

**ONLINE GROCERY STORE**

**JAVA SWING BASED- DATABASE MANAGEMENT  
SYSTEM - SQL CONNECTIVITY USING JDBC**

*A*

*Report*

*Submitted in partial fulfilment of the  
Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

By

**A.TILAKAVATI <1602-19-737-180>**



**Under the Guidance of**

**S. ARUNA**

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh,**

**Hyderabad-31**

**2021**

## **BONAFIDE CERTIFICATE**

This is to certify that the project report titled “**ONLINE GROCERY STORE**” project work of **Miss.A.Tilakavati** bearing **Roll.no:1602-19-737-180** who carried out this project under my supervision in the IV semester for the academic year 2019-2020.

Signature

External examiner

Signature

Internal examiner

# INDEX

<b>S.NO</b>	<b>CONTENT</b>	<b>PAGENO:</b>
<b>1</b>	<b>ABSTRACT</b>	<b>4</b>
<b>2</b>	<b>INTRODUCTION</b>	<b>5-7</b>
<b>3</b>	<b>ER DIAGRAM</b>	<b>8</b>
<b>4</b>	<b>CODE IMPLEMENTATION</b>	<b>10-97</b>
<b>5</b>	<b>TABLES</b>	<b>97-100</b>
<b>6</b>	<b>OUTPUT SCREENSHOTS</b>	<b>101-141</b>

## **ABSTRACT**

This is project “Online Grocery Store” is:It is a global decent.Online shopping has been known as a rapidly growing business, and although online grocery shopping has not followed these same growth patterns in the past, it is now being recognized for its potential. As such, the focus of previous online shopping research has seldom encompassed this specific retail market, with the existing studies focusing essentially on consumers’ motivations and attitudes, rather than how consumers actually shop for groceries online. Therefore, this dissertation has the objective of uncovering some of the details of consumer decision making processes for this specific online retail market, details which can help further both academic research and managerial knowledge.

# INTRODUCTION

## REQUIREMENTS FOR ONLINE GROCERY STORE.

List of Tables:-

1. Employee
2. Customer
3. Grlogin
4. Item
5. Vegetable
6. Water
7. Waterap
8. Milk
9. Milkap
10. Beverages
11. Beveragesap
12. Solid
13. Fruit
14. Seasonal Fruit
15. Cereal
16. Prevorder
17. Sales
18. Bill

Entity	Attribute	Domain
1. Employee	EID EFNAME ELNAME JOB_TITLE DOB DOJ SALARY ADDRESS EMAIL_ID EMPPHONE_NO	NUMBER(38) VARCHAR2(255) VARCHAR2(255) VARCHAR2(255) DATE DATE NUMBER(38) VARCHAR2(255) VARCHAR2(255) VARCHAR2(255)
2. Customer	CUSTOMER_ID CFNAME CLNAME EMAIL_ID GENDER AGE PHONE_NO ADDRESS	NUMBER(38) VARCHAR2(255) VARCHAR2(255) VARCHAR2(255) VARCHAR2(25) NUMBER(38) VARCHAR2(10) VARCHAR2(100)
3. Grlogin	USERNAME PASSWORD CUSTOMER_ID	VARCHAR2(15) VARCHAR2(15) NUMBER(38)
4. Item	ITEM_ID ITEM_NAME	NUMBER(38) VARCHAR2(255)
5. Vegetable	VID ITEM_ID	NUMBER(38) NUMBER(38)

	VEGETABLE_NAME AVAILABILITY PRICE	VARCHAR2(255) NUMBER(38) NUMBER(38)
6. Water	WID LIQUID_ID BRAND	NUMBER(38) NUMBER(38) VARCHAR2(255)
7. Waterap	WID AVAILABILITY PRICE QUANTITY	NUMBER(38) NUMBER(38) NUMBER(38) NUMBER(38)
8. Milk	MILK_ID LIQUID_ID BRAND	NUMBER(38) NUMBER(38) VARCHAR2(255)
9.Milkap	MILK_ID AVAILABILITY PRICE QUANTITY	NUMBER(38) NUMBER(38) NUMBER(38) NUMBER(38)
10.Beverages	BEVERAGES_ID LIQUID_ID BRAND	NUMBER(38) NUMBER(38) VARCHAR2(255)
11. Beveragesap	BEVERAGES_ID AVAILABILITY PRICE QUANTITY	NUMBER(38) NUMBER(38) NUMBER(38) NUMBER(38)
12. Solid	SOLID_ID ITEM_ID SNAME QUANTITY PRICE	NUMBER(38) NUMBER(38) VARCHAR2(255) NUMBER(38) NUMBER(38)
13. Fruit	FRUIT_ID ITEM_ID NAME AVAILABILITY PRICE	NUMBER(38) NUMBER(38) VARCHAR2(255) NUMBER(38) NUMBER(38)
14. Seasonal Fruit	SFRUIT_ID ITEM_ID NAME AVAILABILITY PRICE DOM	NUMBER(38) NUMBER(38) VARCHAR2(255) L NUMBER(38) NUMBER(38) DATE
15. Cereal	CEREAL_ID ITEM_ID NAME AVAILABILITY PRICE	NUMBER(38) NUMBER(38) VARCHAR2(255) NUMBER(38) NUMBER(38)
16. Prevorder	CUSTOMER_ID SNO NAME PRICE QUANTITY TOTAL	NUMBER(38) NUMBER(38) VARCHAR2(25) NUMBER(38) NUMBER(38) NUMBER(38)

17. Sales	SALES_ID CUSTOMER_ID DOP FINAL_PRICE TOP	NUMBER(38) NUMBER(38) DATE NUMBER(38) VARCHAR2(20)
18. Bill	ORDER_ID CUSTOMER_ID SALES_ID TOTAL_BILL DOP MOT AMOUNT_PAYABLE	NUMBER(38) NUMBER(38) NUMBER(38) NUMBER(38) DATE VARCHAR2(255) NUMBER(38)

## AIM OF THE PROJECT

The aim of this project is to manage the details of Shopping, Product, Internet, Payment, and Customer. It manages all the information about shopping, delivery, customer, shopping. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for the managing the shopping, product, delivery, internet. It tracks all the details about the internet, payment, and customer.

## ARCHITECTURE AND TECHNOLOGY USED

Java Eclipse, Oracle 11g Database, Java SE version 7, SQL\*Plus.

**SQL PLUS** is the most basic Oracle Database utility with a basic command-line interface, commonly used by users, administrators and programmers.

The interface of SQL Plus is used for creating the database. DDL and DML commands are implemented for operations being executed. The details of students, their logins, quiz, score are stored in the form of tables in the database.

**Eclipse** is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in java and its primary use is for developing java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Erlang, Javascripts etc.

The front end application code is written in “Java” using eclipse. The portal for front end application is designed through Eclipse, runs and has the capacity to connect with the database which has data inserted using SQL.

**Java Swing** is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

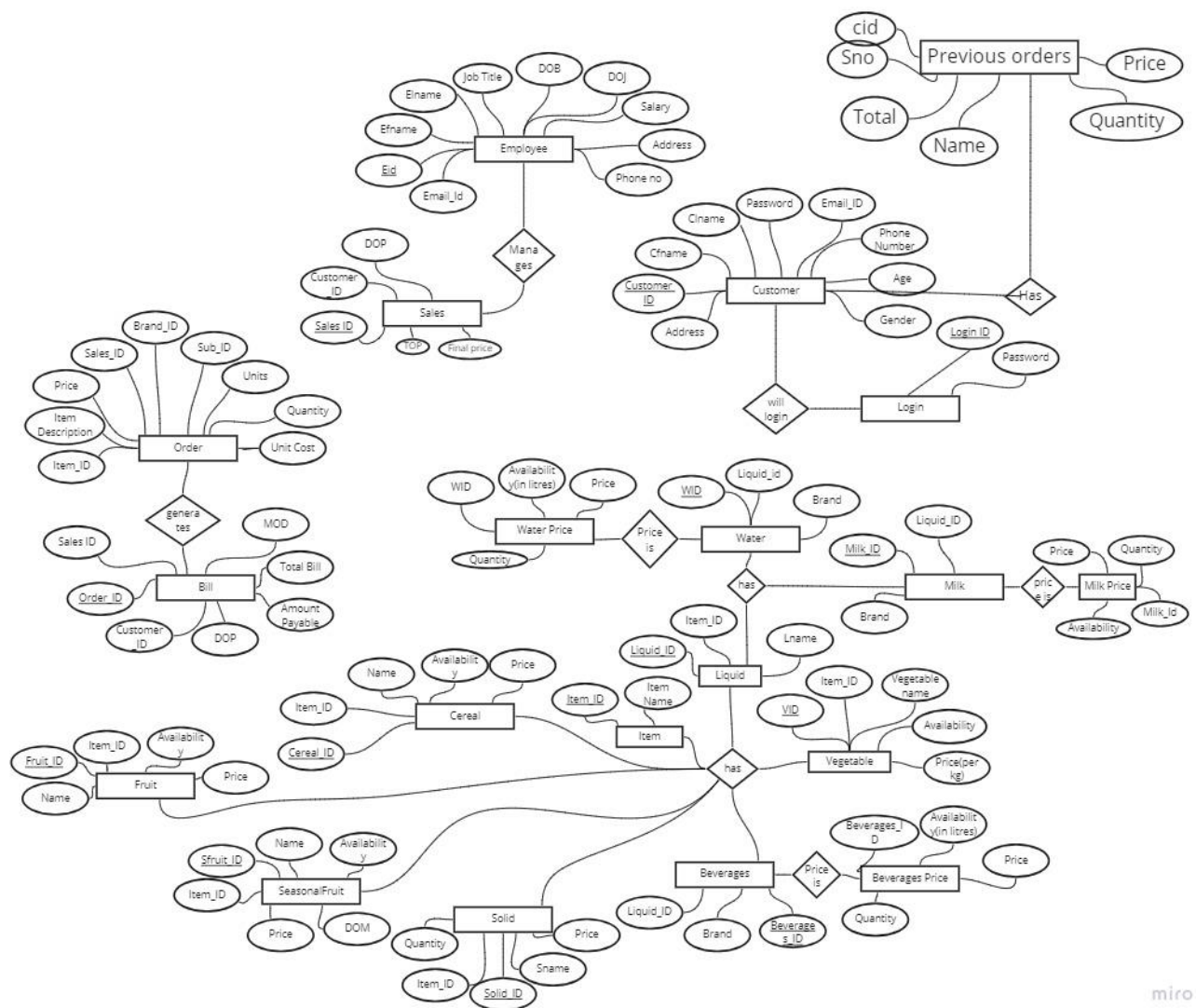
# ER DIAGRAM

## DEFINITION

The ER Diagram or ER Model stands for an Entity-Relationship model. It is a high-level data model which is used to define the data elements and relationship for a specified system. It develops a conceptual design for the database and a very simple and easy view of the data.

A basic ER Diagram consists of entities, attributes and relationships between the entities in the ER Diagram.

