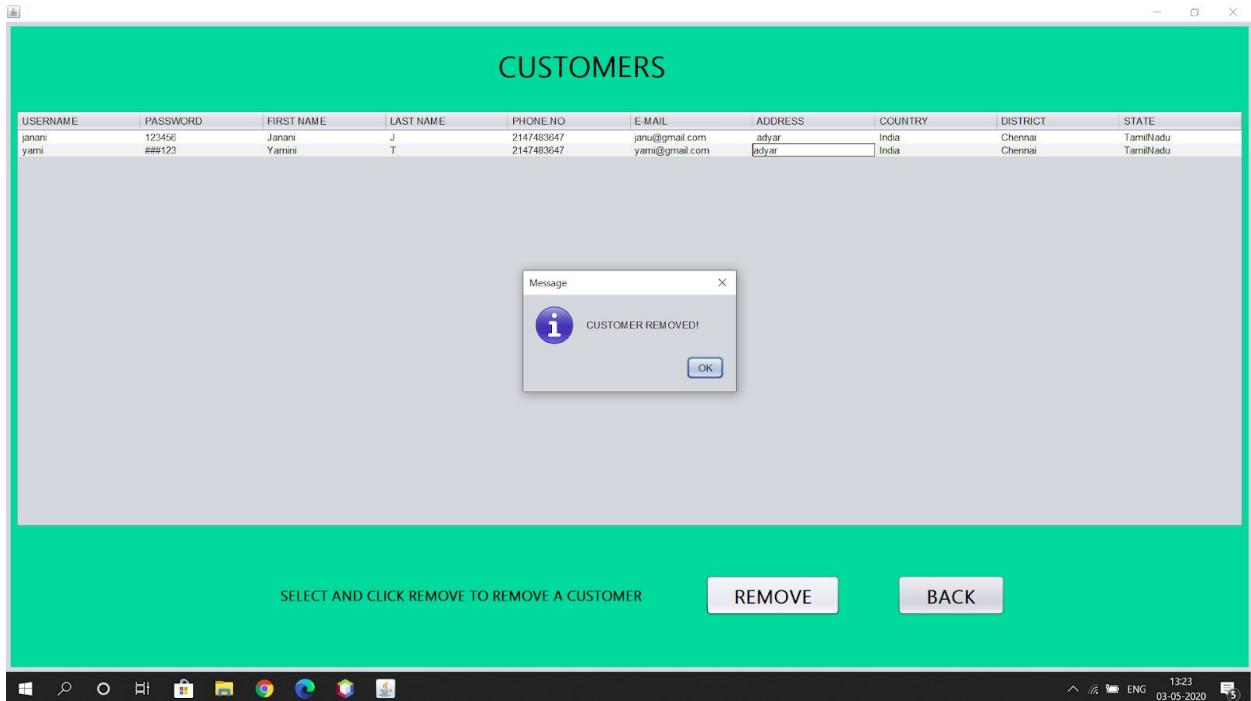
Clicking the ‘MANAGE CUSTOMERS’ button displays the customers who have registered themselves



A particular customer is selected and removed by clicking on the REMOVE button.



On successful removal of customers,a dialog box is displayed stating the same.



## **SQL(CREATION AND INSERTION) CODE->**

```
create table customer(username varchar2(15) constraint pkc primary key,
                      pw varchar2(15),
                      fname char(10),
                      lname char(10),
                      ph_no varchar2(10),
                      email varchar2(20),
                      addr varchar2(20),
                      country char(15),
                      district char(15),
                      state char(15));
create table admin(aid varchar2(15) constraint pkadmin primary key,
                   apw varchar2(15));
insert into admin values('shoppingram','###123');
create table product(pid varchar2(5) constraint prpd primary key,
                     pname varchar2(30),
                     avail number(2),
                     category char(10),
                     price number(10));
insert into product values('cm1','FULL SLEEVE SHIRT',5,'MEN',700);
insert into product values('cm2','CAPTIONED T-SHIRT',6,'MEN',400);
insert into product values('cm3','FRIENDS T-SHIRT',3,'MEN',500);
insert into product values('cm4','DENIM SHIRT',5,'MEN',1200);
insert into product values('cm5','FLORAL SHIRT',5,'MEN',900);
insert into product values('cw1','CHRISTMAS THEME DRESS',10,'WOMEN',3000);
insert into product values('cw2','WEDDING GOWN',2,'WOMEN',10000);
insert into product values('cw3','COTTON BLUE SAREE',5,'WOMEN',800);
insert into product values('cw4','PRINTED T-SHIRT',7,'WOMEN',300);
insert into product values('cw5','FULL SLEEVE KURTI',8,'WOMEN',900);
insert into product values('ck1','SILK KURTA PATIALA',4,'KIDS',2000);
insert into product values('ck2','PARTY FROCK',3,'KIDS',3000);
insert into product values('ck3','NIGHT DRESS',6,'KIDS',500);
insert into product values('ck4','COMFY T-SHIRT',6,'KIDS',300);
insert into product values('ck5','CARNIVAL PRINCESS DRESS',4,'KIDS',1000);
create table cart(username varchar2(15),
                  pid varchar2(5) constraint refp references product(pid));
create table order_c(no number(2) constraint orde primary key,
                    username varchar2(15),
                    npd number(2),
                    totp number(10),
                    d varchar2(15),
                    status char(10));
```

