SWEET SHOP MANAGEMENT SYSTEM (SSMS)

THANGARAJ M (23322038)

RAVICHANDHIRAN B (23322039)

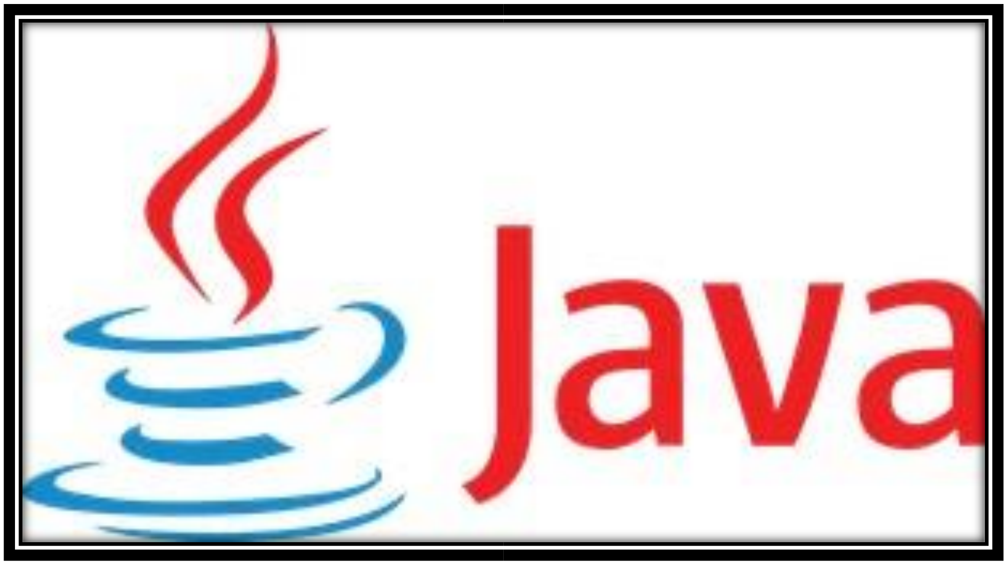
INTRODUCTION:

A Sweet Shop Management System using Swing in Java is an ideal project for students or developers interested in gaining practical experience with Java GUI applications. Swing, a part of Java's Standard Library, provides a rich set of components for building user interfaces, making it well-suited for creating a management system for a sweet shop. This project aims to streamline various operations of a sweet shop, such as inventory management, sales tracking, and customer management. The user interface, developed using Swing components like JFrame, JPanel, JTable, and JButton, ensures an interactive and user-friendly experience. By undertaking this project, developers can enhance their understanding of Java Swing, event handling, and database integration, while also addressing realworld problems faced by small retail businesses. Overall, this mini-project not only bolsters programming skills but also provides valuable insights into effective small business management through software solutions.

SOFTWARE DESCRIPTION:

1. Java
2. JDBC
3. MS Access

1. JAVA:



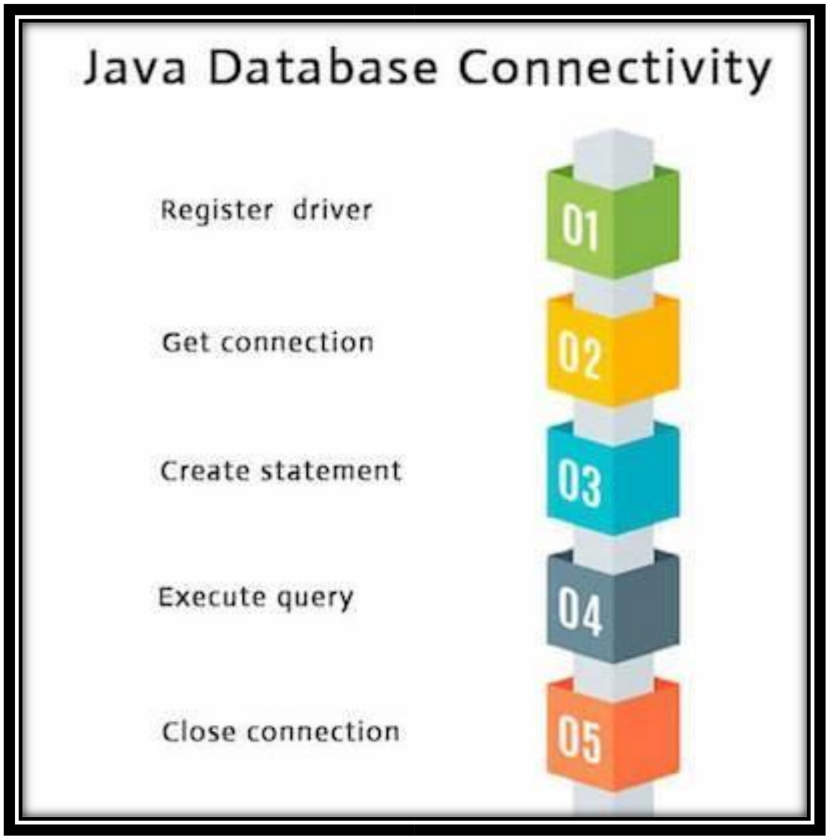
Java is a high-level, class-based, object-oriented programming language designed to have as few implementation dependencies as possible. Created by Sun Microsystems in 1995, Java has since become one of the most popular and widely used programming languages in the world, particularly known for its portability across platforms. This "write once, run anywhere" (WORA) capability means that compiled Java code can run on any device that supports the Java Virtual Machine (JVM), making it highly versatile.

Java's syntax is similar to C++, which makes it relatively easy for programmers familiar with that language to learn. It supports key objectoriented programming concepts like inheritance, polymorphism, encapsulation, and abstraction, allowing developers to create modular and reusable code. Additionally, Java includes a vast standard library that provides a rich set of pre-built classes and methods to perform a wide range of tasks, from data structures and algorithms to graphical user interface creation and network communication.

One of the defining features of Java is its robust memory management, primarily through the use of an automatic garbage collector that helps prevent memory leaks and other related issues. This, along with its strong typechecking mechanism and exception handling features, contributes to Java's reputation for reliability and security.

Java is widely used in various domains, including web development (via frameworks like Spring), enterprise applications, mobile applications (especially Android), and large-scale systems. Its continued evolution, driven by a strong community and regular updates from Oracle Corporation, ensures that Java remains a relevant and powerful tool in the ever-changing landscape of technology.

2. JDBC:



Java Database Connectivity (JDBC) is a Java-based API that allows Java applications to interact with a variety of databases. Introduced as part of the Java Standard Edition, JDBC provides a standard interface for connecting to relational databases, executing SQL queries, and retrieving results. It plays a crucial role in Java enterprise applications, enabling seamless database integration and manipulation.

JDBC is designed around the concept of drivers that act as a bridge between the Java application and the database. Each database vendor provides its own JDBC driver, ensuring compatibility and optimized performance for their specific database systems. The main components of JDBC include the `DriverManager`, `Connection`, `Statement`, `PreparedStatement`, `CallableStatement`, and `ResultSet` classes, each serving distinct functions in the database interaction process.

DriverManager:

Manages a list of database drivers and establishes a connection to a database.

Connection:

Represents an active connection to a database, through which SQL commands are sent.

Statement:

Used for executing static SQL queries and retrieving results.

PreparedStatement:

Extends the functionality of `Statement` by allowing precompiled SQL queries with parameter placeholders, enhancing performance and security by preventing SQL injection.

CallableStatement:

Used to execute stored procedures in the database.

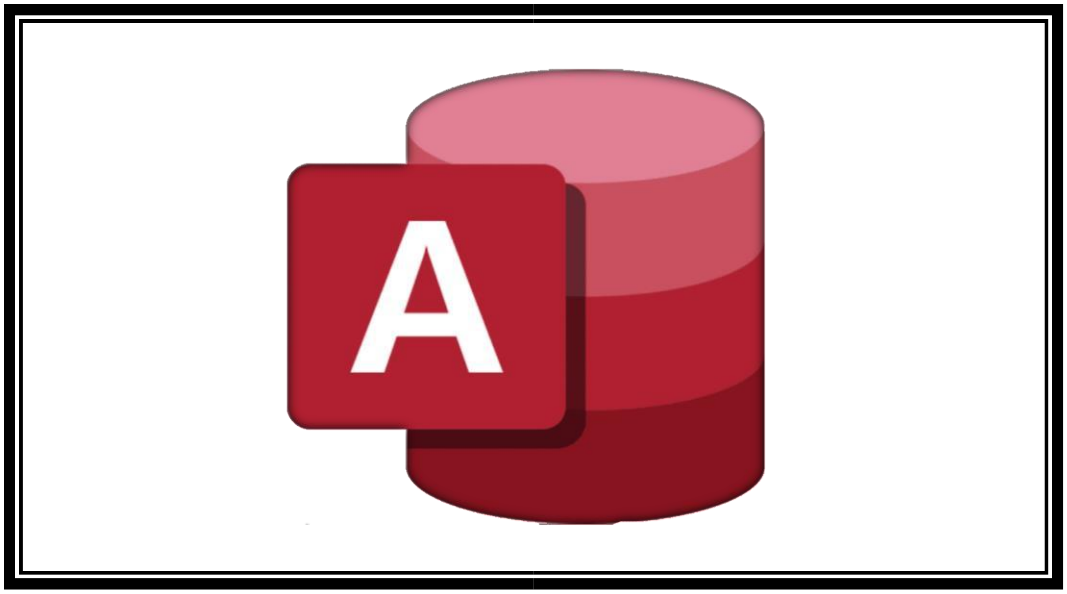
ResultSet:

Represents the result set of a query, allowing navigation and retrieval of query results.

JDBC supports both basic and advanced database operations, including transaction management, batch updates, and metadata retrieval. Its ability to interact with different types of databases, from open-source systems like MySQL and PostgreSQL to commercial solutions like Oracle and SQL Server, makes it a versatile choice for developers.

By providing a consistent and easy-to-use interface for database access, JDBC abstracts the complexity of database interactions, allowing developers to focus on the logic and functionality of their applications. This abstraction, combined with Java's portability, ensures that applications using JDBC can run on any platform that supports Java, further emphasizing Java's "write once, run anywhere" philosophy.

3. MS ACCESS:



Microsoft Access (MS Access) is a powerful relational database management system (RDBMS) developed by Microsoft. It combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools. MS Access is part of the Microsoft 365 suite and is particularly popular among small to medium-sized businesses, educational institutions, and individual users for its ease of use and robust functionality.

MS Access provides a comprehensive set of tools for creating and managing databases. Users can build tables to store data, forms for data entry, queries to retrieve specific information, and reports to present data in a structured format. The intuitive design interface allows users to quickly design and implement databases without requiring extensive programming knowledge, making it accessible to both novice and experienced users.

KEY FEATURE OF MS ACCESS INCLUDE:

TABLES:

Used to store data in rows and columns, similar to a spreadsheet, but with the added capabilities of a relational database, such as enforcing data integrity through primary and foreign keys.

QUERIES:

Powerful tools to retrieve and manipulate data. Access supports both simple queries, created through a visual query builder, and complex SQL-based queries for advanced data processing.

FORMS:

Customizable interfaces for data entry and navigation, allowing users to interact with data in a more user-friendly manner.