## ER Diagram of Online Grocery Store





## Java-SQL Connectivity using JDBC

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

### JDBC Connectivity

```
private void connToDb(){
try {
Class.forName("oracle.jdbc.driver.OracleDriver");
connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:xe","tilakavati","tilak
");
statement = connection.createStatement();
} catch (SQLException connectException) {
System.out.println(connectException.getMessage());
System.out.println(connectException.getSQLState());
System.out.println(connectException.getErrorCode());
System.exit(1);
} catch (
Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}}
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

# IMPLEMENTATION

## Admin.java

```
package dbmsp;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Font;
import javax.swing.JLabel;

public class admin extends JFrame {

    public static boolean admin=false;
    public static boolean user1=false;
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    admin frame = new admin();
                    frame.setTitle("Grocery Store");
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

public admin() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 530, 342);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("New label");
    lblNewLabel.setBounds(27, 11, 461, 281);
    contentPane.add(lblNewLabel);

    ImageIcon icon = new ImageIcon(this.getClass().getResource("/supermarket-management-system.jpg"));
    lblNewLabel.setIcon(icon);

    JButton btnNewButton = new JButton("Admin");
    btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 11));
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            admin=true;
            login.adminloginscreen();
            if(login.flag==true) {
                insadmin.adminscreen();
            }
        }
    });
    btnNewButton.setBounds(190, 52, 111, 37);
    contentPane.add(btnNewButton);

    JButton btnNewButton_1 = new JButton("User");
    btnNewButton_1.setFont(new Font("Tahoma", Font.BOLD, 11));
    btnNewButton_1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            user1=true;
            user.userscreen();
        }
    });
    btnNewButton_1.setBounds(190, 52, 111, 37);
    contentPane.add(btnNewButton_1);
}

```

```

        }

    });

    btnNewButton_1.setBounds(190, 117, 111, 37);

    contentPane.add(btnNewButton_1);

}
}

```

## Login.java

```

package dbmsp;

import java.awt.Component;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.awt.event.ActionEvent;
import java.awt.Color;
import javax.swing.UIManager;

public class login extends JFrame {

    public static boolean flag=false;

    private static final long serialVersionUID = 1L;

    private JPanel contentPane;

```

```

private JTextField textField;

private JPasswordField passwordField;

protected Component frame;

String a,b;

public static int cid;

public static int di;

static login frame1;

static String name="";

public static void adminloginscreen() {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                login frame = new login();
                frame.setTitle("Grocery Store");
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public login() {
    dispose();
    setBounds(100, 100, 601, 353);
    contentPane = new JPanel();
    contentPane.setBackground(UIManager.getColor("InternalFrame.resizeIconHighlight"));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    if(usersc.s==1) {
        dispose();
    }
}

```

```

JLabel lblNewLabel = new JLabel("Username");
lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 20));
lblNewLabel.setBounds(77, 59, 102, 34);
contentPane.add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("Password");
lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 20));
lblNewLabel_1.setBounds(77, 146, 102, 34);
contentPane.add(lblNewLabel_1);

textField = new JTextField();
textField.setBounds(263, 63, 161, 34);
contentPane.add(textField);
textField.setColumns(10);

passwordField = new JPasswordField();
passwordField.setBounds(263, 146, 161, 34);
contentPane.add(passwordField);

JButton btnNewButton = new JButton("Login");
btnNewButton.setBackground(UIManager.getColor("Button.disabledShadow"));
btnNewButton.setForeground(new Color(0, 0, 0));
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String s=textField.getText();
        char[] p=passwordField.getPassword();
        String pass=String.valueOf(p);
        if(admin.admin==true) {
            if(pass.equals("") && s.trim().equals("")) {
JOptionPane.showMessageDialog(frame,"Please fill out the username and password fields");
            }
            else if(pass.equals("") && s.length()>0) {
JOptionPane.showMessageDialog(frame, "Please fill out the password field");
            }
            else if(pass.length()>0 && s.length()==0) {
JOptionPane.showMessageDialog(frame, "Please fill out the user field");
            }
        }
    }
});

```

```

elseif(s.trim().equals("admin") && pass.trim().equals("admin@123")) {
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    dispose();
    flag=true;
    insadmin.adminscreen();
    admin.admin=false;
    flag=false;
}
else {
OptionPane.showMessageDialog(frame,"incorrect username or password");
}
}
else {
try {
    String query="select * from grlogin where username=?";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
    PreparedStatement ps=con.prepareStatement(query);
    ps.setString(1, textField.getText());

    ResultSet rs=ps.executeQuery();
    rs.next();
    a=rs.getString(1);
    b=rs.getString(2);
    cid=rs.getInt(3);
    name=a;
}
catch(ClassNotFoundException|SQLException e1) {
    e1.printStackTrace();
}
if(pass.equals("") && s.trim().equals("")) {
OptionPane.showMessageDialog(frame,"Please fill out the username and password fields");
}
}
}

```

```

        }
        else if(pass.equals("") && s.length()>0) {
JOptionPane.showMessageDialog(frame, "Please fill out the password field");
        }
        else if(pass.length()>0 && s.length()==0) {
JOptionPane.showMessageDialog(frame, "Please fill out the user field");
        }
        else if(s.equals(a) && pass.equals(b)) {
JOptionPane.showMessageDialog(frame,"Hurray! You have made it");
            admin.user1=false;

            usersc.screen();

            dispose();

            setVisible(false);

            di=cid;

        }
        else {
JOptionPane.showMessageDialog(frame,"incorrect username or password");
        }
    }
});

btnNewButton.setBounds(107, 235, 134, 40);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("Clear");
btnNewButton_1.setBackground(UIManager.getColor("Button.disabledShadow"));
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText("");
        passwordField.setText("");
    }
});

btnNewButton_1.setBounds(337, 235, 134, 40);
contentPane.add(btnNewButton_1);

```



```
    }  
}
```

## **Logsc.java**

```
package dbmsp;  
  
import java.awt.Component;  
import java.awt.EventQueue;  
import java.awt.Font;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JPasswordField;  
import javax.swing.JTextField;  
import javax.swing.border.EmptyBorder;  
  
public class logsc extends JFrame {  
    private static final long serialVersionUID = 1L;  
    //reg rg=new reg();  
    private JPanel contentPane;  
    private JTextField textField;  
    private JPasswordField passwordField;  
    private JPasswordField passwordField_1;  
    public static int logintext;  
    Component frame=null;  
    public static boolean suc=false;  
    public static void screen() {
```

```

EventQueue.invokeLater(new Runnable() {
    public void run() {
        try {
            logsc frame = new logsc();
            frame.setTitle("Password Confirmation");
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
}

public logsc() {
    setBounds(100, 100, 599, 384);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("UserName");
    lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 14));
    lblNewLabel.setBounds(65, 60, 108, 36);
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Password");
    lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 14));
    lblNewLabel_1.setBounds(65, 127, 108, 36);
    contentPane.add(lblNewLabel_1);

    textField = new JTextField();
    textField.setBounds(265, 60, 145, 36);
    contentPane.add(textField);
    textField.setColumns(10);

    JLabel lblNewLabel_2 = new JLabel("Confirm Password");

```

```

lblNewLabel_2.setFont(new Font("Tahoma", Font.PLAIN, 14));
lblNewLabel_2.setBounds(65, 199, 110, 36);
contentPane.add(lblNewLabel_2);
JButton btnNewButton = new JButton("Save");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            String query1="select * from customer where cfname=?";
            Class.forName("oracle.jdbc.driver.OracleDriver");

```

Connection

```

con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

PreparedStatement ps1=con1.prepareStatement(query1);
ps1.setString(1, textField.getText());
ps1.executeUpdate();

String p1=String.valueOf(passwordField.getPassword());
String p2=String.valueOf(passwordField_1.getPassword());
if(p1.length()<8) {
    Component frame = null;
    JOptionPane.showMessageDialog(frame, "passwords must be 8 character long");
}
else if(p1.compareTo(p2)!=0 || p1.length()<8) {
    Component frame = null;
    JOptionPane.showMessageDialog(frame, "passwords must match");
}
else if(p1.compareTo(p2)==0 && p1.length()>=8){
    String query="insert into grlogin values(?,?,?)";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

    PreparedStatement ps=con.prepareStatement(query);
    ps.setString(1,textField.getText());

```

```

ps.setString(2,String.valueOf(passwordField.getPassword()));

```

```

        ps.setInt(3, register.id);
        suc=true;
        ps.executeUpdate();
        dispose();
        setVisible(false);
        ps.close();
        con.close();
    }
}

catch (ClassNotFoundException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
} catch (SQLException e1) {
    // TODO Auto-generated catch block
    //e1.printStackTrace();
JOptionPane.showMessageDialog(null,"Username      already      exists","Error      message"
,JOptionPane.ERROR_MESSAGE);
    }}

});

btnNewButton.setBounds(84, 274, 126, 36);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("Clear");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        passwordField.setText("");
        passwordField_1.setText("")
    };
});

btnNewButton_1.setBounds(251, 274, 116, 36);
contentPane.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("Back");
btnNewButton_2.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            dispose();
            setVisible(false);}
    });
    btnNewButton_2.setBounds(407, 274, 116, 36);
    contentPane.add(btnNewButton_2);
    passwordField = new JPasswordField();
    passwordField.setBounds(265, 129, 145, 36);
    contentPane.add(passwordField);
    passwordField_1 = new JPasswordField();
    passwordField_1.setBounds(265, 201, 145, 36);
    contentPane.add(passwordField_1);
    }}

```

## **Additems.java**

```

package dbmsp;
import java.awt.Component;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JComboBox;
import java.awt.Font;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```

import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.ActionEvent;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;
public class additems extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    String cat[]= {"Select","Vegetables","Water","Milk","Beverages","Solid","Fruits","Seasonal
Fruits","Cereals"};
    public static int n,b,c;
    DefaultTableModel model;
    Component frame=null;
    private JTextField textField_3;
    public static void screen() {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    additems frame = new additems();
                    frame.setTitle("Items Info");
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public additems() {
        dispose();
        setBounds(100, 100, 575, 409);
    }
}

```

```

contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
model=new DefaultTableModel();
textField = new JTextField();
textField.setBounds(158, 45, 86, 20);
contentPane.add(textField);
textField.setColumns(10);
JLabel lblNewLabel = new JLabel("Name");
lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 14));
lblNewLabel.setBounds(30, 43, 86, 20);
contentPane.add(lblNewLabel);
textField_1 = new JTextField();
textField_1.setBounds(158, 91, 86, 20);
contentPane.add(textField_1);
textField_1.setColumns(10);
JLabel lblNewLabel_1 = new JLabel("Quantity");
lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 14));
lblNewLabel_1.setBounds(30, 92, 86, 14);
contentPane.add(lblNewLabel_1);
textField_2 = new JTextField();
textField_2.setBounds(158, 145, 86, 20);
contentPane.add(textField_2);
textField_2.setColumns(10);
JLabel lblNewLabel_2 = new JLabel("Price");
lblNewLabel_2.setFont(new Font("Tahoma", Font.PLAIN, 14));
lblNewLabel_2.setBounds(30, 146, 46, 14);
contentPane.add(lblNewLabel_2);
@SuppressWarnings({ "rawtypes", "unchecked" })
JComboBox comboBox = new JComboBox(cat);
comboBox.addItemListener(new ItemListener() {

```