## Coding->

### Connectivity code:

```
package miniproject;
```

```
import java.sql.*;
```

```
public class Miniproject
```

```
{
```

```
    public static void main(String[] args)
```

```
{
```

```
        Connection conn=null;
```

```
        try { String driverName="oracle.jdbc.driver.OracleDriver";
```

```
            Class.forName(driverName);
```

```
            conn=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-N6B  
                                         HVVF:1521:XE", "mini", "mini123");
```

```
            System.out.println("Connection Successful");
```

```
}
```

```
        catch (ClassNotFoundException e)
```

```
{
```

```
            System.out.println("could not find conn db driver"+e.getMessage());
```

```
}
```

```
        catch(SQLException e)
```

```
{
```

```
            System.out.println("could not connect to db"+e.getMessage());
```

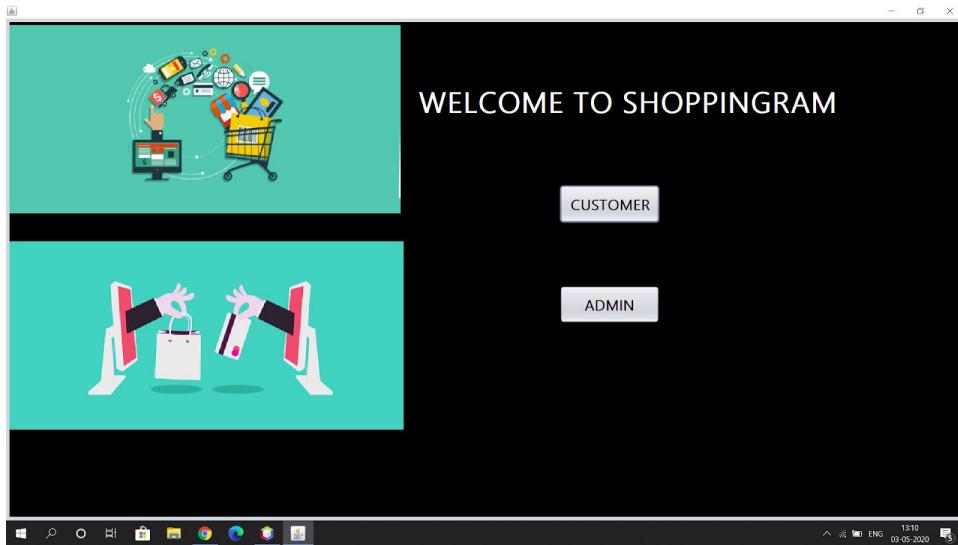
```
}
```

```
}
```

```
}
```

## Implementation code:

Front module(front.java) ->



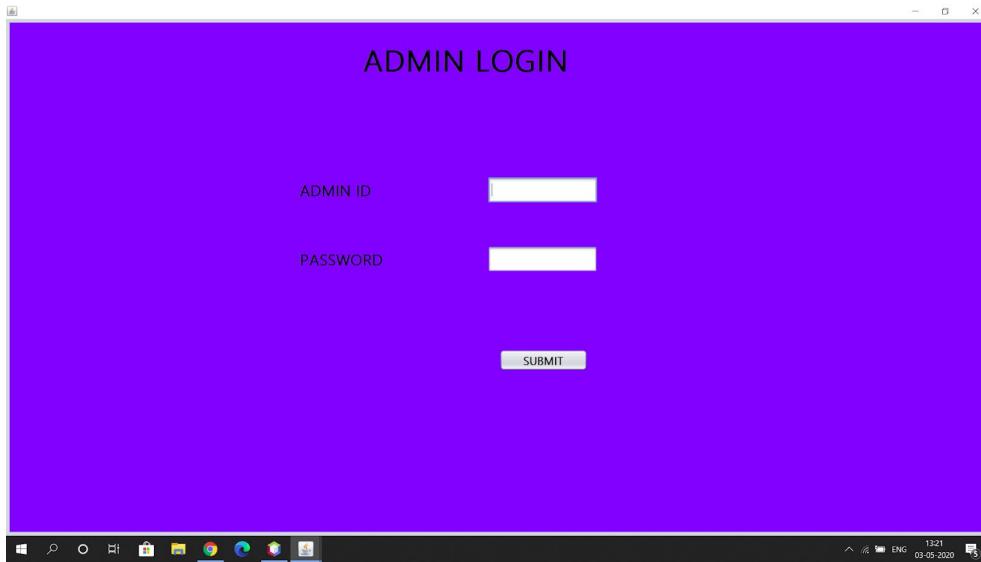
If CUSTOMER button is clicked->

```
cust_login c=new cust_login();
c.setVisible(true);
```

If ADMIN button is clicked->

```
admin_login ad=new admin_login();
ad.setVisible(true);
```