REPORTS:

Tools to create printed or electronic reports for data analysis and presentation, with a variety of formatting options to enhance readability.

MACROS:

Simplified programming language to automate repetitive tasks and add functionality without writing extensive code.

VBA (Visual Basic for Applications):

A robust programming environment for advanced users to write custom code, providing greater control over database operations and user interface customization.

MS Access databases can be easily shared across a network, enabling multiple users to access and update data simultaneously. Additionally, Access can integrate with other Microsoft applications, such as Excel and SharePoint, as well as various external data sources through ODBC (Open Database Connectivity).

While MS Access is not designed for high-volume, enterprise-level applications, it is an excellent solution for smaller-scale applications that require rapid development and deployment. Its combination of ease of use, powerful features, and integration capabilities makes MS Access a versatile tool for a wide range of database management needs.

4. SOURCE CODE:

login.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
class login extends JFrame implements ActionListener{
   JLabel l1,l2,l3;
   JTextField tf1,tf2;
   JButton         btn1,btn2;
String uname,pword;
   login(){
      Font ft = new Font("serif",Font.ITALIC,25);
      this.getContentPane().setBackground(new Color(250, 170, 68));
l1 = new JLabel("Username");       l2 = new JLabel("Password");
      l3 = new JLabel("GfK sweets");l3.setFont(ft);       tf1 = new
JTextField();       tf2 = new JTextField();       btn1 = new
JButton("Login");   btn2 = new JButton("Sign up");
l1.setBounds(50,50,150,30);l1.setForeground(new Color(0,0,0));
l2.setBounds(50,100,150,30);l2.setForeground(new Color(0,0,0));
l3.setBounds(100,8,200,40);l3.setForeground(new Color(8, 15, 174));
tf1.setBounds(150,50,150,25);        tf2.setBounds(150,100,150,25);
       btn1.setBounds(50,150,100,30);              btn2.setBounds(180,150,100,30);
btn1.addActionListener(this); btn2.addActionListener(this);
this.add(l1);this.add(l2);this.add(l3);this.add(tf1);this.add(tf2);this.add(btn1);this.a
dd(btn2);
      this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setTitle("Login Page");       this.setSize(350,250);
this.setLayout(null);
      this.setLocationRelativeTo(null);
   }
   public   void   actionPerformed(ActionEvent   ae){
try{
            uname = tf1.getText();
pword = tf2.getText();
Class.forName("net.ucanaccess.jdbc.U
canaccessDriver");
            Connection con =
DriverManager.getConnection("jdbc:ucanaccess://C://Users//ADMIN//Desktop//s
weetshop//sssm.accdb");
                PreparedStatement ps = con.prepareStatement("select * from account
where uname = '" + uname + "' and pword = '" + pword + "'");
                ResultSet rs = ps.executeQuery();
if(ae.getSource() == btn1){
if(rs.next()){                        items itm =
new items();
itm.setVisible(true);
this.dispose();
                    }else if(uname.equals("admin") & pword.equals("admin")){
items itm = new items();                          itm.pl3.add(itm.b2);
itm.setVisible(true);
                    } else{
                        JOptionPane.showMessageDialog(null,"Invalie User :(");
                    }
                }else if(ae.getSource() == btn2){
signup sign = new signup();
sign.setVisible(true);                    this.dispose();
                }
            }
            catch (ClassNotFoundException cnfex){
                System.out.println(cnfex);
            }
            catch(SQLException sqe){
                System.out.println(sqe);
            }
    }
}

class log {
    public static void main (String args []){
login logiin = new login();
logiin.setVisible(true);
    }
}
```

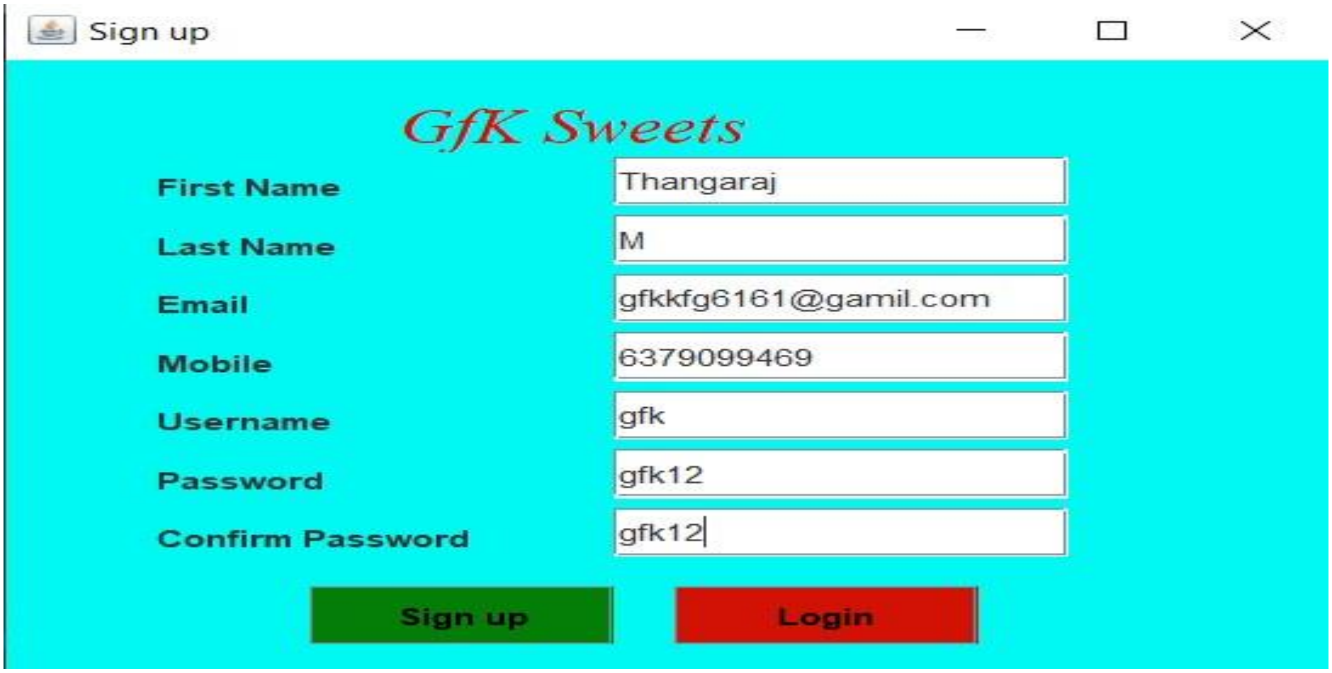
```
2. items.java import
javax.swing.*; import
java.awt.*; import
java.awt.event.*;
class items extends JFrame implements ItemListener,ActionListener{
    JRadioButton ghrb,milkswt,dfswt,krpatti,casew;
    JPanel pl1,pl2,pl3;    JButton b1,b2;    ButtonGroup bg;
ImageIcon ghee,milk,dryft,karupatti,cashew;     items(){
        this.setTitle("Sweets");        ghee = new ImageIcon("ghee.jpeg");
        Image im1 = ghee.getImage().getScaledInstance(130, 80,
Image.SCALE_REPLICATE);             ImageIcon  imic1  =  new  ImageIcon(im1);
milk = new ImageIcon("milk.jpeg");        Image im2 =
milk.getImage().getScaledInstance(130, 80, Image.SCALE_REPLICATE);
        ImageIcon imic2 = new ImageIcon(im2);        dryft = new
ImageIcon("dryfruit.jpeg");        bg = new ButtonGroup();         ghrb = new
JRadioButton(imic1);ghrb.addItemListener(this);bg.add(ghrb);         milkswt =
new
JRadioButton(imic2);milkswt.addItemListener(this);bg.add(milkswt);         dfswt
= new JRadioButton(imic3);dfswt.addItemListener(this);bg.add(dfswt);
krpatti = new
RadioButton(imic4);krpatti.addItemListener(this);bg.add(krpatti);
casew = new
RadioButton(imic5);casew.addItemListener(this);bg.add(casew);
pl1 = new JPanel(new FlowLayout(FlowLayout.CENTER));
pl1.add(ghrb);pl1.add(milkswt);pl1.add(dfswt);pl1.add(krpatti);pl1.add(casew);pl1
```

```
.setSize(740,100);                    pl2 = new JPanel(new GridLayout(5,1,10,10));
pl2.setSize(500,500);       pl3 = new JPanel(new
FlowLayout(FlowLayout.CENTER));                    b1 = new JButton("Go to
cart");b1.addActionListener(this);
pl3.setSize(700,50);pl3.add(b1);b1.setForeground(new
Color(0,0,0));b1.setBackground(new Color(5, 126, 7));       b2 = new
JButton("Account details");b2.addActionListener(this);
pl1.setBounds(5,5,740,100);pl2.setBounds(5,105,740,530);pl3.setBounds(5,650,74
0,50);
this.add(pl1,BorderLayout.NORTH);this.add(pl2,BorderLayout.CENTER);this.add
(pl3,BorderLayout.SOUTH);this.setSize(780,750);this.setLayout(null);

this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

this.setLocationRelativeTo(null);        this.setVisible(true);

   }

   public void itemStateChanged(ItemEvent ie){        ghee ghswt = new ghee();

milk mkswt = new milk();    dfsweet dfswt1 = new dfsweet();       karupatti kp =

new karupatti();     cashew csw = new cashew();        if(ghrb.isSelected()){

pl2.removeAll();

pl2.add(ghswt.mysurpa);pl2.add(ghswt.boondhiladdu);pl2.add(ghswt.athirasam);pl
2.add(ghswt.badusha);
pl2.add(ghswt.modhiladdu);pl2.add(ghswt.chandrakala);pl2.add(ghswt.jangri);pl2.
add(ghswt.wheathalwa);
pl2.add(ghswt.apsihalwa);pl2.add(ghswt.sonpapadi); pl2.revalidate();

pl2.repaint();

      }        else if(milkswt.isSelected()){                        pl2.removeAll();

pl2.add(mkswt.thiratipal);pl2.add(mkswt.milkpeda);pl2.add(mkswt.kesarpeda);pl2

. add(mkswt.vanillaburfi);

pl2.add(mkswt.kesarburfi);pl2.add(mkswt.milkapple);pl2.add(mkswt.milktomato);

pl2.revalidate();

         pl2.repaint();
```

```
        }       else if(dfswt.isSelected()){          pl2.removeAll();
pl2.add(dfswt1.badamhalwa);pl2.add(dfswt1.bombayhalwa);pl2.add(dfswt1.apsiha
lwa);          pl2.revalidate();          pl2.repaint();

        }                            else    if(krpatti.isSelected()){

pl2.removeAll();

pl2.add(kp.kpbadamhalwa);pl2.add(kp.kpkajukatli);                    pl2.revalidate();
pl2.repaint();

        }

        else if(casew.isSelected()){

pl2.removeAll();pl2.add(csw.kajukatli);pl2.add(csw.cashewbites);

pl2.add(csw.cashewroll);pl2.add(csw.figkatli);pl2.add(csw.cashewcrumble);

pl2.revalidate();pl2.repaint();

        }

    }

    public void actionPerformed(ActionEvent ae){

if(ae.getSource() == b1){           cart ct = new

cart(); ct.setVisible(true);           this.dispose();

        }

        else if(ae.getSource() == b2){

account act = new account();

act.setVisible(true);