```

        public void itemStateChanged(ItemEvent e) {
            String d=(String) comboBox.getSelectedItem();
            if(d.equals("Water")||d.equals("Beverages")||d.equals("Milk")){
                textField_3.setEditable(true);
            }
            else {
                textField_3.setEditable(false);
            }
        }
    });

    comboBox.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String s=comboBox.getSelectedItem().toString();
            if(s.equals("Beverages")||s.equals("Milk")||s.equals("Water")) {
                textField_3.setEditable(true);
            }
        }
    });

    comboBox.setBounds(158, 256, 114, 22);
    contentPane.add(comboBox);

    comboBox.getSelectedItem().toString();

    JLabel lblNewLabel_3 = new JLabel("Select Category");
    lblNewLabel_3.setFont(new Font("Tahoma", Font.PLAIN, 14));
    lblNewLabel_3.setBounds(30, 255, 125, 21);
    contentPane.add(lblNewLabel_3);

    if(comboBox.getSelectedItem().toString().equals("Beverages")) {
        textField_3.setEditable(true);
    }

    JButton btnNewButton = new JButton("Save");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(textField.getText().length()<2||textField_1.getText().length()<1||textField_2.getText().length()<1)

```



```

{

    JOptionPane.showMessageDialog(frame, "Please fill the boxes");

}

else if(comboBox.getSelectedIndex()==0) {

    JOptionPane.showMessageDialog(frame, "Please select the cateogry");

}

else {

int a=JOptionPane.showConfirmDialog(frame,"Are you sure you want to save");

if(a==JOptionPane.YES_OPTION) {

    JOptionPane.showMessageDialog(frame,"Saved");

    try {

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

        String d=comboBox.getSelectedItem().toString();

        if(d.equals("Vegetables")) {

            textField_3.setEditable(false);

String query1="insert into vegetable values(seq_veg.nextval,?,?,?,?)";

            PreparedStatement ps1=con.prepareStatement(query1);

            ps1.setInt(1, 1);

            ps1.setString(2, textField.getText());

            ps1.setInt(3,n);

            ps1.setInt(4, b);

            ps1.executeUpdate();

        }

        if(d.equals("Water")) {

            try {

                n=Integer.parseInt(textField_1.getText());

                b=Integer.parseInt(textField_2.getText());

                c=Integer.parseInt(textField_3.getText());

            }

            catch(NumberFormatException e1) {

JOptionPane.showMessageDialog(frame, "Quantity and price should be numbers");

```

```

    }
    String query5="select * from water where brand=?";
    PreparedStatement ps5=con.prepareStatement(query5);
    ps5.setString(1, textField.getText());
    int u=ps5.executeUpdate();
    ResultSet rs=ps5.executeQuery();
    if(u==0) {
String query2="insert into water values(seq_wat.nextval,?,?)";
    PreparedStatement ps2=con.prepareStatement(query2);
        ps2.setInt(1,2);
        ps2.setString(2,textField.getText());
        ps2.executeUpdate();
        String query4="select max(wid) from water";
        Statement ps3=con.createStatement();
        ResultSet rs1=ps3.executeQuery(query4);
        rs1.next();
        int y=rs1.getInt(1);
        String query3="insert into waterap values(?,?,?,?)";
        PreparedStatement ps4=con.prepareStatement(query3);
            ps4.setInt(1, y);
            ps4.setInt(2, c);
            ps4.setInt(3, b);
            ps4.setInt(4,n);
            ps4.executeUpdate();
        }
        if(u>=1) {
            rs.next();
            int wid=rs.getInt(1);

String query3="insert into waterap values(?,?,?,?)";
        PreparedStatement ps4=con.prepareStatement(query3);
            ps4.setInt(1, wid);
            ps4.setInt(2, c);

```

```

        ps4.setInt(3, b);
        ps4.setInt(4,n);
        ps4.execute();
    }
}

if(d.equals("Milk")) {
    try {
        n=Integer.parseInt(textField_1.getText());
        b=Integer.parseInt(textField_2.getText());
        c=Integer.parseInt(textField_3.getText());

    }
    catch(NumberFormatException e1) {

OptionPane.showMessageDialog(frame, "Quantity and price should be numbers");

    }

    String query5="select * from milk where brand=?";
    PreparedStatement ps5=con.prepareStatement(query5);
    ps5.setString(1, textField.getText());
    int u=ps5.executeUpdate();
    ResultSet rs=ps5.executeQuery();
    if(u==0) {
String query2="insert into milk values(seq_milk.nextval,?,?)";
PreparedStatement ps2=con.prepareStatement(query2);
        ps2.setInt(1,7);
        ps2.setString(2,textField.getText());
        ps2.executeUpdate();
        ps2.close();
        String query4="select max(milk_id) from milk";
        Statement ps3=con.createStatement();
        ResultSet rs1=ps3.executeQuery(query4);
        rs1.next();
        int y=rs1.getInt(1);

```

```

        String query3="insert into milkap values(?,?,?,?)";
        PreparedStatement ps4=con.prepareStatement(query3);

        ps4.setInt(1, y);
        ps4.setInt(2, c);
        ps4.setInt(3, b);
        ps4.setInt(4,n);
        ps4.execute();
    }
    else {
        rs.next();
        int milkid=rs.getInt(1);

        String query3="insert into milkap values(?,?,?,?)";
        PreparedStatement ps4=con.prepareStatement(query3);

        ps4.setInt(1, milkid);
        ps4.setInt(2, c);
        ps4.setInt(3, b);
        ps4.setInt(4,n);
        ps4.execute();

    }
}

if(d.equals("Beverages")) {
    try {
        n=Integer.parseInt(textField_1.getText());
        b=Integer.parseInt(textField_2.getText());
        c=Integer.parseInt(textField_3.getText());

    }
    catch(NumberFormatException e1) {

        JOptionPane.showMessageDialog(frame, "Quantity and price should be numbers");
    }

    String query5="select * from beverages where brand=?";

```

```

PreparedStatement ps5=con.prepareStatement(query5);

        ps5.setString(1, textField.getText());
        int u=ps5.executeUpdate();
        ResultSet rs=ps5.executeQuery();
        if(u==0) {
String query2="insert into beverages values(seq_bevr.nextval,?,?)";
        PreparedStatement ps2=con.prepareStatement(query2);
                ps2.setInt(1,8);
                ps2.setString(2,textField.getText());
                ps2.executeUpdate();

                ps2.close();
String query4="select max(beverages_id) from beverages";
                Statement ps3=con.createStatement();
                ResultSet rs1=ps3.executeQuery(query4);
                rs1.next();
                int y=rs1.getInt(1);
String query3="insert into beveragesap values(?,?,?,?)";
        PreparedStatement ps4=con.prepareStatement(query3);
                ps4.setInt(1, y);
                ps4.setInt(2, c);
                ps4.setInt(3, b);
                ps4.setInt(4,n);
                ps4.execute();
                ps2.close();
                ps3.close();
                ps4.close();
                ps5.close();
                }else {
                        rs.next();
                        int bevid=rs.getInt(1);
String query3="insert into beveragesap values(?,?,?,?)";
        PreparedStatement ps4=con.prepareStatement(query3);

```

```

        ps4.setInt(1, bevid);
        ps4.setInt(2, c);
        ps4.setInt(3, b);
        ps4.setInt(4,n);
        ps4.execute();
        ps4.close();

    }
}

if(d.equals("Solid")) {
    textField_3.setEditable(false);
    try {
        n=Integer.parseInt(textField_1.getText());
        b=Integer.parseInt(textField_2.getText());

    }

    catch(NumberFormatException e1) {

JOptionPane.showMessageDialog(frame, "Quantity and price should be numbers");

    }

String query1="insert into solid values(seq_solid.nextval,?,?,?,?)";
PreparedStatement ps1=con.prepareStatement(query1);
        ps1.setInt(1, 3);
        ps1.setString(2, textField.getText());
        ps1.setInt(3,n);
        ps1.setInt(4, b);
        ps1.executeUpdate();
    }

if(d.equals("Fruits")) {
    textField_3.setEditable(false);
    try {
        n=Integer.parseInt(textField_1.getText());
        b=Integer.parseInt(textField_2.getText());

```

```

        }
        catch(NumberFormatException e1) {

JOptionPane.showMessageDialog(frame, "Quantity and price should be numbers");

        }

String query1="insert into fruit values(seq_fruit.nextval,?,?,?,?)";
PreparedStatement ps1=con.prepareStatement(query1);

        ps1.setInt(1, 4);
        ps1.setString(2, textField.getText());
        ps1.setInt(3,n);
        ps1.setInt(4, b);
        ps1.executeUpdate();

    }

    if(d.equals("Seasonal Fruits")) {
        textField_3.setEditable(false);
        try {
            n=Integer.parseInt(textField_1.getText());
            b=Integer.parseInt(textField_2.getText());
        }
        catch(NumberFormatException e1) {

JOptionPane.showMessageDialog(frame, "Quantity and price should be numbers");

        }

        java.util.Date date1 = new java.util.Date();
        long t = date1.getTime();

        java.sql.Date sqlDate = new java.sql.Date(t);
String query1="insert into seasonalfruit values(seq_sfruit.nextval,?,?,?,?,?)";
        PreparedStatement ps1=con.prepareStatement(query1);

        ps1.setInt(1, 5);
        ps1.setString(2, textField.getText());
        ps1.setInt(3,n);
        ps1.setInt(4, b);
        ps1.setDate(5, sqlDate);

```

```

        ps1.executeUpdate();
    }
    if(d.equals("Cereals")) {
        textField_3.setEditable(false);
        try {
            n=Integer.parseInt(textField_1.getText());
            b=Integer.parseInt(textField_2.getText());

        }
        catch(NumberFormatException e1) {

JOptionPane.showMessageDialog(frame, "Quantity and price should be numbers");

        }

String query1="insert into cereal values(seq_cereal.nextval,?,?,?,?)";
        PreparedStatement ps1=con.prepareStatement(query1);

        ps1.setInt(1, 6);
        ps1.setString(2, textField.getText());
        ps1.setInt(3,n);
        ps1.setInt(4, b);
        ps1.executeUpdate();

    }

    }

    catch(ClassNotFoundException|SQLException e1 ) {

        e1.printStackTrace();

    }

    textField_3.setEditable(false);
    textField_3.setText("");

}

}

});

btnNewButton.setBounds(354, 44, 89, 23);

```



```
contentPane.add(btnNewButton);
```

```
JButton btnNewButton_1 = new JButton("Clear");
```

```
btnNewButton_1.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        textField.setText("");
```

```
        textField_1.setText("");
```

```
        textField_2.setText("");
```

```
        textField_3.setText("");
```

```
        textField_3.setEditable(false);
```

```
        comboBox.setSelectedIndex(0);
```

```
    }
```

```
});
```

```
btnNewButton_1.setBounds(354, 103, 89, 23);
```

```
contentPane.add(btnNewButton_1);
```

```
JButton btnNewButton_2 = new JButton("Delete");
```

```
btnNewButton_2.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        textField_1.setText("");
```

```
        textField_2.setText("");
```

```
        textField_3.setText("");
```

```
        textField_1.setEditable(false);
```

```
        textField_2.setEditable(false);
```

```
        textField_3.setEditable(false);
```

```
if(textField.getText().length()>2 && comboBox.getSelectedItem()!="Select") {
```

```
    int a=JOptionPane.showConfirmDialog(frame,"Are you sure you want to delete");
```