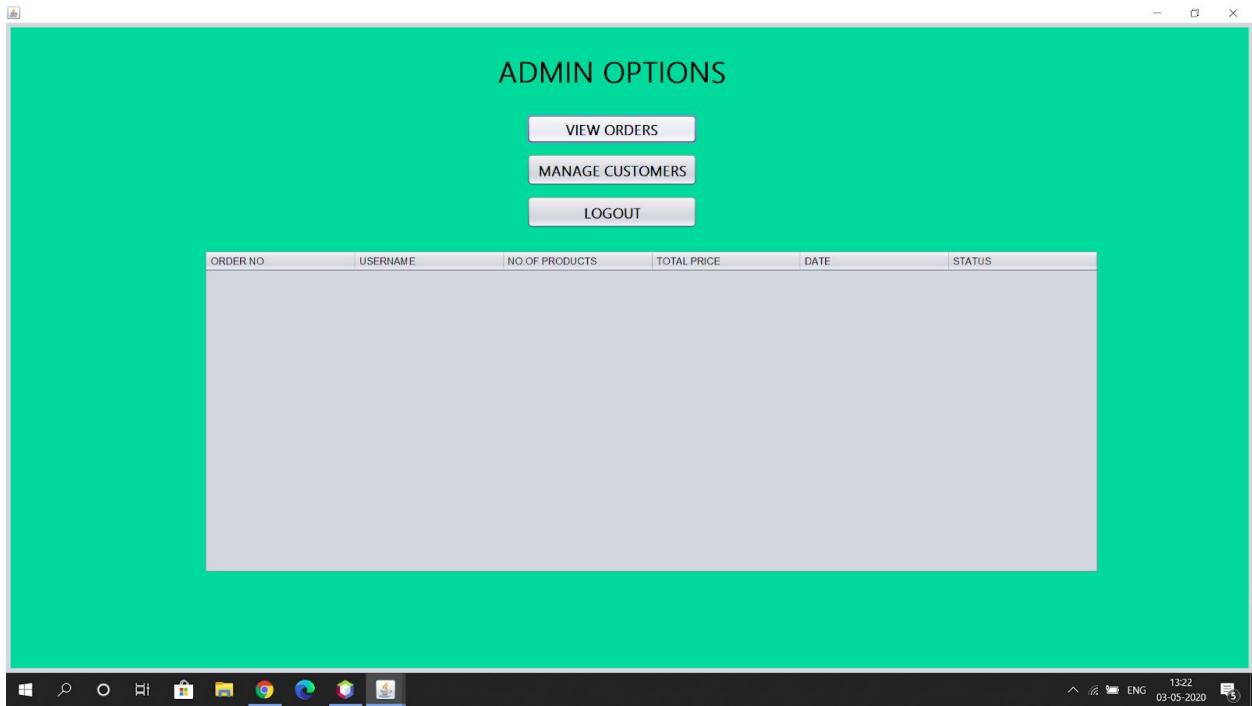
## Admin login module(admin\_login.java)->



If SUBMIT button is clicked->

```
try
{
    String name=jTextField1.getText();
    String pw=jPasswordField1.getText();
    String msg="select * from admin where aid='"+name+"' and apw='"+pw+"'";
    PreparedStatement s=conn.prepareStatement(msg);
    ResultSet res = s.executeQuery();
    if(res.next()){
        this.dispose();
        a_op a=new a_op();
        a.setVisible(true);
        s.close();
        res.close();
    }
    else
    {
        JOptionPane.showMessageDialog(null,"INVALID ADMIN DETAILS");
        this.dispose();
    }
}
catch(SQLException e)
{e.printStackTrace();}
```

## Admin options module(a\_op.java)->



Once the module is opened the following code is executed->

```
public a_op()
```

```
    initComponents();
    jTable1.setVisible(false);
}
```

If 'VIEW ORDERS' button is clicked->

```
DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();
```

```
String s2,s5,s6;
```

```
int s1,s3,s4;
```

```
try
```

```
{
```

```
    String msg="select * from order_c";
```

```
    PreparedStatement ss=conn.prepareStatement(msg);
```

```
    ResultSet ress=ss.executeQuery();
```

```
    while(ress.next())
```

```
{
```

```
    s1=ress.getInt(1);
```

```
    s2=ress.getString(2);
```

```
    s3=ress.getInt(3);
```

```
    s4=ress.getInt(4);
```

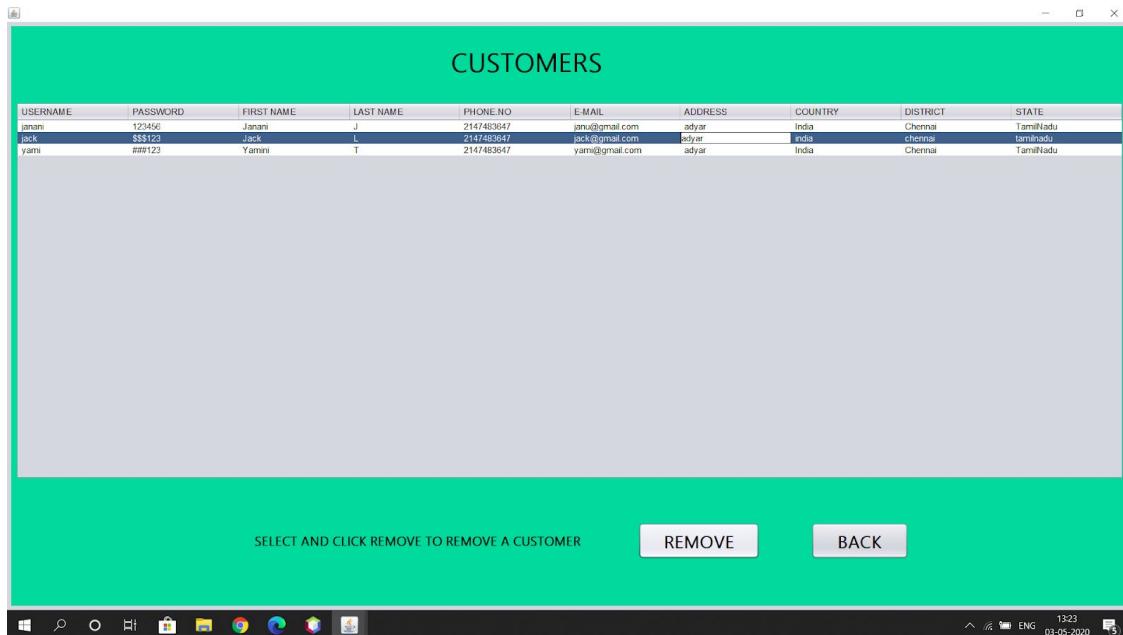
```
    s5=ress.getString(5);
```

```

s6=ress.getString(6);
Object data[]={s1,s2,s3,s4,s5,s6};
tb.addRow(data);
}
ss.close();
ress.close();
}
catch(SQLException e)
{e.printStackTrace(); }
If 'MANAGE CUSTOMERS' button is clicked->
remove r=new remove();
r.setVisible(true);
If LOGOUT button is clicked->
this.dispose();