        }

    }

}
```

5. SCREENSHOT:

1. SIGN-UP:



Sign up

GfK Sweets

First Name Thangaraj

Last Name M

Email gfkkfg6161@gamil.com

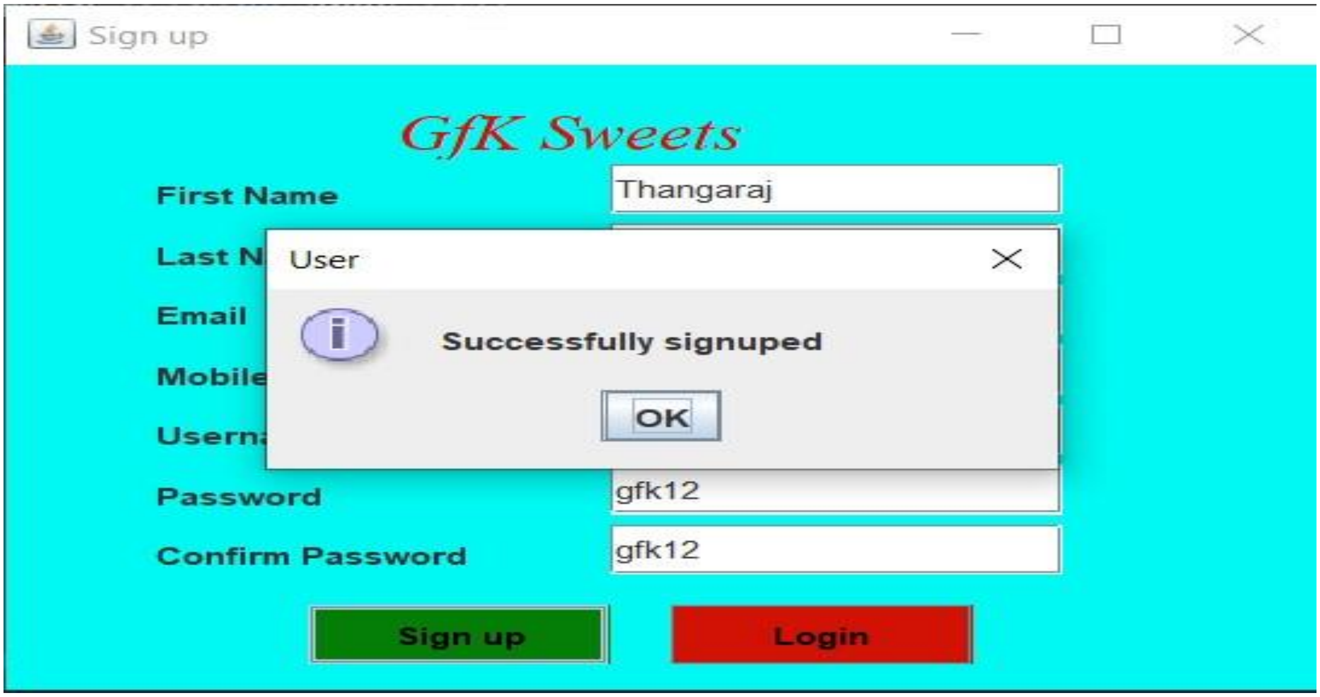
Mobile 6379099469

Username gfk

Password gfk12

Confirm Password gfk12

2. SIGN-UP MESSAGE:



Sign up

GfK Sweets

First Name Thangaraj

Last N

Email

Mobile

Usern

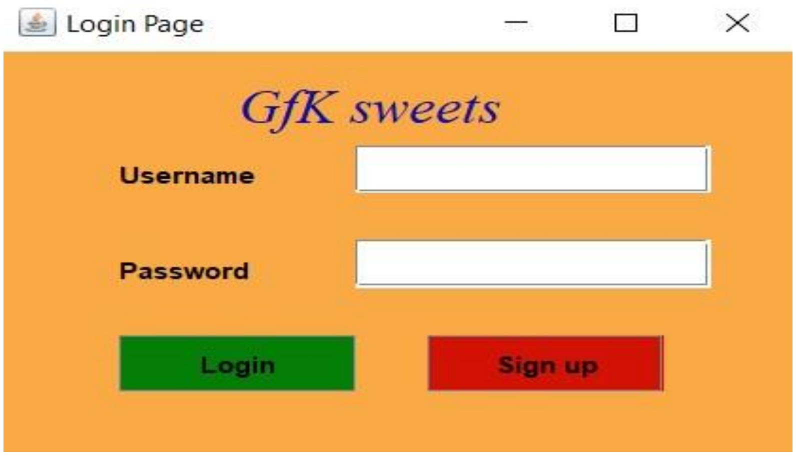
Password gfk12

Confirm Password gfk12