```
    if(a==JOptionPane.YES_OPTION) {
```

```
        try {
```

```

String a1=textField.getText();

String f=comboBox.getSelectedItemAt().toString();

if(f.equals("Vegetables")) {

String query="delete from vegetable where vegetable_name=?";

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

PreparedStatement ps=con.prepareStatement(query);

ps.setString(1,a1);

int y=ps.executeUpdate();

if(y==0) {
JOptionPane.showMessageDialog(frame, "Item name doesnt exist");
}

else {

JOptionPane.showMessageDialog(frame,"Deleted");

}

System.out.println(y);

}

if(f.equals("Water")) {

String query="delete from water where brand=?";

String query1="select wid from water where brand=?";

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

PreparedStatement ps=con.prepareStatement(query);

PreparedStatement ps1=con.prepareStatement(query1);

ps.setString(1,a1);

ps1.setString(1,a1);

ResultSet rs=ps1.executeQuery();

if(rs==null) {
JOptionPane.showMessageDialog(frame, "Item name doesnt exist");
}

else {

```

```

        rs.next();

        int s=rs.getInt(1);

        String query2="delete from waterap where wid=?";

        PreparedStatement ps2=con.prepareStatement(query2);

        ps2.setInt(1, s);

        int h=ps2.executeUpdate();

        System.out.println(h);

        if(h==1) {

JOptionPane.showMessageDialog(frame,"Deleted");

        }

    }

}

if(f.equals("Milk")) {

        String query="delete from milk where brand=?";

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

        PreparedStatement ps=con.prepareStatement(query);

        ps.setString(1,a1);

        int y=ps.executeUpdate();

        if(y==0) {

JOptionPane.showMessageDialog(frame, "Item name doesnt exist");

        }

        else {

JOptionPane.showMessageDialog(frame,"Deleted");

        }

    }

if(f.equals("Beverages")) {

        String query="delete from beverages where brand=?";

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

```

```

        PreparedStatement ps=con.prepareStatement(query);
        ps.setString(1,a1);
        int y=ps.executeUpdate();
        if(y==0) {
            JOptionPane.showMessageDialog(frame, "Item name doesnt exist");
        }else {

JOptionPane.showMessageDialog(frame,"Deleted");

        }

    }
    if(f.equals("Solid")) {
        String query="delete from solid where sname=?";
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
        PreparedStatement ps=con.prepareStatement(query);
        ps.setString(1,a1);
        int y=ps.executeUpdate();
        if(y==0) {
            JOptionPane.showMessageDialog(frame, "Item name doesnt exist");
        }else {

JOptionPane.showMessageDialog(frame,"Deleted");

        }    }
    if(f.equals("Fruits")) {
        String query="delete from fruit where name=?";
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
        PreparedStatement ps=con.prepareStatement(query);
        ps.setString(1,a1);
        int y=ps.executeUpdate();
        if(y==0) {
            JOptionPane.showMessageDialog(frame, "Item name doesnt exist");

```

```

        }else {

JOptionPane.showMessageDialog(frame,"Deleted");

        }

    }

    if(f.equals("Seasonal Fruits")){
        String query="delete from seasonalfruit where name=?";

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

        PreparedStatement ps=con.prepareStatement(query);
        ps.setString(1,a1);
        int y=ps.executeUpdate();

        if(y==0) {
JOptionPane.showMessageDialog(frame, "Item name doesnt exist");
        }else {

JOptionPane.showMessageDialog(frame,"Deleted");

        }

    }

    if(f.equals("Cereals")) {

        String query="delete from cereal where name=?";

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

        PreparedStatement ps=con.prepareStatement(query);
        ps.setString(1,a1);
        int y=ps.executeUpdate();

        if(y==0) {
JOptionPane.showMessageDialog(frame, "Item name doesnt exist");
        }else {

JOptionPane.showMessageDialog(frame,"Deleted");

        }

    }

```

```

        }
        catch(ClassNotFoundException | SQLException e2) {
            e2.printStackTrace();
        }
        textField.setText("");
        textField_1.setEditable(true);
        textField_2.setEditable(true);
        comboBox.setSelectedIndex(0);
    }
    else {
        textField.setText("");
        textField_1.setEditable(true);
        textField_2.setEditable(true);
        comboBox.setSelectedIndex(0);
    }
}
else { JOptionPane.showMessageDialog(frame,"Write the name
and select the category you want to delete");
}
});
btnNewButton_2.setBounds(354, 166, 89, 23);
contentPane.add(btnNewButton_2);
JButton btnNewButton_3 = new JButton("Back");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
btnNewButton_3.setBounds(354, 228, 89, 23);
contentPane.add(btnNewButton_3);
JLabel lblAvailability = new JLabel("Availability");
lblAvailability.setFont(new Font("Tahoma", Font.PLAIN, 14));
lblAvailability.setBounds(30, 205, 86, 20);
contentPane.add(lblAvailability);

```

```

        textField_3 = new JTextField();
        textField_3.setColumns(10);
        textField_3.setBounds(158, 207, 86, 20);
        contentPane.add(textField_3);
        textField_3.setEditable(false);
    }
}

```

## Items.java

```

package dbmsp;

import java.awt.Component;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JComboBox;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JTable;
import javax.swing.JScrollPane;
import java.sql.*;
import java.awt.Font;

```

```

public class items extends JFrame {

    private static final long serialVersionUID = 1L;

    private JPanel contentPane;

String          s1[]={ "Select","Vegetables","Water","Milk","Beverages","Solid","Fruits","Seasonal
Fruits","Cereals"};

    String veg[]={ "Brinjal","Lady's finger","Tomato","Potato","Bottle Guard","Beans"};
    String sol[]={ "Cheese","Jaggery","Paneer","Bread","Rice","Pasta","Maggi"};
    String fru[]={ "Banana","Apple","Guava","Carrot","Kiwi","Plum"};
    String sesfru[]={ "Mango","Orange","Custard Apple","Sweet Lime"};
    String cer[]={ "Corn Flakes","Chocos","Oats","Honey Rings"};
    String units[]={ "kg","g","L","ml","dozen","none"};
    String cat[]={ "SubCategory","niha"};
    String milkb[]={ "Amul","Jersey","Arogya"};
    String waterb[]={ "Oxygen","Bisleri","Kinley"};
    String bevr[]={ "Red Bull","Coca Cola","Sprite","Maaza","Sting","Flavored Milk"};
    private JTextField textField;
    Component frame=null;
    private JTable table;
    DefaultTableModel model;
    DefaultTableModel model1;
    private JTextField textField_1;
    private JTextField textField_2;
    private JScrollPane scrollPane;
    @SuppressWarnings("rawtypes")
    JComboBox comboBox ;
    private JTable table_1;
    public static int id=0;
    public int sum=0;
    public static void itemscreen() {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    items frame = new items();

```



```

        frame.setTitle(login.name);

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

});
}

```

```

@SuppressWarnings({ "unchecked", "rawtypes" })
public items() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 861, 707);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    scrollPane = new JScrollPane();
    scrollPane.setBounds(10, 345, 649, 312);
    contentPane.add(scrollPane);
    scrollPane.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            int i=table.getSelectedRow();
            textField_1.setText(model.getValueAt(i, 2).toString());
            textField_2.setText(model.getValueAt(i, 4).toString());
        }
    });

    JScrollPane scrollPane_1 = new JScrollPane();
    scrollPane_1.setBounds(267, 41, 568, 226);
    contentPane.add(scrollPane_1);
}

```

```

table_1 = new JTable();
model1=new DefaultTableModel();
Object t1[]= {"id","Item","price","quantity","total"};
Object r1[]=new Object[5];
model1.setColumnIdentifiers(t1);
table_1.setModel(model1);
table_1.setDefaultEditor(Object.class,null);
scrollPane_1.setViewportView(table_1);

table = new JTable();
model=new DefaultTableModel();
table.setDefaultEditor(Object.class, null);

Object column1[]= {"vid","item_id","vegetable_name","availability","price"};
Object column2[]= {"wid","liquid_id","brand","availability","price","quantity"};
Object column3[]= {"milk_id","liquid_id","brand","availability","price","quantity"};
Object column4[]= {"beverages_id","liquid_id","brand","availability","price","quantity"};
Object column5[]= {"solid_id","item_id","sname","availability","price"};
Object column6[]= {"fruit_id","item_id","name","availability","price"};
Object column7[]= {"sfruit_id","item_id","name","availability","price","dom"};
Object column8[]= {"cereal_id","item_id","name","availability","price"};

Object row1[]=new Object[5];
Object row2[]=new Object[6];
Object row3[]=new Object[6];
Object row4[]=new Object[6];
Object row5[]=new Object[5];
Object row6[]=new Object[5];
Object row7[]=new Object[6];
Object row8[]=new Object[5];

table.setModel(model);
scrollPane.setViewportView(table);
comboBox = new JComboBox(s1);
comboBox.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```

try {
String s=(String)comboBox.getSelectedItemAt();
if(s.equals("Vegetables")) {
    model.setColumnIdentifiers(column1);
    model.setRowCount(0);
    String query="select * from vegetable";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
    PreparedStatement ps=con.prepareStatement(query);
    ResultSet rs=ps.executeQuery();
    while(rs.next()) {
        row1[0]=rs.getInt(1);
        row1[1]=rs.getInt(2);
        row1[2]=rs.getString(3);
        row1[3]=rs.getInt(4);
        row1[4]=rs.getInt(5);
        model.addRow(row1);
    }

}

if(s.equals("Water")) {
    model.setColumnIdentifiers(column2);
    model.setRowCount(0);

String query=" select * from water natural join waterap order by wid,availability,quantity,price";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
    PreparedStatement ps=con.prepareStatement(query);
    ResultSet rs=ps.executeQuery();
    while(rs.next()) {
        row2[0]=rs.getInt(1);
        row2[1]=rs.getInt(2);

```

```

        row2[2]=rs.getString(3);
        row2[3]=rs.getInt(4);
        row2[4]=rs.getInt(5);
        row2[5]=rs.getInt(6);
        model.addRow(row2);
    }
}

if(s.equals("Milk")) {
    model.setColumnIdentifiers(column3);
    model.setRowCount(0);

    String query=" select * from milk natural join milkap order by
milk_id,availability,quantity,price";

    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
    PreparedStatement ps=con.prepareStatement(query);
    ResultSet rs=ps.executeQuery();
    while(rs.next()) {
        row3[0]=rs.getInt(1);
        row3[1]=rs.getInt(2);
        row3[2]=rs.getString(3);
        row3[3]=rs.getInt(4);
        row3[4]=rs.getInt(5);
        row3[5]=rs.getInt(6);
        model.addRow(row3);
    }
}

if(s.equals("Beverages")) {
    model.setColumnIdentifiers(column4);
    model.setRowCount(0);

    String query=" select * from beverages natural join beveragesap order
by beverages_id,availability,quantity,price";

    Class.forName("oracle.jdbc.driver.OracleDriver");

```

### Connection

```
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

PreparedStatement ps=con.prepareStatement(query);

ResultSet rs=ps.executeQuery();

while(rs.next()) {

    row4[0]=rs.getInt(1);

    row4[1]=rs.getInt(2);

    row4[2]=rs.getString(3);

    row4[3]=rs.getInt(4);

    row4[4]=rs.getInt(5);

    row4[5]=rs.getInt(6);

    model.addRow(row4);