```

### Remove customers module(remove.java)->



Once the module is opened the following code is executed->

```

public remove() {
    initComponents();
    DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();
    try{
        String msg="select * from customer";
        PreparedStatement ss=conn.prepareStatement(msg);
        ResultSet ress=ss.executeQuery();

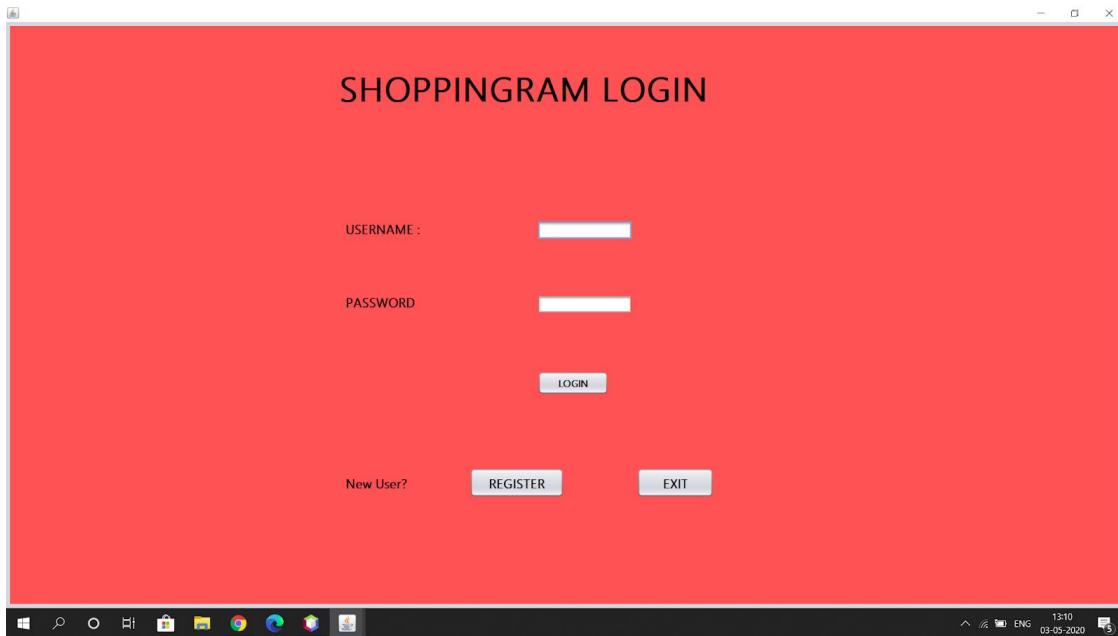
```

```

        while(ress.next())
        {
            Object
            data[]={ress.getString(1),ress.getString(2),ress.getString(3),ress.getString(4),ress.getInt
(5),ress.getString(6),ress.getString(7),ress.getString(8),ress.getString(9),ress.getString(
10)};
            tb.addRow(data);
        }
        ss.close();
        ress.close();
    }
    catch(SQLException e)
    {
        e.printStackTrace();
    }
If REMOVE button is clicked->
try
{
    DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();
    int r=jTable1.getSelectedRow();
    String id;
    id = tb.getValueAt(r,0).toString();
    tb.removeRow(jTable1.getSelectedRow());
    JOptionPane.showMessageDialog(null,"CUSTOMER REMOVED!");
    int sum=o;
    String msg="delete from customer where username='"+id+"'";
    PreparedStatement st=conn.prepareStatement(msg);
    int res=st.executeUpdate();
    st.close();
}
catch (SQLException ex) {ex.printStackTrace(); }
If BACK button is clicked->
this.dispose();

```

## Customer login module(cust\_login.java)->



Once the module is opened the following code is executed->

```
public c_op(String user)
{
    initComponents();
    jLabel2.setText("WELCOME "+user+"!");
    u=user;
}

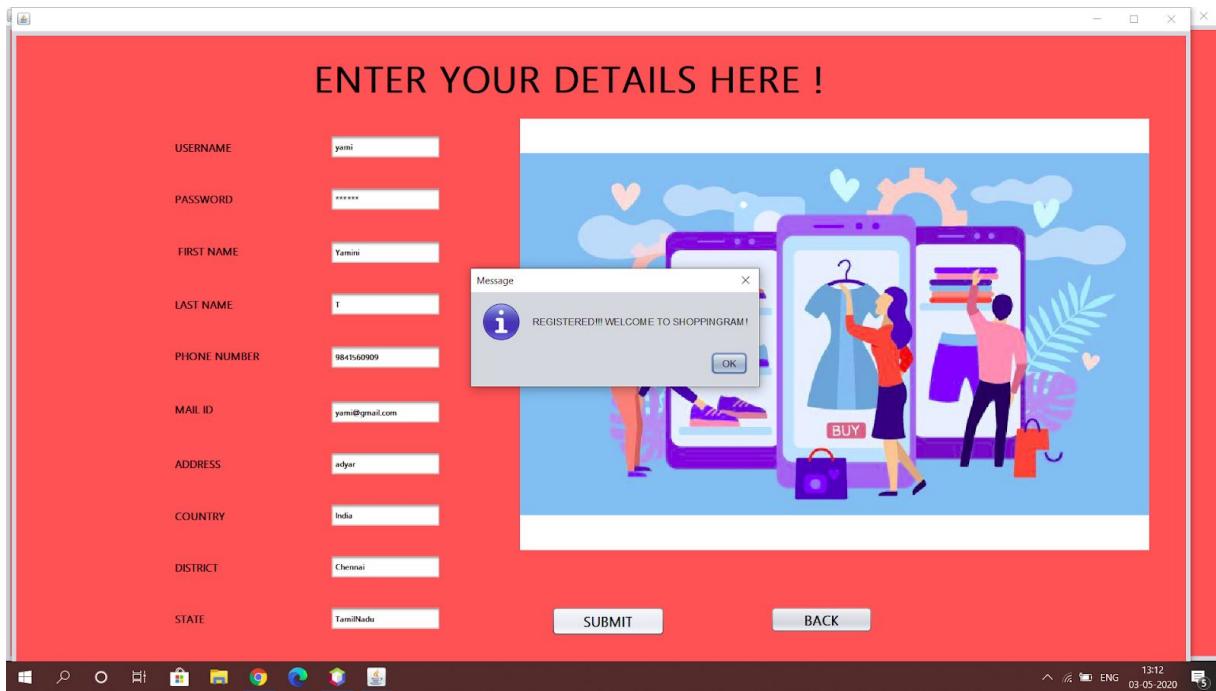
If LOGIN button is clicked->
try
{
    String name=jTextField1.getText();
    String pw=jPasswordField1.getText();
    String msg="select * from customer where username='"+name+"' and
              pw='"+pw+"'";
    PreparedStatement s=conn.prepareStatement(msg);
    ResultSet res = s.executeQuery();
    if(res.next()){
        this.dispose();
        s.close();
        res.close();
        c_op c=new c_op(name);
        c.setVisible(true); }
    else
```

```

{ JOptionPane.showMessageDialog(null,"INVALID LOGIN CREDENTIALS");
    this.dispose();}
}catch(SQLException e){e.printStackTrace(); }
If REGISTER button is clicked->
cust_reg cr = new cust_reg();
cr.setVisible(true);
If EXIT button is clicked->
this.dispose();