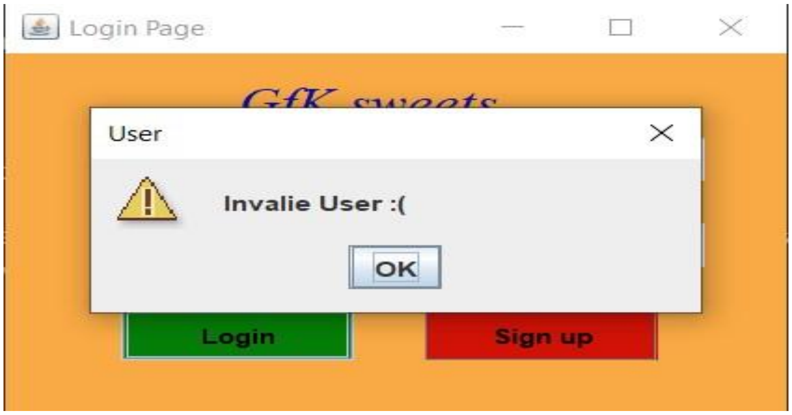
User

Successfully signuped

3. LOGIN:

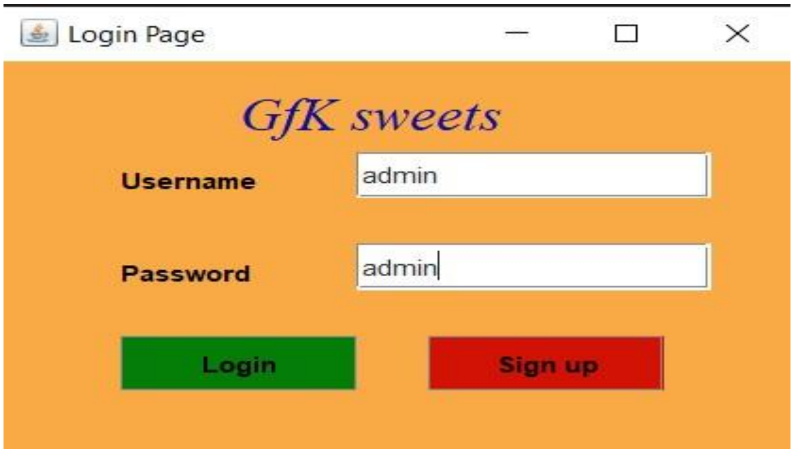


4. LOGIN MESSAGE:



Invalie User :(

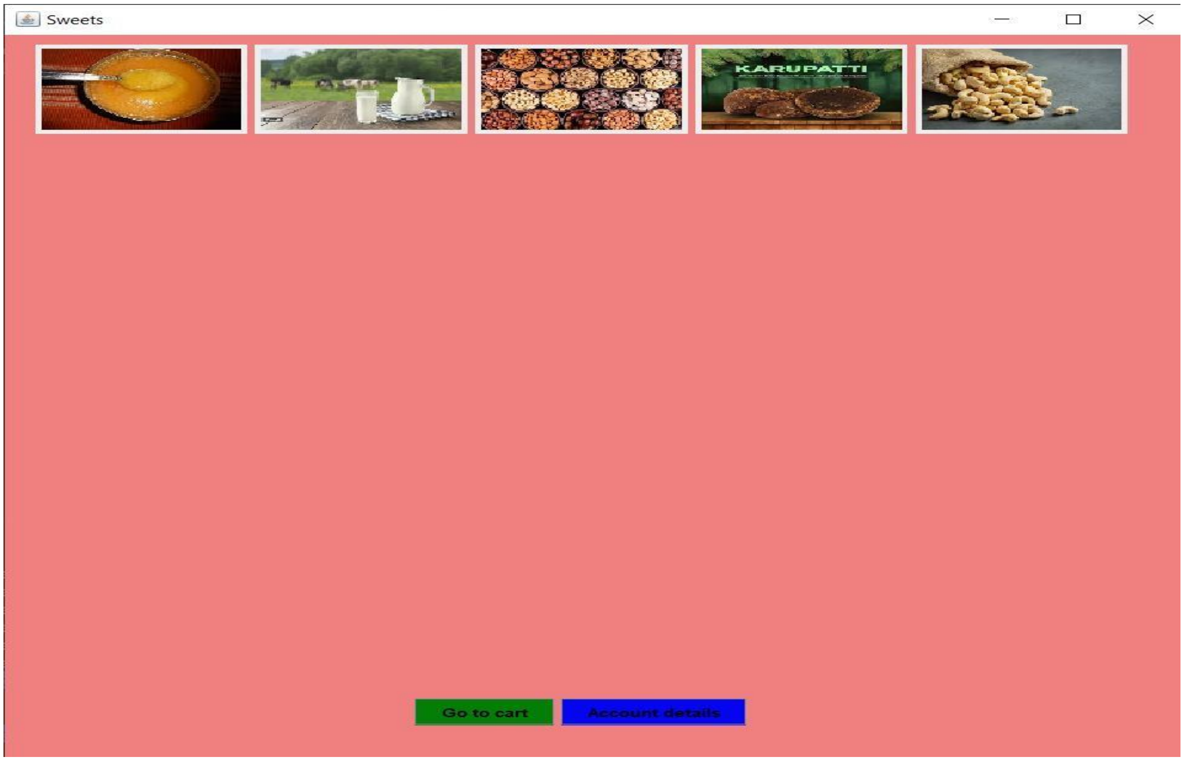
5. ADMIN LOGIN:



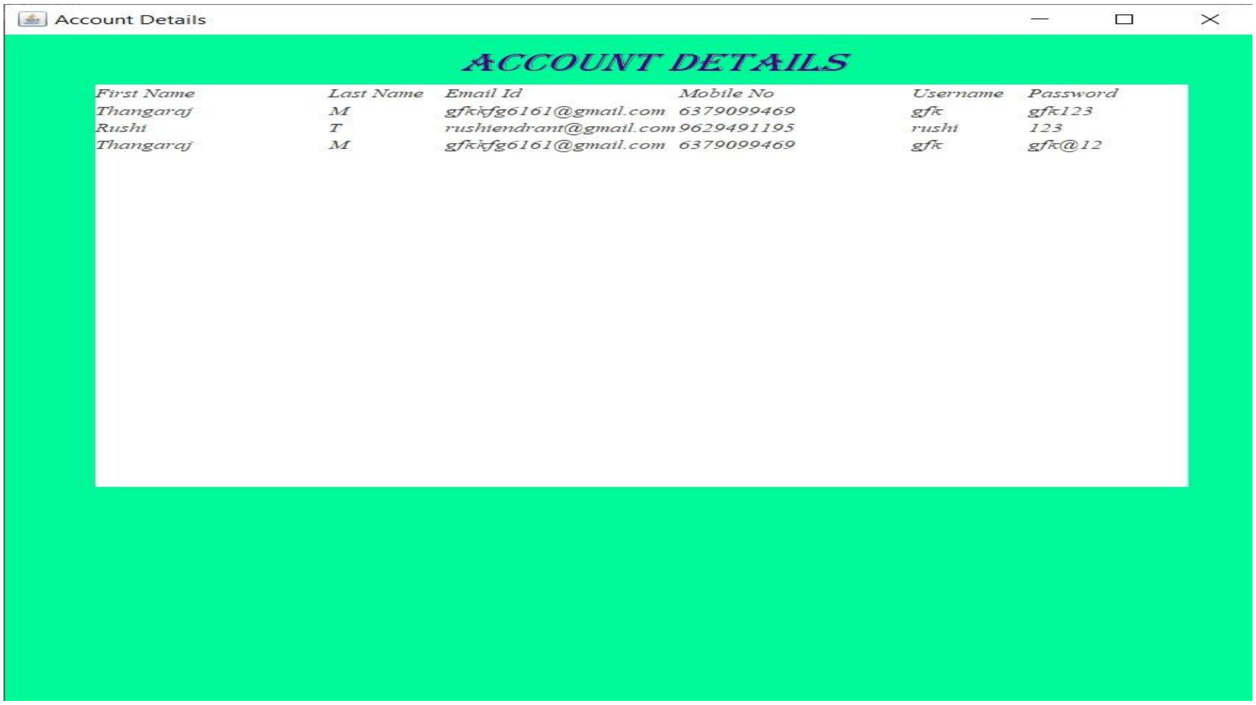
Username admin

Password admin

6. ADMIN PAGE:



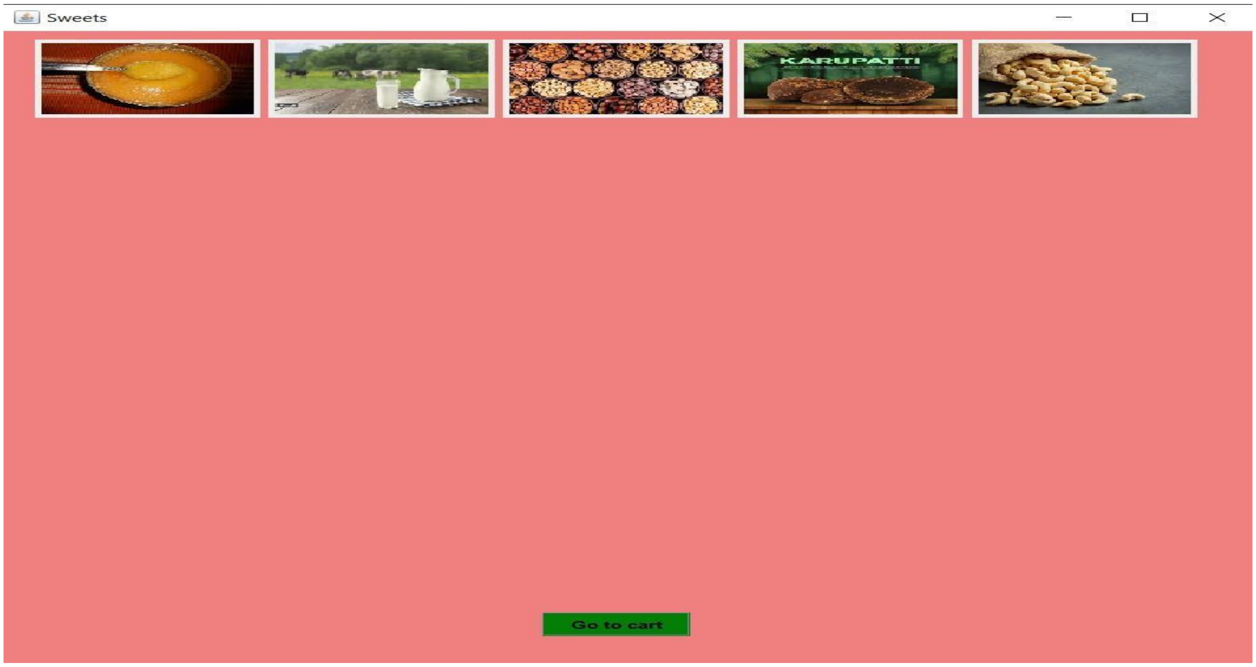
7. ADDOUNT DETAILS:



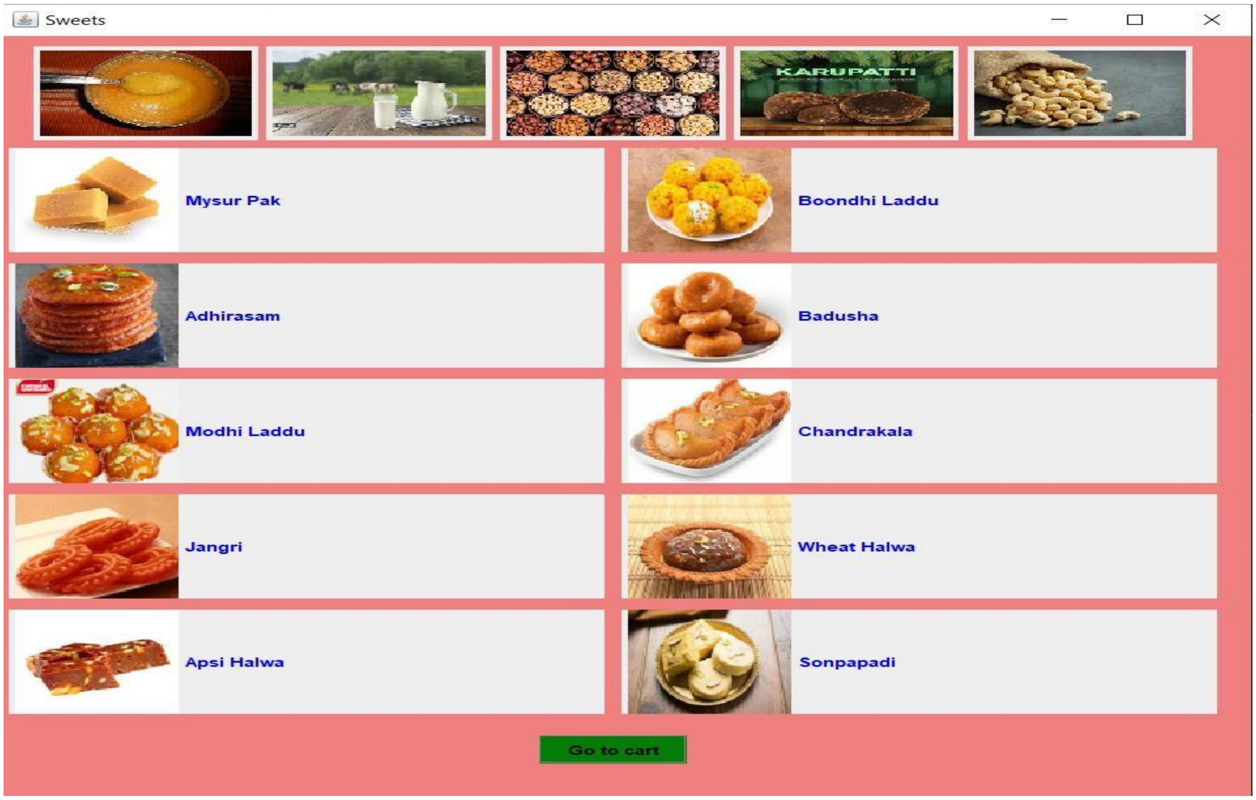
ACCOUNT DETAILS

First Name	Last Name	Email Id	Mobile No	Username	Password