}

}

if(s.equals("Solid")) {

    model.setColumnIdentifiers(column5);

    model.setRowCount(0);

    String query="select * from solid";

    Class.forName("oracle.jdbc.driver.OracleDriver");

    Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

    PreparedStatement ps=con.prepareStatement(query);

    ResultSet rs=ps.executeQuery();

    while(rs.next()) {

        row5[0]=rs.getInt(1);

        row5[1]=rs.getInt(2);

        row5[2]=rs.getString(3);

        row5[3]=rs.getInt(4);

        row5[4]=rs.getInt(5);

        model.addRow(row5);

    }

}

if(s.equals("Fruits")) {
```

```

        model.setColumnIdentifiers(column6);
        model.setRowCount(0);
        String query="select * from fruit";
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
        PreparedStatement ps=con.prepareStatement(query);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            row6[0]=rs.getInt(1);
            row6[1]=rs.getInt(2);
            row6[2]=rs.getString(3);
            row6[3]=rs.getInt(4);
            row6[4]=rs.getInt(5);
            model.addRow(row6);
        }
    }
    if(s.equals("Seasonal Fruits")) {
        model.setColumnIdentifiers(column7);
        model.setRowCount(0);
        String query="select * from seasonalfruit";
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
        PreparedStatement ps=con.prepareStatement(query);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            row7[0]=rs.getInt(1);
            row7[1]=rs.getInt(2);
            row7[2]=rs.getString(3);
            row7[3]=rs.getInt(4);
            row7[4]=rs.getInt(5);
            row7[5]=rs.getDate(6);

```

```

        model.addRow(row7);
    }
}
if(s.equals("Cereals")) {
    model.setColumnIdentifiers(column8);
    model.setRowCount(0);
    String query="select * from cereal";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
    PreparedStatement ps=con.prepareStatement(query);
    ResultSet rs=ps.executeQuery();
    while(rs.next()) {
        row8[0]=rs.getInt(1);
        row8[1]=rs.getInt(2);
        row8[2]=rs.getString(3);
        row8[3]=rs.getInt(4);
        row8[4]=rs.getInt(5);
        model.addRow(row8);
    }
}
}
catch(SQLException|ClassNotFoundException e1) {
    e1.printStackTrace();
}
});

model1.setRowCount(0);
comboBox.setToolTipText("Items");
comboBox.setBounds(50, 32, 116, 33);
contentPane.add(comboBox);
textField = new JTextField();
textField.setBounds(116, 227, 122, 33);
contentPane.add(textField);
textField.setColumns(10);

```

```

JLabel lblNewLabel = new JLabel("Quantity");
lblNewLabel.setBounds(35, 232, 73, 23);
contentPane.add(lblNewLabel);
JButton btnNewButton = new JButton("Add To Bill");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(comboBox.getSelectedIndex()==0) {
            JOptionPane.showMessageDialog(frame, "Select any category");
        }
        else if(textField.getText().equals("")) {
            JOptionPane.showMessageDialog(frame, "Please enter the quantity");
        }
        else if(textField.getText().length()!=0) {

            try {
                int y=++id;
                String name= textField_1.getText();
                int price=Integer.parseInt(textField_2.getText());
                int quan=Integer.parseInt(textField.getText());
                int tot=price*quan;
                try{

                    String query="insert into prevorder values(?,?,?, ?,?,?)";
                    Class.forName("oracle.jdbc.driver.OracleDriver");
                    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
                    PreparedStatement ps=con.prepareStatement(query);
                    ps.setInt(1,login.cid );
                    ps.setInt(2, y);
                    ps.setString(3,name);
                    ps.setInt(4, price);
                    ps.setInt(5, quan);
                    ps.setInt(6, tot);

```



```

        sum+=tot;
        ps.executeUpdate();
    }
    catch(SQLException|ClassNotFoundException e1) {
        e1.printStackTrace();
    }
    r1[0]=y;
    r1[1]=name;
    r1[2]=price;
    r1[3]=quan;
    r1[4]=tot;
    model1.addRow(r1);

}

catch(NumberFormatException e1) {
    JOptionPane.showMessageDialog(frame,"Please enter a number");
}}

}

});

btnNewButton.setBounds(35, 289, 116, 39);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("Clear");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText("");
        textField_1.setText("");
        textField_2.setText("");
        comboBox.setSelectedIndex(0);
    }
}

```

```
});
btnNewButton_1.setBounds(187, 289, 116, 39);
contentPane.add(btnNewButton_1);
```

```
JButton btnNewButton_1_1 = new JButton("Back");
btnNewButton_1_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
```

```
btnNewButton_1_1.setBounds(338, 289, 116, 39);
contentPane.add(btnNewButton_1_1);
```

```
JButton btnNewButton_2 = new JButton("Buy");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
```

```
        try {
            String query="insert into sales
values(seq_sales.nextval,?,sysdate,?,to_char(sysdate,'hh24:mi:ss'))";
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
            PreparedStatement ps=con.prepareStatement(query);
            ps.setInt(1,login.cid);
            ps.setInt(2,sum);
            ps.executeUpdate();
            JOptionPane.showMessageDialog(null,"Your total is "+sum);
            dispose();
        }
        catch(SQLException|ClassNotFoundException e1) {
            e1.printStackTrace();
        }
    }
}
```

```

});

btnNewButton_2.setBounds(496, 287, 116, 42);

contentPane.add(btnNewButton_2);


JLabel lblName = new JLabel("Item");
lblName.setBounds(35, 94, 40, 23);
contentPane.add(lblName);


textField_1 = new JTextField();
textField_1.setColumns(10);
textField_1.setBounds(116, 89, 122, 33);
contentPane.add(textField_1);


JLabel lblName_1 = new JLabel("Price");
lblName_1.setBounds(35, 164, 65, 23);
contentPane.add(lblName_1);


textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBounds(116, 159, 122, 33);
contentPane.add(textField_2);


textField_1.setEditable(false);
textField_2.setEditable(false);

JLabel lblBilling = new JLabel("Bill");
lblBilling.setFont(new Font("Tahoma", Font.PLAIN, 20));
lblBilling.setBounds(550, 11, 139, 23);
contentPane.add(lblBilling);

}

}

```

## **Insadmin.java**

```
package dbmsp;
```

```

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class insadmin extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;

    public static void adminscreen() {

        EventQueue.invokeLater(new Runnable() {

            public void run() {

                try {

                    insadmin frame = new insadmin();
                    frame.setTitle("admin login");
                    frame.setVisible(true);

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

            }

        });
    }

    public insadmin() {

        //setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        dispose();
        setBounds(100, 100, 529, 378);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
    }

```

```

JButton btnNewButton = new JButton("Add Items");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        additems.screen();
    }
});
btnNewButton.setBounds(164, 70, 140, 39);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("Sales Report");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        frame.framescreen();
    }
});
btnNewButton_1.setBounds(164, 132, 140, 39);
contentPane.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("Customer Details");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cust.custscreen();
    }
});
btnNewButton_2.setBounds(164, 193, 140, 39);
contentPane.add(btnNewButton_2);

JButton btnNewButton_3 = new JButton("Logout");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(usersc.am==1) {
            dispose();
        }
    }
});

```

```

        }
        dispose();
    }
});
btnNewButton_3.setBounds(164, 255, 140, 39);
contentPane.add(btnNewButton_3);

JButton btnNewButton_4 = new JButton("Employees");
btnNewButton_4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        vemp.screen();
    }
});
btnNewButton_4.setBounds(164, 11, 140, 39);
contentPane.add(btnNewButton_4);
}
}

```

## **Cust.java**

```

package dbmsp;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.table.DefaultTableModel;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

```

```

import java.awt.event.ActionEvent;

import javax.swing.JTable;

import javax.swing.JLabel;

import javax.swing.JOptionPane;


public class cust extends JFrame {

    private static final long serialVersionUID = 1L;

    private JPanel contentPane;

    private JLabel lblNewLabel;

    private JTable table;

    DefaultTableModel model;

    public static void custscreen() {

        EventQueue.invokeLater(new Runnable() {

            public void run() {

                try {

                    cust frame = new cust();

                    frame.setTitle("Customer Details");

                    frame.setVisible(true);

                } catch (Exception e) {

                    e.printStackTrace();

                }

            }

        });

    }

    public cust() {

        dispose();

        setBounds(100, 100, 1170, 540);

        contentPane = new JPanel();

        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);

        contentPane.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();

```

```

scrollPane.setBounds(241, 39, 903, 434);

contentPane.add(scrollPane);

table = new JTable();

model=new DefaultTableModel();

Object      column[]=      {"customer_id","first      name","last
name","email_id","gender","age","Phone no","address"};

Object row[]=new Object[8];

model.setColumnIdentifiers(column);

table.setModel(model);

table.getColumnModel().getColumn(3).setPreferredWidth(90);

table.getColumnModel().getColumn(5).setPreferredWidth(3);

table.getColumnModel().getColumn(7).setPreferredWidth(90);

scrollPane.setViewportView(table);

JButton btnNewButton = new JButton("View Customer");

btnNewButton.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        model.setRowCount(0);

        String y=JOptionPane.showInputDialog(null,"Enter the customer id");

        String query="select * from customer where customer_id=?";

        try {

            Class.forName("oracle.jdbc.driver.OracleDriver");

            Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

            PreparedStatement ps=con.prepareStatement(query);

            ps.setInt(1,Integer.parseInt(y));

            ResultSet rs=ps.executeQuery();

            if(rs.next()==false) {
JOptionPane.showMessageDialog(null, "No customer_id");
            }

            else {

                row[0]=rs.getInt(1);

                row[1]=rs.getString(2);

                row[2]=rs.getString(3);

```



```

        row[3]=rs.getString(4);
        row[4]=rs.getString(5);
        row[5]=rs.getInt(6);
        row[6]=rs.getString(7);
        row[7]=rs.getString(8);
        model.addRow(row);
        ps.close();
    }
}

catch(SQLException|ClassNotFoundException e11) {
    e11.printStackTrace();
}

}

});

btnNewButton.setBounds(32, 39, 125, 35);
contentPane.add(btnNewButton);

table = new JTable();

JScrollPane jScrollPane = new JScrollPane();
jScrollPane.setViewportView(table);

JButton btnNewButton_1 = new JButton("View all customers");
btnNewButton_1.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        model.setRowCount(0);

        String query="select * from customer";

try {

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

        PreparedStatement ps=con.prepareStatement(query);

        ResultSet rs1=ps.executeQuery();

        while(rs1.next()) {

            row[0]=rs1.getInt(1);

            row[1]=rs1.getString(2);

```

```

        row[2]=rs1.getString(3);
        row[3]=rs1.getString(4);
        row[4]=rs1.getString(5);
        row[5]=rs1.getInt(6);
        row[6]=rs1.getString(7);
        row[7]=rs1.getString(8);
        model.addRow(row);
    }
}
catch(SQLException|ClassNotFoundException e11) {
    e11.printStackTrace();
}}
});
btnNewButton_1.setBounds(32, 85, 148, 35);
contentPane.add(btnNewButton_1);
lblNewLabel = new JLabel("Select to view the details of customer ");
lblNewLabel.setBounds(47, 11, 315, 33);
contentPane.add(lblNewLabel);
JButton btnNewButton_1_1 = new JButton("Back");
btnNewButton_1_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
btnNewButton_1_1.setBounds(32, 131, 104, 35);
contentPane.add(btnNewButton_1_1);
}
}

```

## Register.java