```

### Customer register module(cust\_reg.java)->



### If SUBMIT button is clicked->

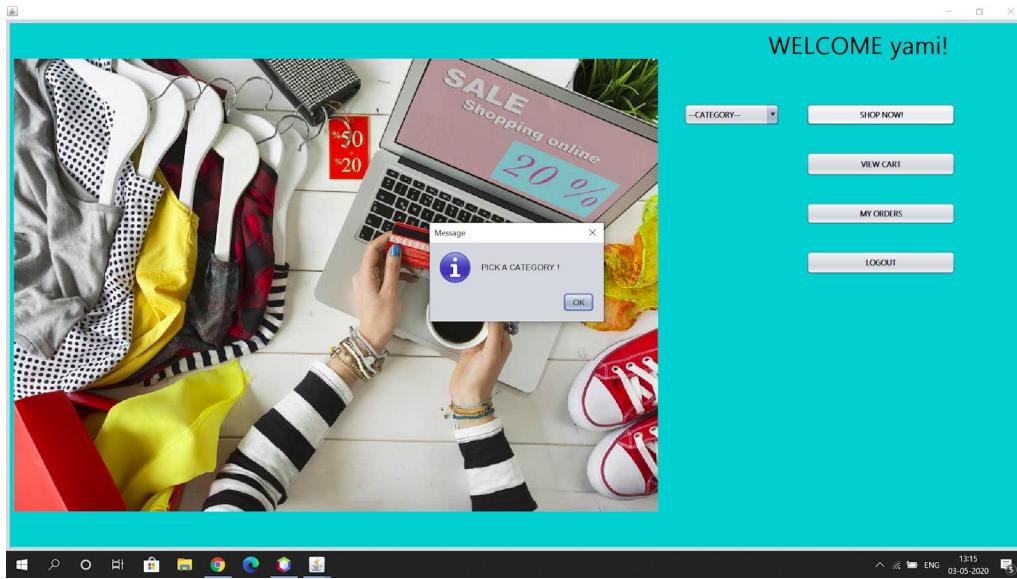
```

try{
    String name=jTextField1.getText();
    String msg="select * from customer where username='"+name+"'";
    PreparedStatement s=conn.prepareStatement(msg);
    ResultSet res = s.executeQuery();
    if(res.next()){
        JOptionPane.showMessageDialog(null,"USER ALREADY EXISTS");
        this.dispose();
        s.close();
        res.close();
    }
    else

```

```
{  
    PreparedStatement st=conn.prepareStatement("insert into customer  
values(?,?,?,?,?,?,?,?,?,?)");  
    st.setString(1,jTextField1.getText());  
    st.setString(2,jPasswordField1.getText());  
    st.setString(3,jTextField2.getText());  
    st.setString(4,jTextField3.getText());  
    st.setString(5,jTextField4.getText());  
    st.setString(6,jTextField5.getText());  
    st.setString(7,jTextField6.getText());  
    st.setString(8,jTextField7.getText());  
    st.setString(9,jTextField8.getText());  
    st.setString(10,jTextField9.getText());  
    int rs=st.executeUpdate();  
    System.out.println(rs+" status");  
    st.close();  
    JOptionPane.showMessageDialog(null,"REGISTERED!!! WELCOME TO  
SHOPPINGRAM!");  
    this.dispose();  
}  
}  
catch(SQLException e){e.printStackTrace();}  
If BACK button is clicked->  
this.dispose();
```