Thangaraj	M	gfkkfg6161@gmail.com	6379099469	gfk	gfk123
Rushi	T	rushiendrant@gmail.com	9629491195	rushi	123
Thangaraj	M	gfkkfg6161@gmail.com	6379099469	gfk	gfk@12

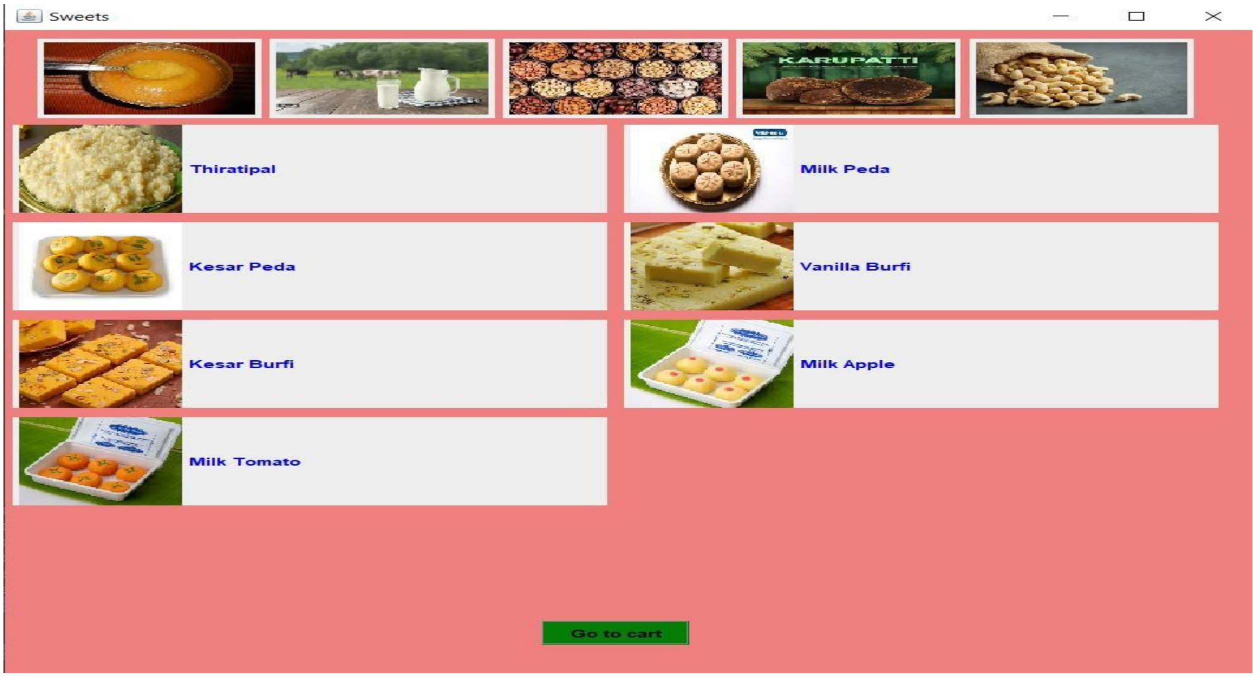
8. USER LOGIN:



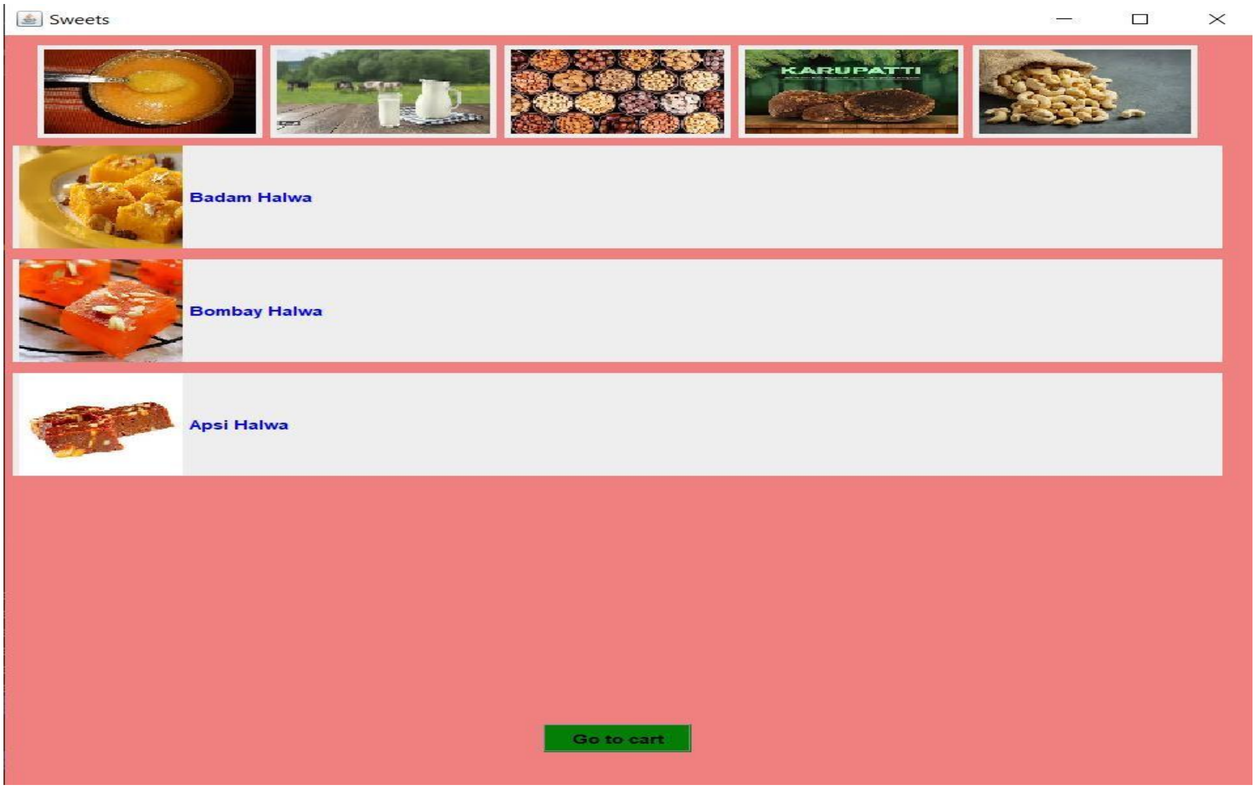
9. GHEE SWEET:



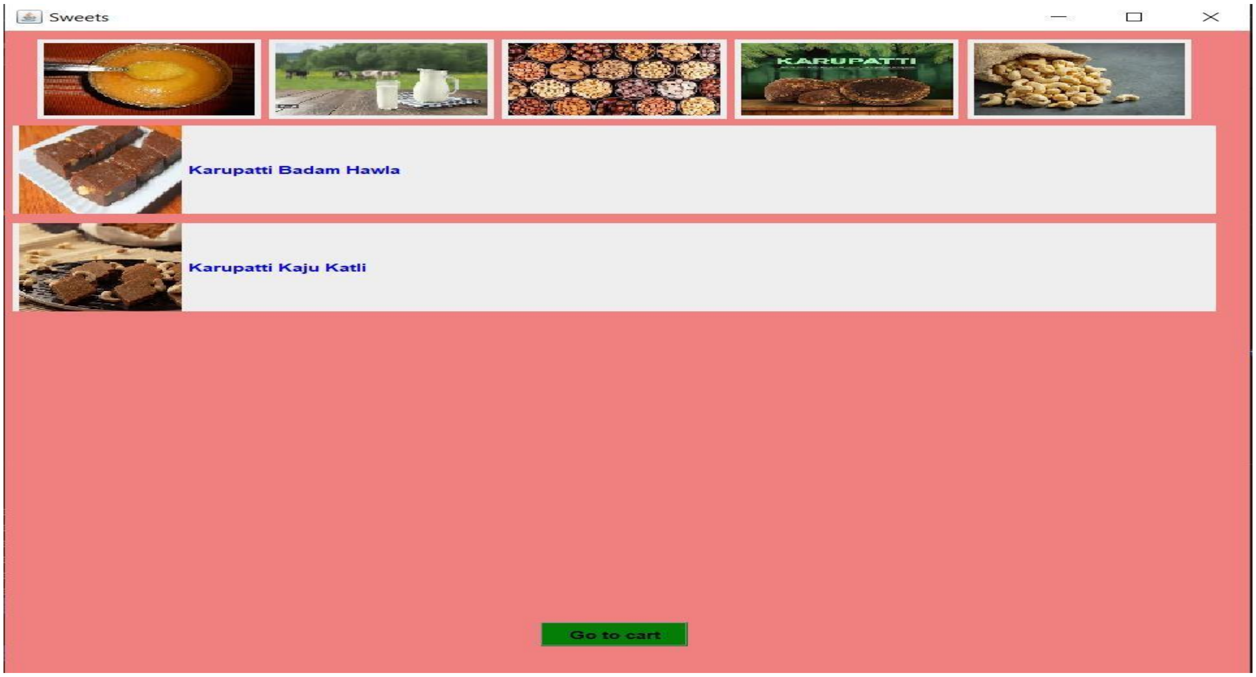
10. MILK SWEET:



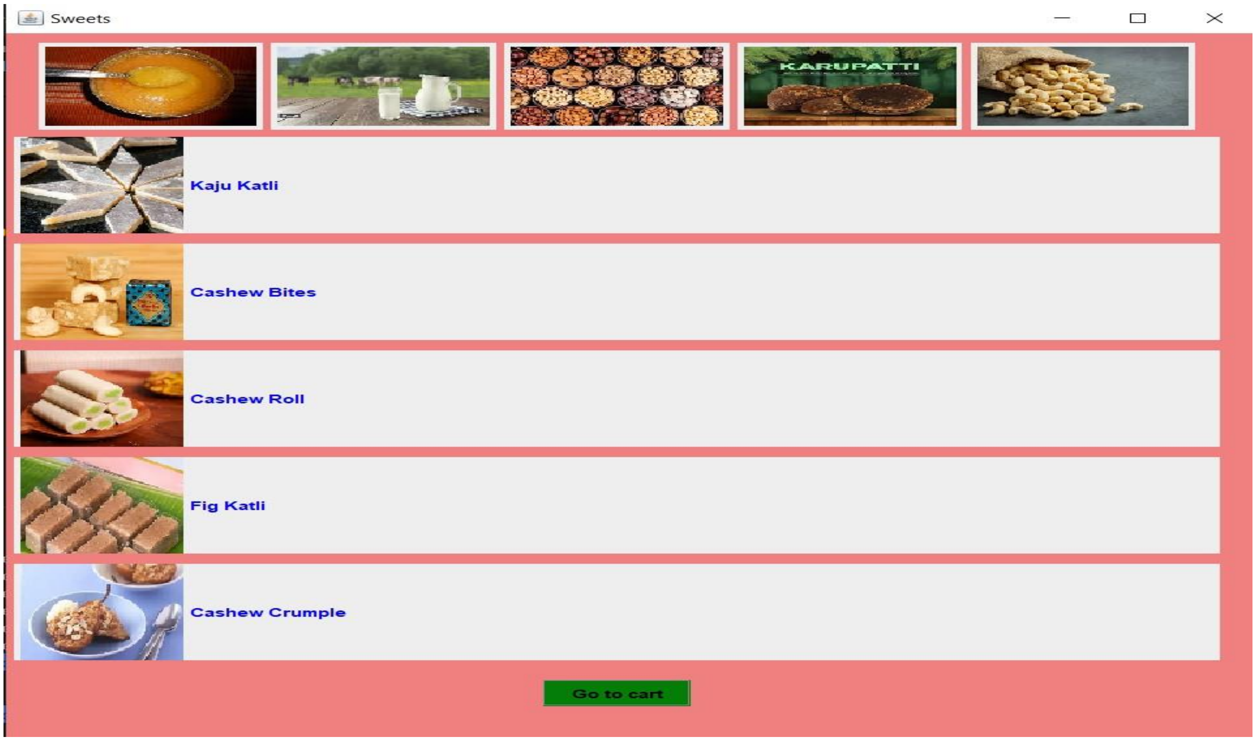
11. DRY FRUIT SWEET:



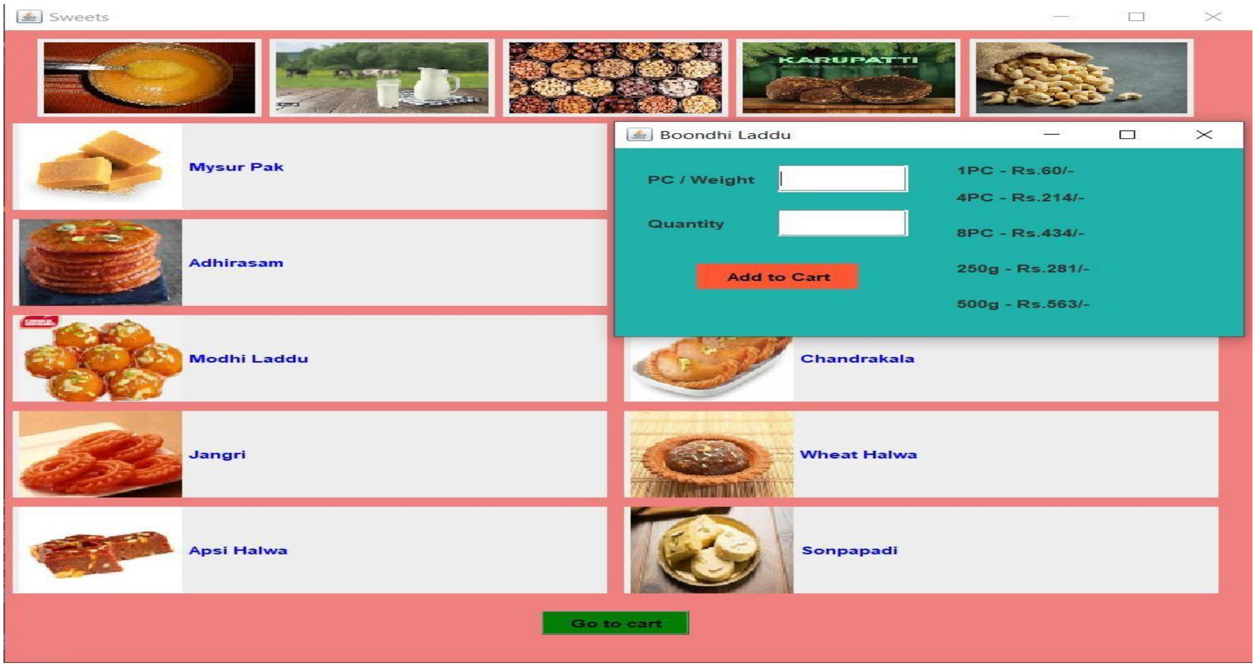
12. KARUPATTI SWEET:



13. CASHEW SWEET:



14. BOONDHI LADDU:

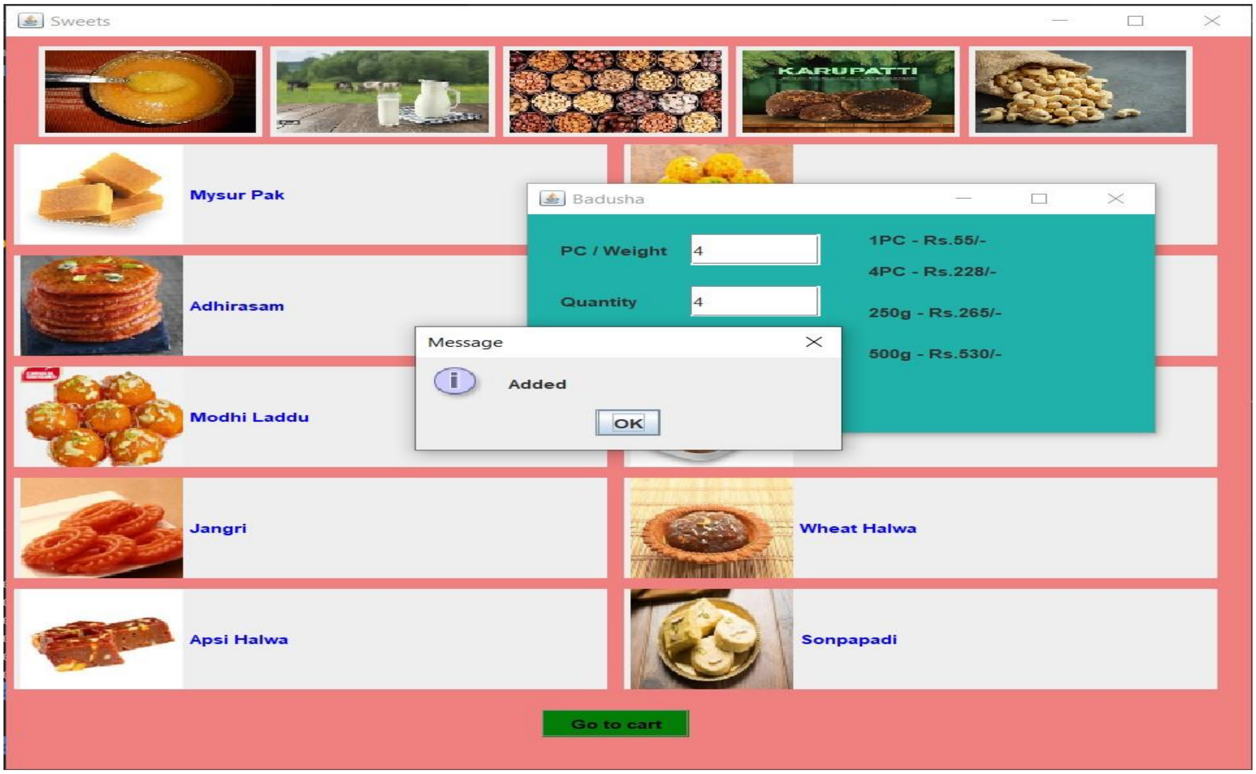


PC / Weight

Quantity

1PC - Rs.60/-
4PC - Rs.214/-
8PC - Rs.434/-
250g - Rs.281/-
500g - Rs.563/-

15. ADD TO CART:



PC / Weight 4

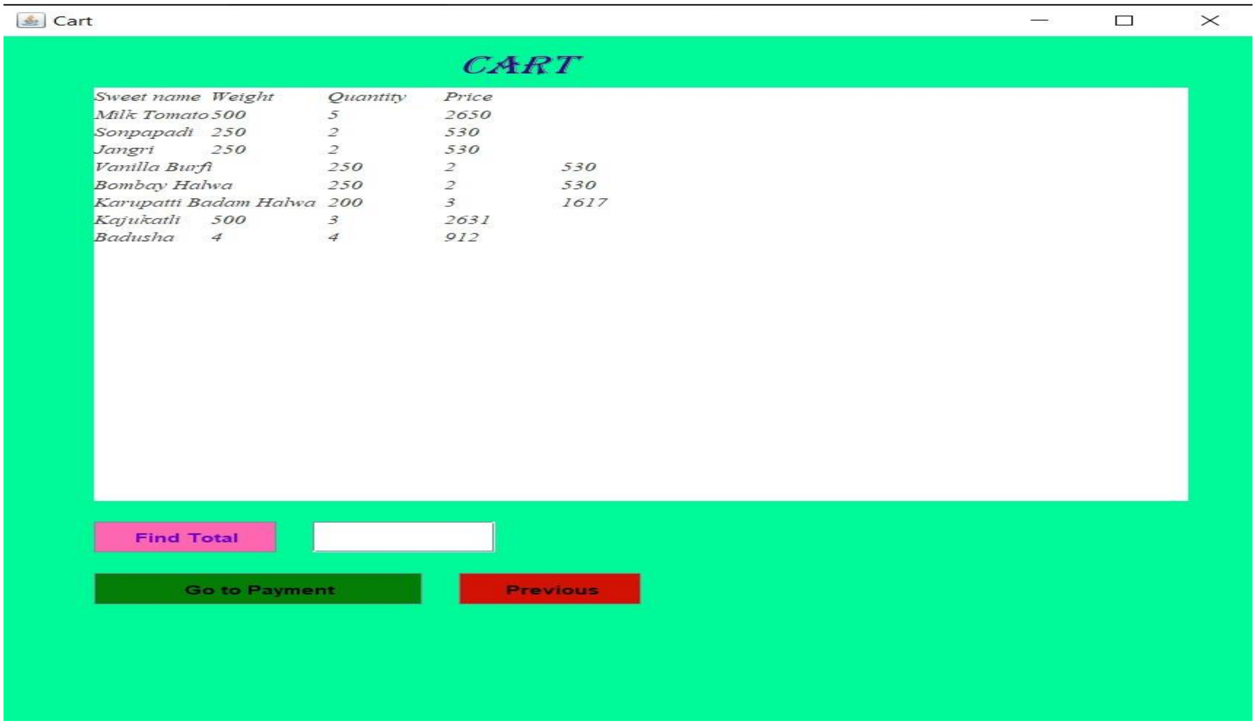
Quantity 4

1PC - Rs.55/-
4PC - Rs.228/-
250g - Rs.265/-
500g - Rs.530/-

Message

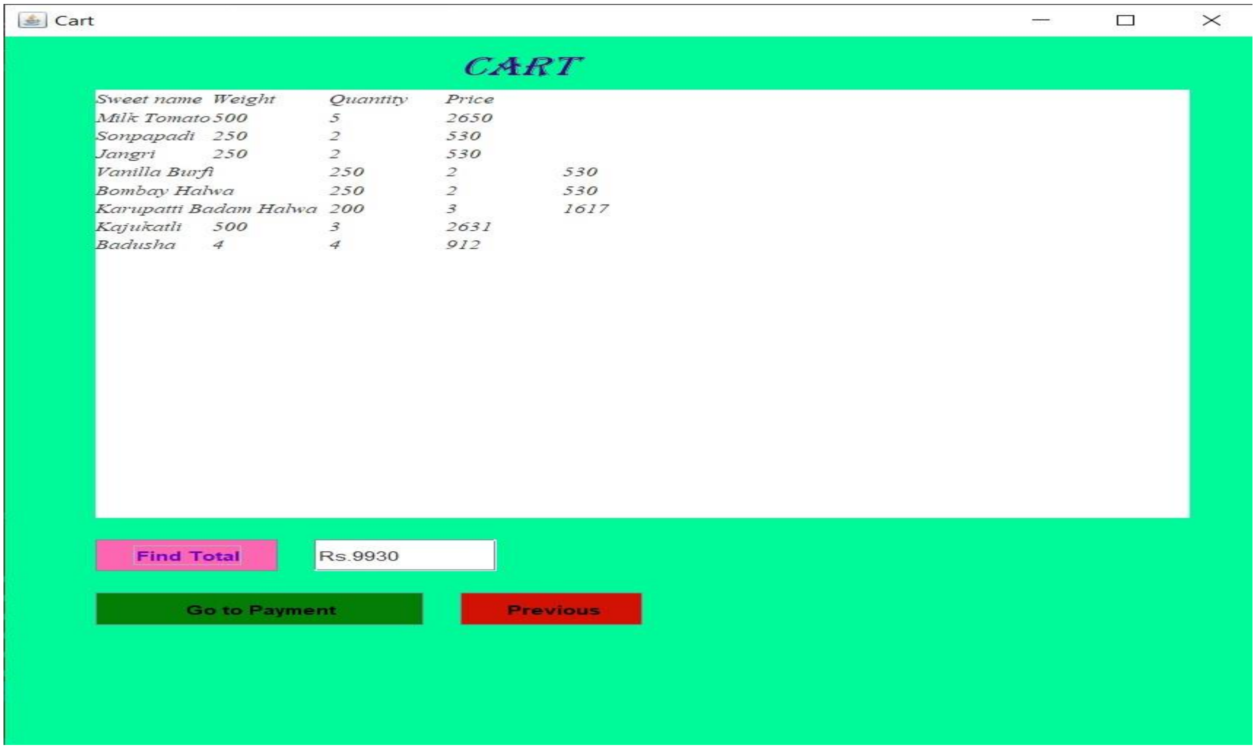
Added

16. CART:



```
Sweet name  Weight          Quantity       Price
Milk Tomato 500             5              2650
Sonpapadi    250            2              530
Jangri        250           2              530
Vanilla Burfi               250            2              530
Bombay Halwa                250            2              530
Karupatti Badam Halwa  200  3              1617
Kajukatli     500           3              2631
Badusha       4             4              912
```

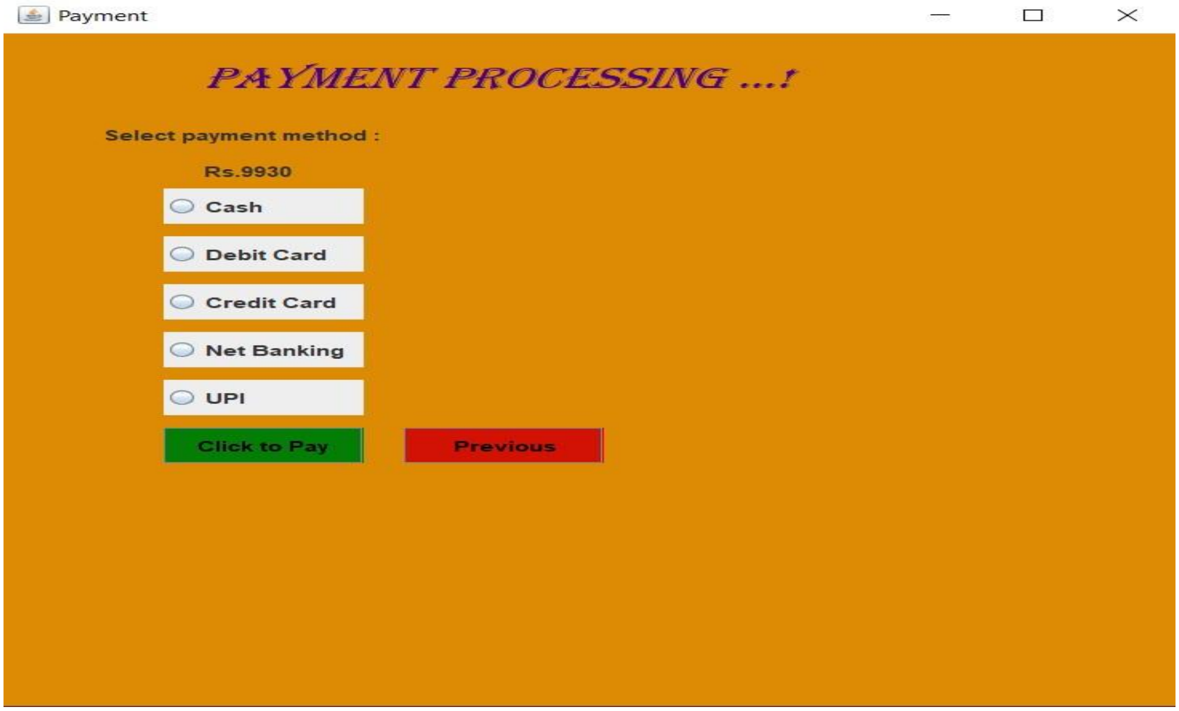
17. TOTAL:



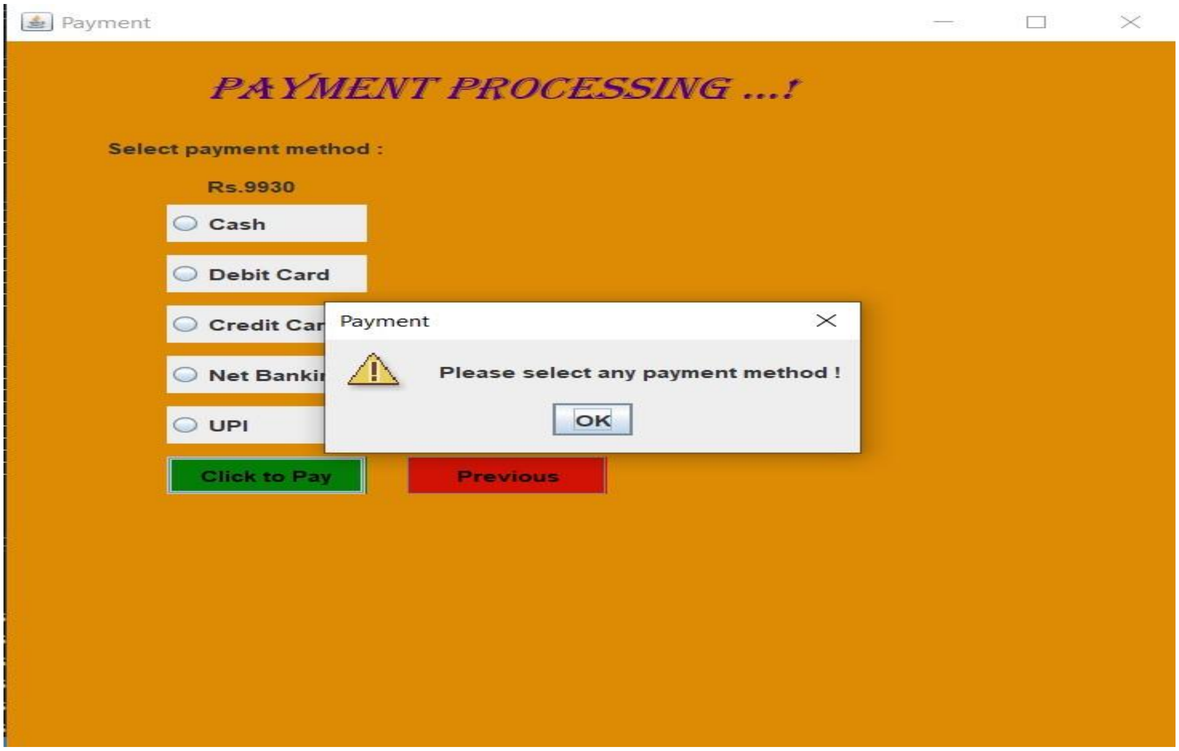
```
Sweet name  Weight          Quantity       Price
Milk Tomato 500             5              2650
Sonpapadi    250            2              530
Jangri        250           2              530
Vanilla Burfi               250            2              530
Bombay Halwa                250            2              530
Karupatti Badam Halwa  200  3              1617
Kajukatli     500           3              2631
Badusha       4             4              912
```

Rs.9930

18. PAYMENT:

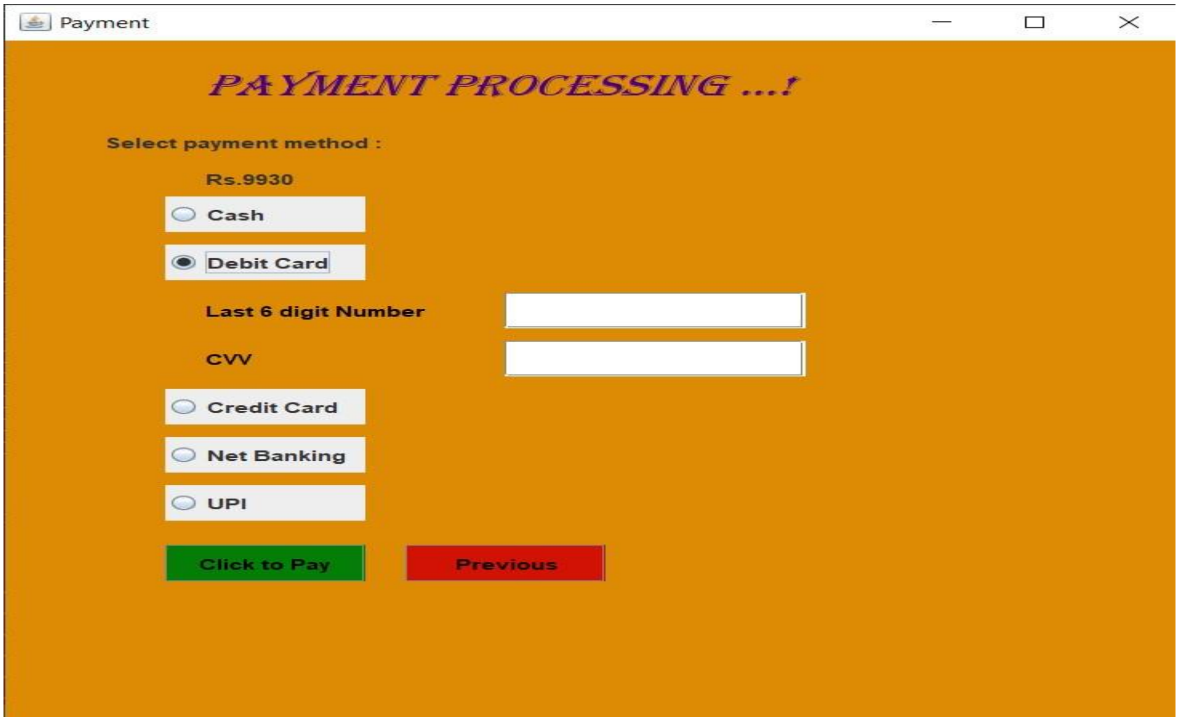


19. PAYMENT MESSAGE:



Please select any payment method !

20. CARD DETAILS:



Payment

PAYMENT PROCESSING ...!

Select payment method :