```

package dbmsp;
import java.awt.EventQueue;
import javax.swing.JFrame;

```

```

import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JRadioButton;
import javax.swing.JTextArea;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import java.awt.Color;
import java.awt.Component;
import java.awt.Font;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.*;

public class register extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField textField;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_3;
    private JLabel lblNewLabel_4;
    private JLabel lblNewLabel_5;
    private JLabel lblNewLabel_6;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JRadioButton rdbtnNewRadioButton;
    private JRadioButton rdbtnNewRadioButton_1;
    private JRadioButton rdbtnNewRadioButton_2;
    private JTextArea textArea;
    private JTextField textField_4;

```

```

private JButton btnNewButton;

private JButton btnClear;

private JLabel lblNewLabel_7;

ButtonGroup g1;

Component frame=null;

public static String fn="";

public static int id;

public static void registerscreen() {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                register frame = new register();
                frame.setTitle("User Registration Form");
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public register() {
    g1=new ButtonGroup();
    dispose();
    setBounds(400, 400, 697, 465);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("First Name");
    lblNewLabel.setBounds(35, 53, 85, 25);
    contentPane.add(lblNewLabel);

    textField = new JTextField();
    textField.setBounds(187, 53, 115, 25);

```

```

contentPane.add(textField);

textField.setColumns(10);

JLabel lblNewLabel_1 = new JLabel("Last Name");
lblNewLabel_1.setBounds(35, 89, 85, 25);
contentPane.add(lblNewLabel_1);

lblNewLabel_2 = new JLabel("Email ID");
lblNewLabel_2.setBounds(35, 120, 85, 25);
contentPane.add(lblNewLabel_2);

lblNewLabel_3 = new JLabel("Gender");
lblNewLabel_3.setBounds(35, 156, 85, 25);
contentPane.add(lblNewLabel_3);

lblNewLabel_4 = new JLabel("Address");
lblNewLabel_4.setBounds(35, 192, 85, 25);
contentPane.add(lblNewLabel_4);

lblNewLabel_5 = new JLabel("Age");
lblNewLabel_5.setBounds(35, 259, 85, 25);
contentPane.add(lblNewLabel_5);

lblNewLabel_6 = new JLabel("Phone Number");
lblNewLabel_6.setBounds(35, 295, 85, 25);
contentPane.add(lblNewLabel_6);

textField_1 = new JTextField();
textField_1.setBounds(187, 89, 115, 25);
contentPane.add(textField_1);
textField_1.setColumns(10);

textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBounds(187, 125, 115, 25);
contentPane.add(textField_2);

textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBounds(187, 259, 115, 25);
contentPane.add(textField_3);

```

```

rdbtnNewRadioButton = new JRadioButton("Male");
rdbtnNewRadioButton.setBounds(193, 157, 109, 23);
contentPane.add(rdbtnNewRadioButton);
rdbtnNewRadioButton_1 = new JRadioButton("Female");
rdbtnNewRadioButton_1.setBounds(298, 157, 109, 23);
contentPane.add(rdbtnNewRadioButton_1);
rdbtnNewRadioButton_2 = new JRadioButton("Others");
rdbtnNewRadioButton_2.setBounds(409, 157, 109, 23);
contentPane.add(rdbtnNewRadioButton_2);
g1.add(rdbtnNewRadioButton);
g1.add(rdbtnNewRadioButton_1);
g1.add(rdbtnNewRadioButton_2);
textArea = new JTextArea();
textArea.setBounds(187, 192, 209, 51);
contentPane.add(textArea);
textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBounds(187, 295, 115, 25);
contentPane.add(textField_4);
btnNewButton = new JButton("Save");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(textField.getText().length()<3) {
JOptionPane.showMessageDialog(frame, "Please fill the first name field");
        }
        else if(textField_1.getText().length()<3) {
JOptionPane.showMessageDialog(frame, "Please fill the last name field");
        }
        else if(textField_2.getText().length()<3) {
            JOptionPane.showMessageDialog(frame, "Please enter the email");
        }
        else if(g1.getSelection()==null) {
JOptionPane.showMessageDialog(frame, "Please select the gender");
        }
    }
});

```

```

    }
    else if(textArea.getText().length()<1) {
JOptionPane.showMessageDialog(frame, "Please fill the address field");
    }
    else if(textField_3.getText().length()<1) {
        JOptionPane.showMessageDialog(frame, "Please fill age field");
    }
    else if(textField_4.getText().length()<3) {
JOptionPane.showMessageDialog(frame, "Please fill phone number field");
    }
    else {
        String gender="";
        if(rdbtnNewRadioButton.isSelected()) {
            gender="male";
        }
        else if(rdbtnNewRadioButton_1.isSelected()) {
            gender="female";
        }
        else {
            gender="others";
        }
        dispose();

        String query="insert into customer
values(seq_cust.nextval,?,?,?,?,?,?)";

        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
            PreparedStatement ps=con.prepareStatement(query);
            ps.setString(1, textField.getText());
            ps.setString(2, textField_1.getText());
            ps.setString(3, textField_2.getText());
            ps.setString(4, gender);
            ps.setInt(5, Integer.parseInt(textField_3.getText()));

```

```

        ps.setString(6, textField_4.getText());
        ps.setString(7, textArea.getText());
        ps.execute();

        ps.close();
        con.close();
        fn=textField.getText();

        String query1="select max(customer_id) from customer";
        Connection
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
        PreparedStatement ps1=con1.prepareStatement(query1);
        ResultSet rs=ps1.executeQuery();
        rs.next();
        id=rs.getInt(1);
        logsc.screen();
    }

    catch(ClassNotFoundException|SQLException e1 ) {
        e1.printStackTrace();
    }

    }

    }

});

btnNewButton.setBounds(133, 343, 95, 36);
contentPane.add(btnNewButton);
btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText("");
        textField_1.setText("");
        textField_2.setText("");
        textField_3.setText("");
        textField_4.setText("");

```



```

        g1.clearSelection();
        textArea.setText("");
    }
});
btnClear.setBounds(334, 343, 102, 36);
contentPane.add(btnClear);
lblNewLabel_7 = new JLabel("Register Form");
lblNewLabel_7.setFont(new Font("Tahoma", Font.PLAIN, 20));
lblNewLabel_7.setBackground(new Color(240, 240, 240));
lblNewLabel_7.setBounds(58, 8, 157, 34);
contentPane.add(lblNewLabel_7);
}
}

```

## **Cedit.java**

```

package dbmsp;

import java.awt.EventQueue;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import java.awt.event.ActionListener;

```

```

import java.awt.event.ActionEvent;

public class cedit extends JFrame {

    private static final long serialVersionUID = 1L;

    private JPanel contentPane;
    private JTextField textField;
    private JLabel lblNewLabel_2;
    private JLabel lblNewLabel_3;
    private JLabel lblNewLabel_4;
    private JLabel lblNewLabel_5;
    private JLabel lblNewLabel_6;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JRadioButton rdbtnNewRadioButton;
    private JRadioButton rdbtnNewRadioButton_1;
    private JRadioButton rdbtnNewRadioButton_2;
    private JTextArea textArea;
    private JTextField textField_4;
    ButtonGroup g1;
    private JButton btnNewButton_1;
    private JButton btnNewButton_2;

    public static void screen() {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    cedit frame = new cedit();
                    frame.setTitle(login.name);
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        })
    }
}

```

```

    });
}
public cedit() {
    g1=new ButtonGroup();
    dispose();
    setBounds(400, 400, 697, 465);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    JLabel lblNewLabel = new JLabel("First Name");
    lblNewLabel.setBounds(35, 53, 85, 25);
    contentPane.add(lblNewLabel);
    textField = new JTextField();
    textField.setBounds(187, 53, 115, 25);
    contentPane.add(textField);
    textField.setColumns(10);
    JLabel lblNewLabel_1 = new JLabel("Last Name");
    lblNewLabel_1.setBounds(35, 89, 85, 25);
    contentPane.add(lblNewLabel_1);
    lblNewLabel_2 = new JLabel("Email ID");
    lblNewLabel_2.setBounds(35, 120, 85, 25);
    contentPane.add(lblNewLabel_2);
    lblNewLabel_3 = new JLabel("Gender");
    lblNewLabel_3.setBounds(35, 156, 85, 25);
    contentPane.add(lblNewLabel_3);
    lblNewLabel_4 = new JLabel("Address");
    lblNewLabel_4.setBounds(35, 192, 85, 25);
    contentPane.add(lblNewLabel_4);
    lblNewLabel_5 = new JLabel("Age");
    lblNewLabel_5.setBounds(35, 259, 85, 25);
    contentPane.add(lblNewLabel_5);
}

```

```

lblNewLabel_6 = new JLabel("Phone Number");
lblNewLabel_6.setBounds(35, 306, 85, 25);
contentPane.add(lblNewLabel_6);
textField_1 = new JTextField();
textField_1.setBounds(187, 89, 115, 25);
contentPane.add(textField_1);
textField_1.setColumns(10);
textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBounds(187, 125, 115, 25);
contentPane.add(textField_2);
textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBounds(187, 259, 115, 25);
contentPane.add(textField_3);
rdbtnNewRadioButton = new JRadioButton("Male");
rdbtnNewRadioButton.setBounds(193, 157, 109, 23);
contentPane.add(rdbtnNewRadioButton);
rdbtnNewRadioButton_1 = new JRadioButton("Female");
rdbtnNewRadioButton_1.setBounds(298, 157, 109, 23);
contentPane.add(rdbtnNewRadioButton_1);
rdbtnNewRadioButton_2 = new JRadioButton("Others");
rdbtnNewRadioButton_2.setBounds(409, 157, 109, 23);
contentPane.add(rdbtnNewRadioButton_2);
g1.add(rdbtnNewRadioButton);
g1.add(rdbtnNewRadioButton_1);
g1.add(rdbtnNewRadioButton_2);
textArea = new JTextArea();
textArea.setBounds(187, 192, 209, 51);
contentPane.add(textArea);
textField_4 = new JTextField();
textField_4.setColumns(10);

```

```

textField_4.setBounds(187, 306, 115, 25);
contentPane.add(textField_4);
try {
    String query="select * from customer where customer_id=?";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
    PreparedStatement ps=con.prepareStatement(query);
    ps.setInt(1, login.cid);
    ResultSet rs=ps.executeQuery();
    rs.next();
    textField.setText(rs.getString(2));
    textField_1.setText(rs.getString(3));
    textField_2.setText(rs.getString(4));//fn,ln email age phonen
    textField_3.setText(rs.getInt(6)+"");
    textField_4.setText(rs.getString(7));
    textArea.setText(rs.getString(8));
    String y=rs.getString(5);
    if(y.equals("female")) {
        rdbtnNewRadioButton_1.setSelected(true);
    }
    else if(y.equals("male")) {
        rdbtnNewRadioButton.setSelected(true);
    }
    else {
        rdbtnNewRadioButton_2.setSelected(true);
    }