## Customer options module(c\_op.java)->



If ‘SHOP NOW!’ button is clicked->

```
if(jComboBox1.getSelectedItem().toString().equals("MEN"))
{
    men m=new men(u);
    m.setVisible(true);
}
else if(jComboBox1.getSelectedItem().toString().equals("WOMEN"))
{
    women w=new women(u);
    w.setVisible(true);
}
else if(jComboBox1.getSelectedItem().toString().equals("KIDS"))
{
    kids k=new kids(u);
    k.setVisible(true);
}
else
    JOptionPane.showMessageDialog(null,"PICK A CATEGORY !");

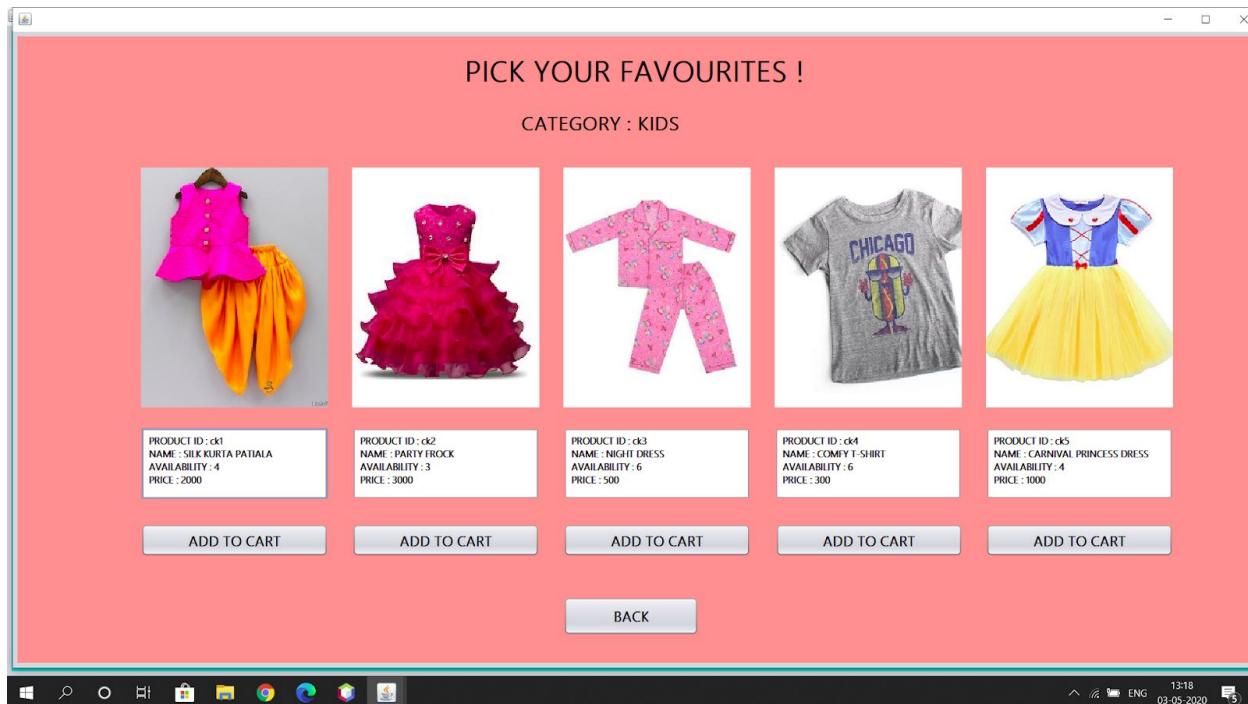
If ‘VIEW CART’ button is clicked->
cart ct=new cart(u);
ct.setVisible(true);

If ‘MY ORDERS’ button is clicked->
orders o=new orders(u);
o.setVisible(true);
```