Rs.9930

○ Cash

● Debit Card

Last 6 digit Number

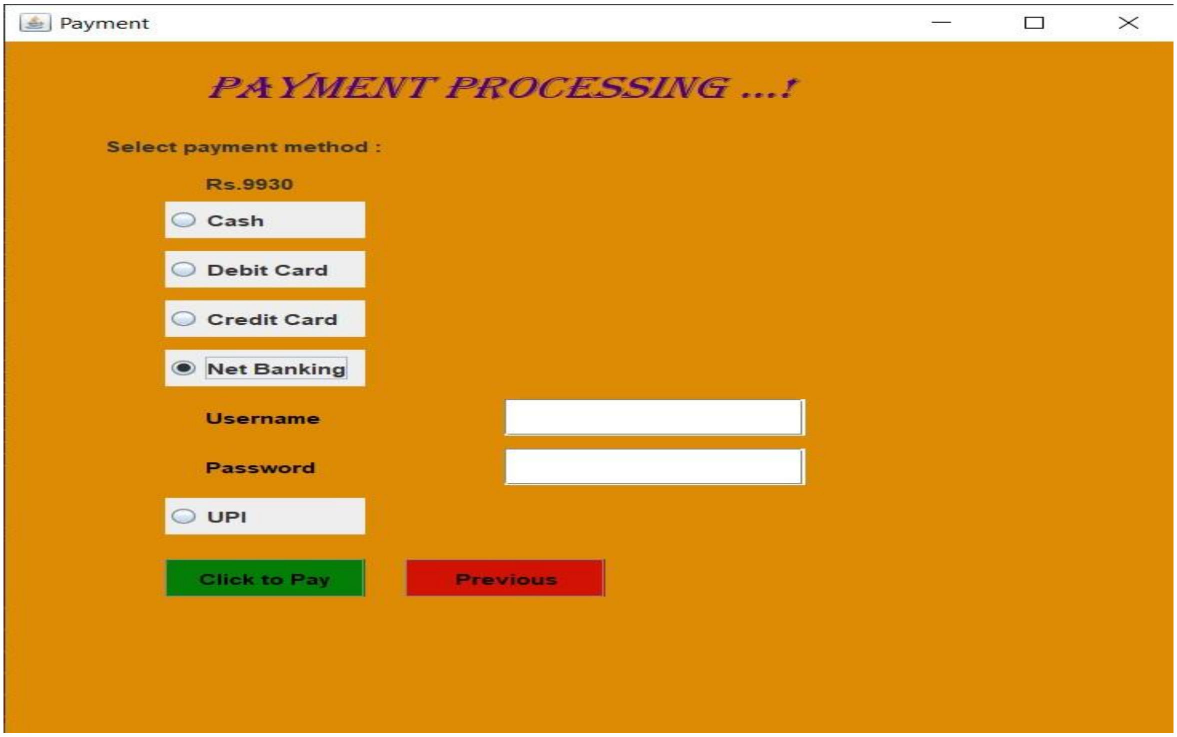
CVV

○ Credit Card

○ Net Banking

○ UPI

21. NET BANKING:



Payment

PAYMENT PROCESSING ...!

Select payment method :

Rs.9930

○ Cash

○ Debit Card

○ Credit Card

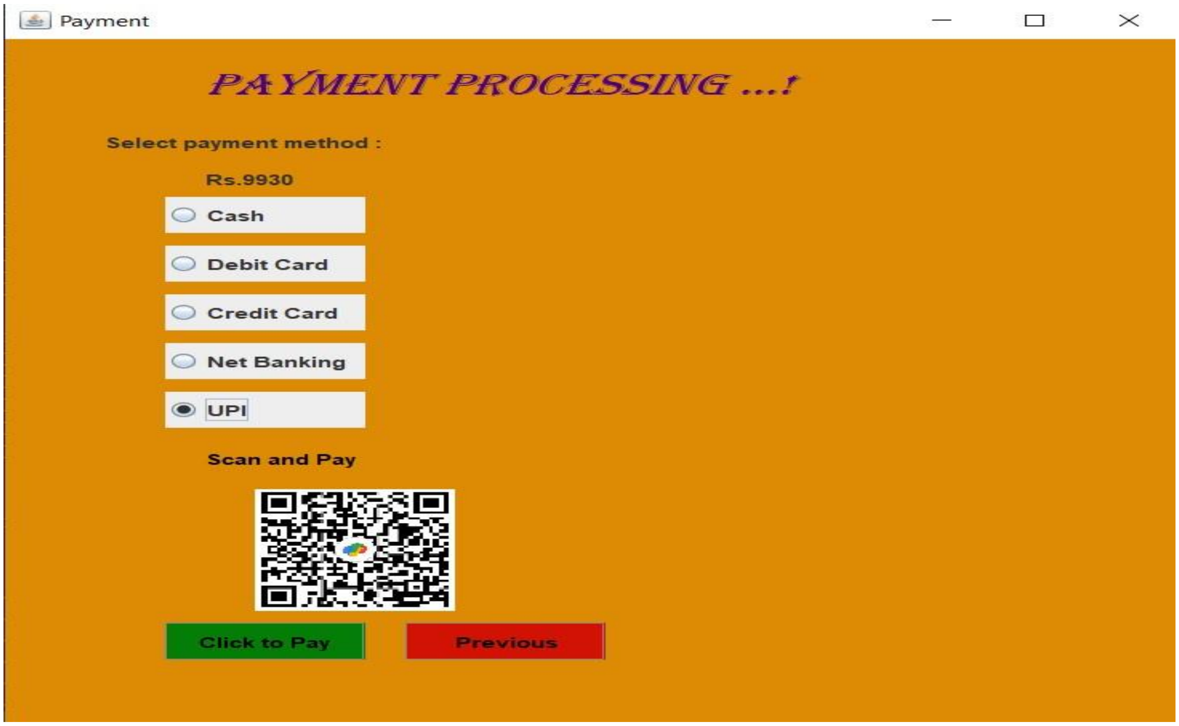
● Net Banking

Username

Password

○ UPI

22. UPI:



PAYMENT PROCESSING ...!

Select payment method :

Rs.9930

○ Cash

○ Debit Card

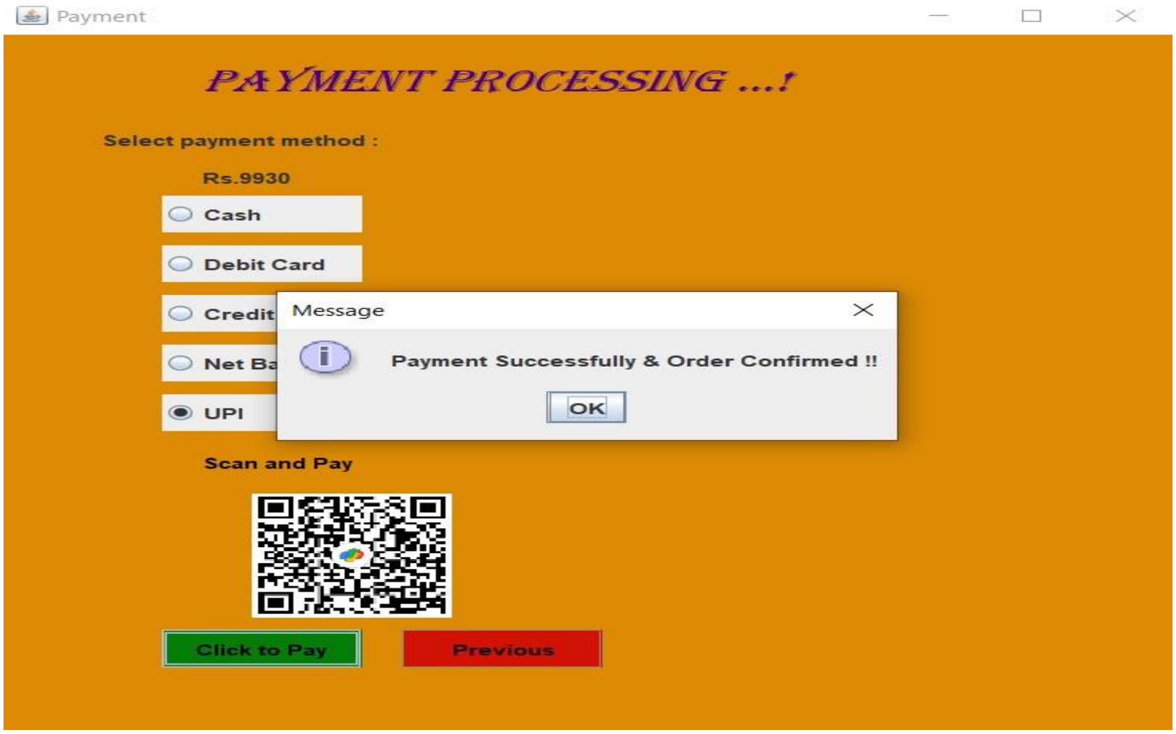
○ Credit Card

○ Net Banking

● UPI

Scan and Pay

23. PAYMENT SUCCESSFUL MESSAGE:



PAYMENT PROCESSING ...!

Select payment method :

Rs.9930

○ Cash

○ Debit Card

○ Credit

○ Net Ba

● UPI

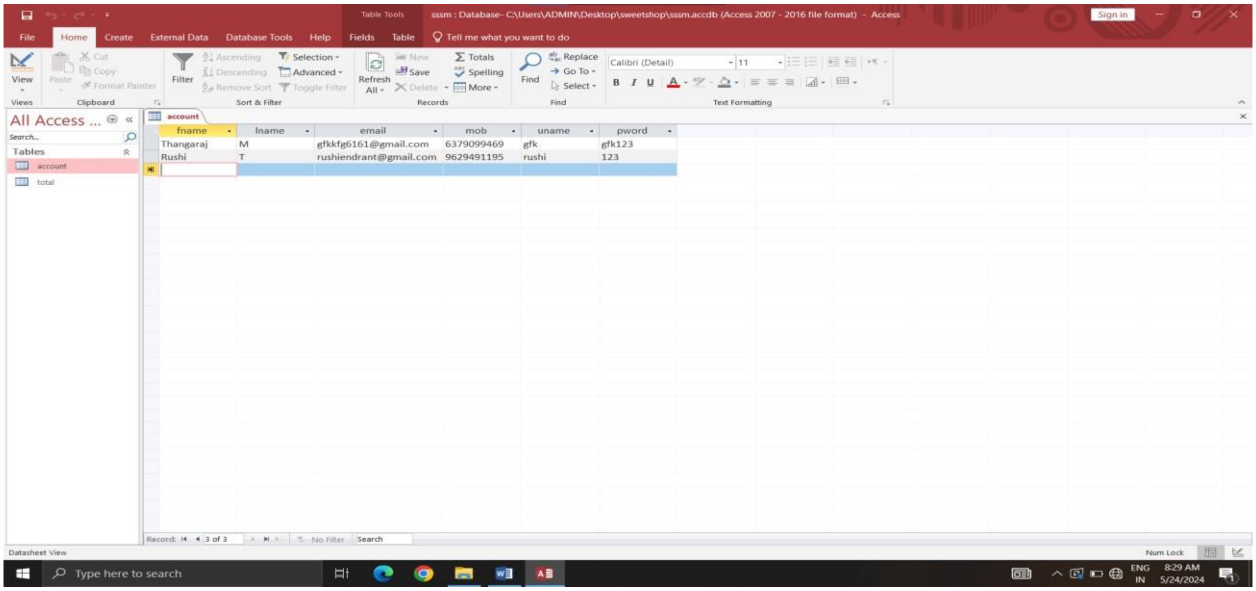
Scan and Pay

Message

Payment Successfully & Order Confirmed !!

MS ACCESS TABLE:

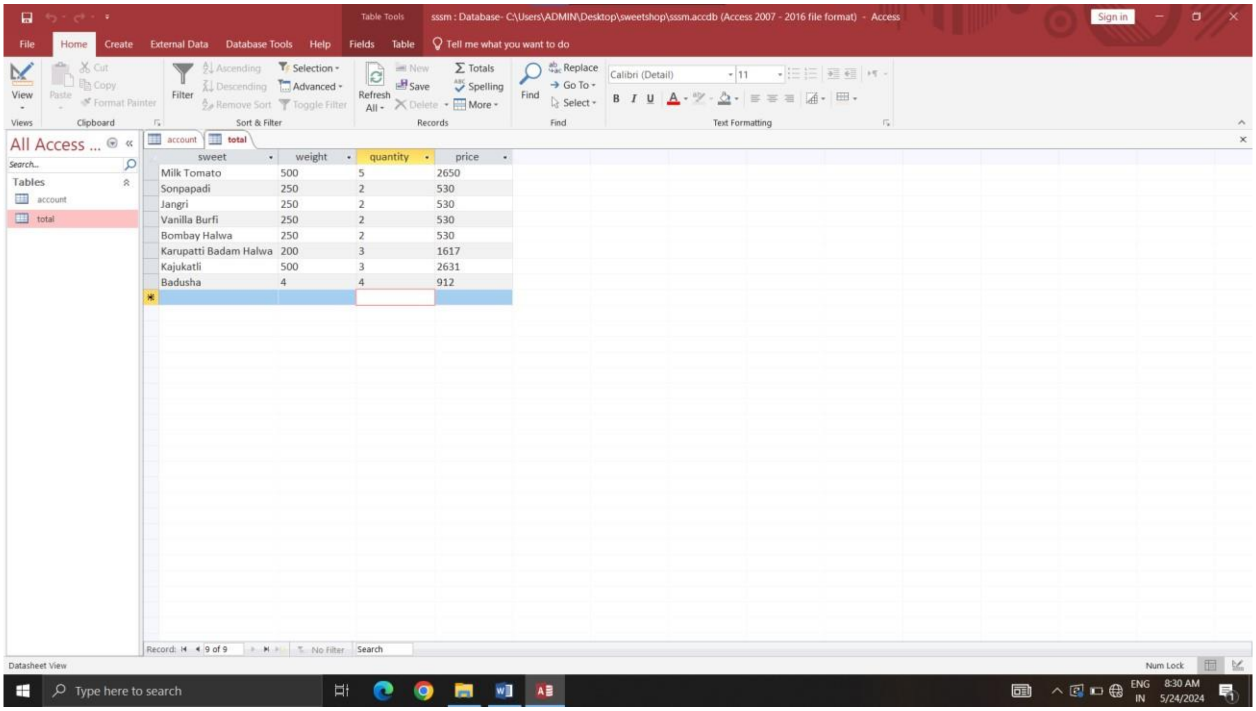
1. ACCOUNT:



account

fname	lname	email	mob	uname	pword
Thangaraj	M	gfkkfg6161@gmail.com	6379099469	gfk	gfk123
Rushi	T	rushiendrant@gmail.com	9629491195	rushi	123

2. TOTAL:



account | total

sweet	weight	quantity	price
Milk Tomato	500	5	2650
Sonpapadi	250	2	530
Jangri	250	2	530
Vanilla Burfi	250	2	530
Bombay Halwa	250	2	530
Karupatti Badam Halwa	200	3	1617
Kajukatli	500	3	2631
Badusha	4	4	912