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

btnNewButton_1 = new JButton("Save");
btnNewButton_1.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            try {
                if(textField.getText().length()<3) {
JOptionPane.showMessageDialog(null, "Please fill the first name field");
                }
                else if(textField_1.getText().length()<3) {
JOptionPane.showMessageDialog(null, "Please fill the last name field");
                }
                else if(textField_2.getText().length()<3) {
JOptionPane.showMessageDialog(null, "Please enter the email");
                }
                else if(g1.getSelection()==null) {
JOptionPane.showMessageDialog(null, "Please select the gender");
                }
                else if(textArea.getText().length()<1) {
JOptionPane.showMessageDialog(null, "Please fill the address field");
                }
                else if(textField_3.getText().length()<1) {
JOptionPane.showMessageDialog(null, "Please fill age field");
                }
                else if(textField_4.getText().length()<3) {
JOptionPane.showMessageDialog(null, "Please fill phone number field");
                }
                else {
                    String gender="";
                    if(rdbtnNewRadioButton.isSelected()) {
                        gender="male";
                    }
                    else if(rdbtnNewRadioButton_1.isSelected()) {
                        gender="female";
                    }
                    else {
                        gender="others";
                    }
                }
            }
        }
    }

```

```

        String query="update customer set
cfname=?,cname=?,email_id=?,gender=?,age=?,phone_no=?,address=? where customer_id=?";

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

        PreparedStatement ps=con.prepareStatement(query);
        ps.setString(1,textField.getText());
        ps.setString(2,textField_1.getText());
        ps.setString(3,textField_2.getText());
        ps.setString(4,gender);
        ps.setInt(5,Integer.parseInt(textField_3.getText()));
        ps.setString(6, textField_4.getText());
        ps.setString(7, textArea.getText());
        ps.setInt(8,login.cid);
        ps.execute();
        JOptionPane.showMessageDialog(null, "Successfully saved");
    }
    dispose();
} catch(SQLException|ClassNotFoundException e1) {
    e1.printStackTrace();
}

}

});

btnNewButton_1.setBounds(100, 356, 100, 31);
contentPane.add(btnNewButton_1);

btnNewButton_2 = new JButton("Back");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});

btnNewButton_2.setBounds(322, 356, 100, 31);
contentPane.add(btnNewButton_2);

```

```
}}
```

## **Frame.java**

```
package dbmsp;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.UIManager;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.awt.event.ActionEvent;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import com.toedter.calendar.JDateChooser;
import javax.swing.JRadioButton;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JTextField;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;

public class frame {

    private JFrame frame;
    private JTextField textField;
    private ButtonGroup g1;
    int sum=0;
    public static void framescreen() {
```



```

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

    public frame() {
        initialize();
    }

    private void initialize() {
        g1=new ButtonGroup();
        frame = new JFrame();
        frame.getContentPane().setBackground(UIManager.getColor("CheckBox.light"));
        frame.getContentPane().setLayout(null);
        frame.setTitle("Sales Info");
        JLabel lblNewLabel = new JLabel("Choose the date for sales,which you want to generate");
        lblNewLabel.setBounds(51, 11, 359, 32);
        frame.getContentPane().add(lblNewLabel);
        JDateChooser dateChooser = new JDateChooser();
        dateChooser.setBounds(243, 72, 142, 32);
        frame.getContentPane().add(dateChooser);
        JRadioButton rdbtnNewRadioButton = new JRadioButton("Date");
        rdbtnNewRadioButton.setBounds(66, 100, 109, 23);
        frame.getContentPane().add(rdbtnNewRadioButton);
        g1.add(rdbtnNewRadioButton);
        JButton btnNewButton = new JButton("Get date");
        btnNewButton.addActionListener(new ActionListener() {

```

```

public void actionPerformed(ActionEvent e) {

    try {
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-yyyy");
        String date =sdf.format(dateChooser.getDate().getTime());
        java.util.Date dt_1=null;
        try {
            dt_1=sdf.parse(date);
        } catch (ParseException e1) {
            e1.printStackTrace();
        }

        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
        if(rdbtnNewRadioButton.isSelected()) {
            textField.setText("");
            PreparedStatement          ps=con.prepareStatement("select
sum(final_price) from sales where dop like ?");
            ps.setDate(1,new java.sql.Date(dt_1.getTime()));
            ResultSet rs=ps.executeQuery();
            rs.next();
            textField.setText(rs.getInt(1)+"");
        }
        else {
            JOptionPane.showMessageDialog(null, "click on date and then select the date");
        }

    } catch(SQLException|ClassNotFoundException e1) {
        e1.printStackTrace();
    }

});

btnNewButton.setBounds(51, 204, 109, 23);

```

```

frame.getContentPane().add(btnNewButton);

textField = new JTextField();

textField.setBounds(243, 115, 126, 47);

frame.getContentPane().add(textField);

textField.setColumns(10);

textField.setEditable(false);

JButton btnNewButton_1 = new JButton("Back");

btnNewButton_1.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        frame.dispose();

    }

});

btnNewButton_1.setBounds(191, 204, 89, 23);

frame.getContentPane().add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("Clear");

btnNewButton_2.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        g1.clearSelection();

        dateChooser.setCalendar(null);

        textField.setText("");

    }

});

btnNewButton_2.setBounds(308, 204, 89, 23);

frame.getContentPane().add(btnNewButton_2);

frame.setBounds(100, 100, 450, 300);

frame.dispose();

}

}

```

## **Emp.java**

```

package dbmsp;

import java.sql.*;

import java.sql.Date;

```

```

import java.sql.Connection;
import java.text.SimpleDateFormat;
import java.awt.Component;
import java.awt.EventQueue;
import java.text.ParseException;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import com.toedter.calendar.JDateChooser;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JTable;
import javax.swing.JScrollPane;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
public class emp extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField textField;
    private JTextField textField_1;
    private JLabel lblJobTitle;
    private JLabel lblLastName_2;
    private JLabel lblLastName_3;
    private JLabel lblLastName_4;

```

```

private JLabel lblLastName_5;
private JLabel lblLastName_6;
private JTextField textField_2;
private JTextField textField_3;
private JTextField textField_4;
Component frame=null;
String jt[]= {"Manager","Senior Analyst","Analyst","Operator"};
private JTextField textField_5;
private static final String regex = "^(.+ )@(.+)$";
private JTable table;
DefaultTableModel model;
private JScrollPane scrollPane;
private JButton btnNewButton_3;
@SuppressWarnings("rawtypes")
private JComboBox comboBox ;
private JDateChooser dateChooser;
private JDateChooser dateChooser_1;
public static void empscreen() {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                emp frame = new emp();
                frame.setTitle("Employees");
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
@SuppressWarnings({ "unchecked", "rawtypes" })
public emp() {

```

```

dispose();
setBounds(100, 100, 1223, 625);

contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

scrollPane = new JScrollPane();
scrollPane.addMouseListener(new MouseAdapter() {

    @Override
    public void mouseClicked(MouseEvent e) {

        int i=table.getSelectedRow();

        textField.setText(model.getValueAt(i, 1).toString());
        textField_1.setText(model.getValueAt(i, 2).toString());
        comboBox.setSelectedItem(model.getValueAt(i, 3));
        dateChooser.setDate((java.util.Date) model.getValueAt(i, 4));
        dateChooser_1.setDate((java.util.Date) model.getValueAt(i, 5));
        textField_4.setText(model.getValueAt(i, 6).toString());
        textField_2.setText(model.getValueAt(i, 7).toString());
        textField_5.setText(model.getValueAt(i, 8).toString());
        textField_3.setText(model.getValueAt(i, 9).toString());

    }

});

scrollPane.setBounds(324, 21, 873, 554);
contentPane.add(scrollPane);

table = new JTable();
model=new DefaultTableModel();
table.setDefaultEditor(Object.class, null);

Object column[]={"id","first name","last name","designation","dob","doj","salary","address","email-id","phone no"};

Object row[]=new Object[10];
model.setColumnIdentifiers(column);
table.setModel(model);

scrollPane.setViewportViewView(table);

```

```

JLabel lblNewLabel = new JLabel("First Name");
lblNewLabel.setBounds(32, 38, 77, 23);
contentPane.add(lblNewLabel);
textField = new JTextField();
textField.setBounds(159, 33, 124, 28);
contentPane.add(textField);
textField.setColumns(10);
JLabel lblLastName = new JLabel("Last Name");
lblLastName.setBounds(32, 78, 77, 23);
contentPane.add(lblLastName);
textField_1 = new JTextField();
textField_1.setBounds(159, 79, 124, 28);
contentPane.add(textField_1);
textField_1.setColumns(10);
lblJobTitle = new JLabel("Job Title");
lblJobTitle.setBounds(32, 138, 77, 23);
contentPane.add(lblJobTitle);
lblLastName_2 = new JLabel("Date of Birth");
lblLastName_2.setBounds(32, 185, 77, 23);
contentPane.add(lblLastName_2);
lblLastName_3 = new JLabel("Date of Joining");
lblLastName_3.setBounds(32, 219, 94, 23);
contentPane.add(lblLastName_3);
lblLastName_4 = new JLabel("Salary");
lblLastName_4.setBounds(32, 268, 77, 23);
contentPane.add(lblLastName_4);
lblLastName_5 = new JLabel("Address");
lblLastName_5.setBounds(32, 316, 77, 23);
contentPane.add(lblLastName_5);
lblLastName_6 = new JLabel("Mobile Number");
lblLastName_6.setBounds(32, 417, 89, 23);
contentPane.add(lblLastName_6);

```

```

comboBox= new JComboBox(jt);
comboBox.setBounds(159, 138, 124, 22);
contentPane.add(comboBox);
dateChooser = new JDateChooser();
dateChooser.setBounds(159, 180, 124, 28);
contentPane.add(dateChooser);
dateChooser_1 = new JDateChooser();
dateChooser_1.setBounds(159, 219, 124, 28);
contentPane.add(dateChooser_1);
textField_2 = new JTextField();
textField_2.setColumns(10);
textField_2.setBounds(159, 313, 124, 28);
contentPane.add(textField_2);
textField_3 = new JTextField();
textField_3.setColumns(10);
textField_3.setBounds(159, 414, 124, 28);
contentPane.add(textField_3);
JButton btnNewButton = new JButton("Save");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(textField.getText().trim().length()<2 ) {
            JOptionPane.showMessageDialog(frame, "Please fill the first name field");
        }
        else if(textField_1.getText().trim().length()<2) {
JOptionPane.showMessageDialog(frame, "Please fill the last name field");
        }
        else if(comboBox.getSelectedIndex()==0) {
            JOptionPane.showMessageDialog(frame, "Please select the role");
        }
    }
});
elseif(dateChooser.getCalendar()==null || dateChooser_1.getCalendar()==null) {
    JOptionPane.showMessageDialog(frame, "Please select the dates");
}
}

```



```

        else if(textField_4.getText().trim().length()<2) {
JOptionPane.showMessageDialog(frame, "Please fill the salary field");

        }else if(textField_2.getText().trim().length()<2) {
JOptionPane.showMessageDialog(frame, "Please fill the address field");
        }
        else if(textField_5.getText().trim().length()<2) {

            JOptionPane.showMessageDialog(frame, "Please enter valid mail");
        }
        else if(textField_5.getText().trim().length()<2) {

            boolean f=false;