If 'LOGOUT' button is clicked->

```
this.dispose();
```

Men,Women and Kids module->



Once the module is opened the following code is executed->

(similar for women and kids)

```
public men(String user) {  
    initComponents();  
    u=user;  
    try{  
        String msg="select * from product";  
        PreparedStatement ss=conn.prepareStatement(msg);  
        ResultSet ress=ss.executeQuery();  
        while(ress.next()){  
            if(ress.getString(1).equals("cm1"))  
                {jTextPane1.setText("PRODUCT ID : "+ress.getString(1)+"\nNAME :  
"+ress.getString(2)+"\nAVAILABILITY : "+ress.getInt(3)+"\nPRICE : "+ress.getInt(5));  
                jTextPane1.setEditable(false);}  
  
            if(ress.getString(1).equals("cm2"))  
                {jTextPane2.setText("PRODUCT ID : "+ress.getString(1)+"\nNAME :  
"+ress.getString(2)+"\nAVAILABILITY : "+ress.getInt(3)+"\nPRICE : "+ress.getInt(5));  
                jTextPane2.setEditable(false);}}}}}
```

```

jTextPane2.setEditable(false);}

if(ress.getString(1).equals("cm3"))
    {jTextPane3.setText("PRODUCT ID : "+ress.getString(1)+"\nNAME :
"+ress.getString(2)+"\nAVAILABILITY : "+ress.getInt(3)+"\nPRICE : "+ress.getInt(5));
jTextPane3.setEditable(false);}

if(ress.getString(1).equals("cm4"))
    {jTextPane4.setText("PRODUCT ID : "+ress.getString(1)+"\nNAME :
"+ress.getString(2)+"\nAVAILABILITY : "+ress.getInt(3)+"\nPRICE : "+ress.getInt(5));
jTextPane4.setEditable(false);}

if(ress.getString(1).equals("cm5"))
    {jTextPane5.setText("PRODUCT ID : "+ress.getString(1)+"\nNAME :
"+ress.getString(2)+"\nAVAILABILITY : "+ress.getInt(3)+"\nPRICE : "+ress.getInt(5));
jTextPane5.setEditable(false);}
} ss.close();
ress.close();
}

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

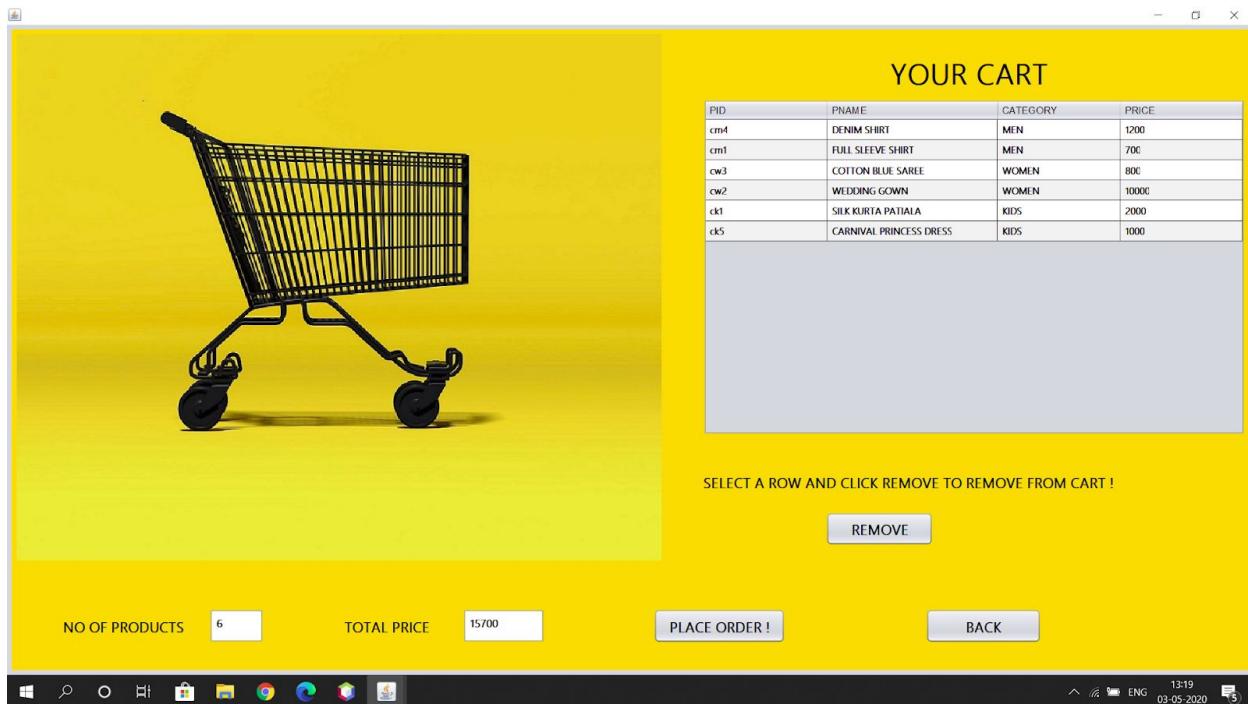
If 'ADD TO CART' button is clicked(this button inserts the user and corresponding product id into cart table)->
try{
    String msg="insert into cart values('"+u+"','cm1')";
    PreparedStatement ss=conn.prepareStatement(msg);
    ResultSet ress=ss.executeQuery();
    ss.close();
    ress.close();
    JOptionPane.showMessageDialog(null,"ADDED TO CART!");
}

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

If 'EXIT' button is clicked->
this.dispose();

```

## Cart module(cart.java)->



Once the module is opened the following code is executed->

```
public cart(String user) {
    u=user;
    initComponents();
    DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();
    TableColumnModel tc=jTable1.getColumnModel();
    tc.getColumn(0).setPreferredWidth(3);
    tc.getColumn(2).setPreferredWidth(3);
    tc.getColumn(3).setPreferredWidth(3);
    String s1,s2,s3;
    int s4,c=0,s=0;
    try{
        String msg="select p.pid,p.pname,p.category,p.price from cart c,product p where
c.pid=p.pid and c.username='"+u+"'";
        PreparedStatement ss=conn.prepareStatement(msg);
        ResultSet ress=ss.executeQuery();
        while(ress.next())
        {
            s1=ress.getString(1);
            s2=ress.getString(2);
            s3=ress.getString(3);
    
```

```

s4=ress.getInt(4);
c=c+1;
s=s+s4;
Object data[]={s1,s2,s3,s4};
tb.addRow(data);
}
ss.close();
ress.close();
jTextPane1.setText(String.valueOf(c));
jTextPane2.setText(String.valueOf(s));
}
catch(SQLException e)
{
    e.printStackTrace();
}
If 'REMOVE' button is clicked->
try {
    DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();
    int r=jTable1.getSelectedRow();
    String id;
    id = tb.getValueAt(r,0).toString();
    tb.removeRow(jTable1.getSelectedRow());
    JOptionPane.showMessageDialog(null,"ITEM REMOVED!");
    int sum=0;
    String msg="delete from cart where pid='"+id+"'";
    PreparedStatement st=conn.prepareStatement(msg);
    int res=st.executeUpdate();
    st.close();
    for(int i=0;i<jTable1.getRowCount();i++)
    {
        int p=Integer.parseInt((String) jTable1.getValueAt(i,3).toString());
        sum=sum+p;
    }
    jTextPane1.setText(String.valueOf(jTable1.getRowCount()));
    jTextPane2.setText(String.valueOf(sum));
}
catch (SQLException ex) {
    ex.printStackTrace();
}
If 'PLACE ORDER' button is clicked->
try{

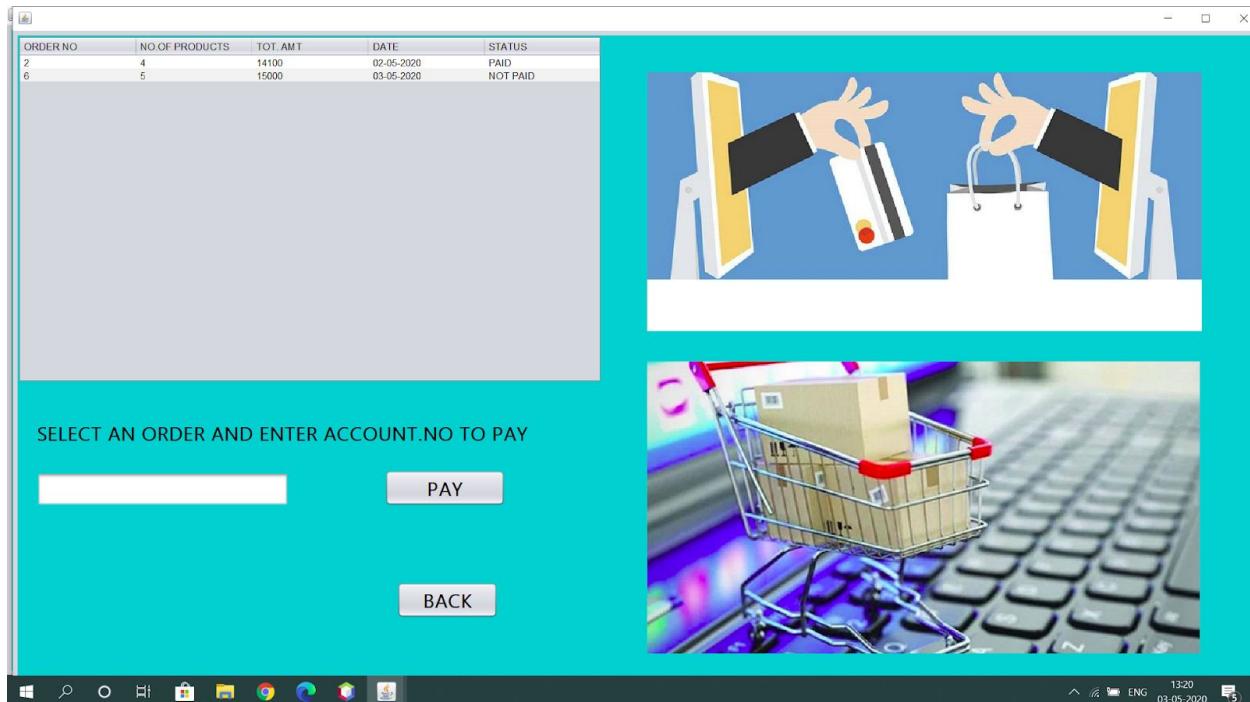
```

```

int r=0;
String msg="select count(*) from order_c";
PreparedStatement st=conn.prepareStatement(msg);
ResultSet res=st.executeQuery();
if(res.next())
{
    r=res.getInt(1);
}
r=r+1;
st.close();
res.close();
int x=Integer.parseInt(jTextPane1.getText());
int y=Integer.parseInt(jTextPane2.getText());
Date d=new Date();
SimpleDateFormat df=new SimpleDateFormat("dd-MM-yyyy");
PreparedStatement s=conn.prepareStatement("insert into order_c
values(?,?,?,?,?,?)");
s.setInt(1, r);
s.setString(2, u);
s.setInt(3,x);
s.setInt(4,y);
s.setString(5,df.format(d).toString());
s.setString(6,"NOT PAID");
int rs=s.executeUpdate();
JOptionPane.showMessageDialog(null,"ORDER PLACED !");
s.close();
PreparedStatement stt=conn.prepareStatement("delete cart");
int result=stt.executeUpdate();
stt.close();
jTable1.setModel(new DefaultTableModel(null,new
String[]{"PID","PNAME","CATEGORY","PRICE"}));
If BACK button is clicked->
this.dispose();

```

## Orders module(order.java)->



Once the module is opened the following code is executed->

```
public orders(String user) {  
    initComponents();  
    u=user;  
    DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();  
    String s4,s5;  
    int s1,s2,s3;  
    try{  
        String msg="select no,npd,totp,d,status from order_c where username='"+u+"';  
        PreparedStatement ss=conn.prepareStatement(msg);  
        ResultSet ress=ss.executeQuery();  
        while(ress.next())  
        {  
            s1=ress.getInt(1);  
            s2=ress.getInt(2);  
            s3=ress.getInt(3);  
            s4=ress.getString(4);  
            s5=ress.getString(5);  
            Object data[]={s1,s2,s3,s4,s5};  
            System.out.println(s1+" "+s2+" "+s3+" "+s4+" "+s5);  
            tb.addRow(data);  
        }  
    }  
}
```

```

    }
    ss.close();
    ress.close();
}
catch(SQLException e)
{
    e.printStackTrace();
}

If BACK button is clicked->
this.dispose();

If PAY button is clicked->
try {
    DefaultTableModel tb=(DefaultTableModel)jTable1.getModel();
    int r=jTable1.getSelectedRow();
    String id;
    id = tb.getValueAt(r,4).toString();
    int o_no=Integer.parseInt((String) tb.getValueAt(r,0).toString());
    if(id.equals("NOT PAID "))
    {
        String n=jTextField1.getText().toString();
        if(n.length()===12)
        {
            tb.setValueAt("PAID",r,4);
            JOptionPane.showMessageDialog(null,"PAID!");
        }
        else
        {
            JOptionPane.showMessageDialog(null,"INVALID CARD NO(NOT IN 12
DIGITS)");
        }
    }
    else
        JOptionPane.showMessageDialog(null,"YOU HAVE ALREADY PAID!");
    String msg="update order_c set status='PAID' where no="+o_no;
    PreparedStatement st=conn.prepareStatement(msg);
    int res=st.executeUpdate();
    st.close();
}
catch (SQLException ex) {
    ex.printStackTrace();
}

```