            Pattern pattern=Pattern.compile(regex);

            Matcher matcher = pattern.matcher(textField_5.getText().trim());


            if(matcher.matches()? f=true:false) {

                if(f==false) {
JOptionPane.showMessageDialog(frame,"Emaild ID is invalid");

                }

            }

        }

        else if(textField_3.getText().trim().length()<2) {
JOptionPane.showMessageDialog(frame, "Please fill the phone number");
        }

        else if(textField_3.getText().trim().length()!=10) {
JOptionPane.showMessageDialog(frame, "Phone number must contain 10 digits");
        }

        else {

            SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-yyyy");

            String date = sdf.format(dateChooser.getDate().getTime());

            String date_1=sdf.format(dateChooser_1.getDate().getTime());

            java.util.Date dt_1=null;

            try {

                dt_1 = sdf.parse(date);

            } catch (ParseException e2) {

```

```

        e2.printStackTrace();
    }
    java.util.Date dt_2 = null;
    try {
        dt_2 = sdf.parse(date_1);
    } catch (ParseException e2) {
        e2.printStackTrace();
    }

    try {
        String query="insert into employee
values(seq_emp.nextval,?,?,?,?,?,?,?,?)";

        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
        PreparedStatement ps=con.prepareStatement(query);
        ps.setString(1,textField.getText());
        ps.setString(2,textField_1.getText());
        ps.setString(3,(String) comboBox.getSelectedItem());
        String g=textField_3.getText();
        ps.setDate(4,new java.sql.Date(dt_1.getTime()));
        ps.setDate(5,new java.sql.Date(dt_2.getTime()));
        ps.setInt(6,Integer.parseInt(textField_4.getText().trim()));
        ps.setString(7,textField_2.getText());
        ps.setString(8,textField_5.getText());
        ps.setString(9,textField_3.getText());

        int y=ps.executeUpdate();

        System.out.println(y);

        PreparedStatement ps1=con.prepareStatement("select eid
from employee where empphone_no=?");

        ps1.setString(1,g);
        ResultSet rs=ps1.executeQuery();
        rs.next();

```

```

        int sds=rs.getInt(1);
        row[0]=sds;
        row[1]=textField.getText();
        row[2]=textField_1.getText();
        row[3]=(String) comboBox.getSelectedItemAt();
        row[4]=new java.sql.Date(dt_1.getTime());
        row[5]=new java.sql.Date(dt_2.getTime());
        row[6]=textField_4.getText().trim();
        row[7]=textField_2.getText();
        row[8]=textField_5.getText();
        row[9]=textField_3.getText();
        model.addRow(row);
        textField.setText("");
        textField_1.setText("");
        textField_2.setText("");
        textField_3.setText("");
        textField_4.setText("");
        textField_5.setText("");
        dateChooser.setCalendar(null);
        dateChooser_1.setCalendar(null);
        comboBox.setSelectedIndex(0);
        ps.close();
        con.close();
    } catch (ClassNotFoundException e1) {
        e1.printStackTrace();
    } catch (SQLException e1) {
        e1.printStackTrace();
    }
    JOptionPane.showMessageDialog(frame, "Saved");
}
}

});

```

```

btnNewButton.setBounds(20, 463, 89, 23);

contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("Clear");

btnNewButton_1.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        textField.setText("");

        textField_1.setText("");

        textField_2.setText("");

        textField_3.setText("");

        textField_4.setText("");

        textField_5.setText("");

        dateChooser.setCalendar(null);

        dateChooser_1.setCalendar(null);

        comboBox.setSelectedIndex(0);

    }

});

btnNewButton_1.setBounds(20, 509, 89, 23);

contentPane.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("Delete");

btnNewButton_2.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        if(table.getSelectionModel().isSelectionEmpty()){

JOptionPane.showMessageDialog(null,"To delete click on view all button and then select a row to delete");

        }

        else{

            int i=table.getSelectedRow();

            if(i>=0) {

                int y=(int) table.getModel().getValueAt(0, 0);

                String query="delete from employee where eid=?";

                try {

                    Class.forName("oracle.jdbc.driver.OracleDriver");

                    Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");

```

```

        PreparedStatement ps=con.prepareStatement(query);
        ps.setInt(1, y);
        ps.execute();
        ps.close();
    }
    catch(SQLException|ClassNotFoundException e1) {
        e1.printStackTrace();
    }
    model.removeRow(i);
    JOptionPane.showMessageDialog(null, "successfully deleted");
}
else {
    JOptionPane.showMessageDialog(null, "Please select a row");
}
}

});
btnNewButton_2.setBounds(179, 509, 89, 23);
contentPane.add(btnNewButton_2);
JButton btnNewButton_2_1 = new JButton("Back");
btnNewButton_2_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
btnNewButton_2_1.setBounds(179, 552, 89, 23);
contentPane.add(btnNewButton_2_1);
JButton btnNewButton_1_1 = new JButton("Update");
btnNewButton_1_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        java.sql.Date date=null;
        if(table.getSelectionModel().isSelectionEmpty()){

```

```
JOptionPane.showMessageDialog(null,"To update click on view all button and then select a row to update");
```

```
    }
    else {
        java.sql.Date date1=null;
        int i=table.getSelectedRow();
        model.setValueAt(textField.getText(),i,1);
        model.setValueAt(textField_1.getText(),i,2);
        model.setValueAt((String) comboBox.getSelectedItem(),i,3);
        model.setValueAt(textField_4.getText(),i,6);
        model.setValueAt(textField_2.getText(),i,7);
        model.setValueAt(textField_5.getText(),i,8);
        model.setValueAt(textField_3.getText(),i,9);
        //System.out.println(i);
        //System.out.println(model.getValueAt(i,1));
        try {
            date=(Date) model.getValueAt(i,4);
            date1=(Date)model.getValueAt(i, 5);
            model.setValueAt(new java.sql.Date(date.getTime()), i, 4);
            model.setValueAt(new java.sql.Date(date1.getTime()), i, 5);
```

```
                String          query="update          employee          set
efname=?,elname=?,job_title=?,dob=?,doj=?,salary=?,address=?,email_id=?,empphone_no=?          where
empphone_no=?";
```

```
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
                PreparedStatement ps=con.prepareStatement(query);
                ps.setString(1,(String)model.getValueAt(i, 1));
                ps.setString(2,(String)model.getValueAt(i, 2));
                ps.setString(3,(String) comboBox.getSelectedItem());
                textField_3.getText();
                ps.setDate(4,new java.sql.Date(date.getTime()));
                ps.setDate(5,new java.sql.Date(date1.getTime()));
                ps.setInt(6,Integer.parseInt(textField_4.getText().trim()));
```

```

        ps.setString(7,textField_2.getText());
        ps.setString(8,(String)table.getModel().getValueAt(i, 7));
        ps.setString(9,textField_3.getText());
        ps.setString(10,textField_3.getText());
        int u=ps.executeUpdate();
        System.out.println(u);
    }
    catch(SQLException|ClassNotFoundException e1) {
        e1.printStackTrace();
    }
}
});

btnNewButton_1_1.setBounds(179, 463, 89, 23);
contentPane.add(btnNewButton_1_1);
textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBounds(159, 265, 124, 28);
contentPane.add(textField_4);
textField_5 = new JTextField();
textField_5.setColumns(10);
textField_5.setBounds(159, 366, 124, 28);
contentPane.add(textField_5);
JLabel lblLastName_6_1 = new JLabel("Email ID");
lblLastName_6_1.setBounds(32, 366, 77, 23);
contentPane.add(lblLastName_6_1);

btnNewButton_3 = new JButton("View all");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        model.setRowCount(0);
        String query="select * from employee";

```

```

try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
    PreparedStatement ps=con.prepareStatement(query);
    ResultSet rs1=ps.executeQuery();
    while(rs1.next()) {
        row[0]=rs1.getInt(1);
        row[1]=rs1.getString(2);
        row[2]=rs1.getString(3);
        row[3]=rs1.getString(4);
        row[4]=rs1.getDate(5);
        row[5]=rs1.getDate(6);
        row[6]=rs1.getInt(7);
        row[7]=rs1.getString(8);
        row[8]=rs1.getString(9);
        row[9]=rs1.getString(10);
        model.addRow(row);
    }
}
catch(SQLException|ClassNotFoundException e11) {
    e11.printStackTrace();}
});
btnNewButton_3.setBounds(20, 552, 89, 23);
contentPane.add(btnNewButton_3);
}
}

```

## User.java

```

package dbmsp;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

```



```

import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class user extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    public static void userscreen() {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    user frame = new user();
                    frame.setTitle("User Screen");
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public user() {
        dispose();
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton btnNewButton = new JButton("Login");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                login.adminloginscreen();
                dispose();
            }
        });
    }
}

```

```

        }

    });

    btnNewButton.setBounds(76, 74, 122, 35);
    contentPane.add(btnNewButton);

    JButton btnNewButton_1 = new JButton("Register");
    btnNewButton_1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            register.registerscreen();
            dispose();        }    });
    btnNewButton_1.setBounds(248, 74, 128, 35);
    contentPane.add(btnNewButton_1);
}
}

```

## **Usersc.java**

```

package dbmsp;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class usersc extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    public static int s=0;
    public static int us=0;
    public static int am=0;
    public static void screen() {
        EventQueue.invokeLater(new Runnable() {
            public void run() {

```

```

        try {
            usersc frame = new usersc();
            frame.setTitle(login.name);
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
}

public usersc() {
    dispose();
    setBounds(100, 100, 520, 337);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JButton btnNewButton = new JButton("Buy");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            items.itemscreen();
        }
    });
    btnNewButton.setBounds(162, 23, 133, 39);
    contentPane.add(btnNewButton);

    JButton btnNewButton_1 = new JButton("Previous Orders");
    btnNewButton_1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            prevord.screen();
        }
    });

```

```

});
btnNewButton_1.setBounds(162, 92, 133, 39);
contentPane.add(btnNewButton_1);
JButton btnNewButton_1_2 = new JButton("Logout");
btnNewButton_1_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
btnNewButton_1_2.setBounds(162, 226, 133, 39);
contentPane.add(btnNewButton_1_2);

JButton btnNewButton_1_2_1 = new JButton("Edit");
btnNewButton_1_2_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cedit.screen();
    }
});
btnNewButton_1_2_1.setBounds(162, 156, 133, 39);
contentPane.add(btnNewButton_1_2_1);
}
}

```

## **Vemp.java**

```

package dbmsp;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import java.awt.event.ActionListener;

```

```

import java.awt.event.ActionEvent;

public class vemp extends JFrame {

    private static final long serialVersionUID = 1L;

    private JPanel contentPane;

    public static void screen() {

        EventQueue.invokeLater(new Runnable() {

            public void run() {

                try {

                    vemp frame = new vemp();

                    frame.setTitle("Employee Info");

                    frame.setVisible(true);

                } catch (Exception e) {

                    e.printStackTrace(); } } });

    public vemp() {

        dispose();

        setBounds(100, 100, 450, 300);

        contentPane = new JPanel();

        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);

        contentPane.setLayout(null);

        JButton btnNewButton = new JButton("Add Employees");

        btnNewButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                emp.empscreen(); } });

        btnNewButton.setBounds(145, 58, 127, 38);

        contentPane.add(btnNewButton);

        JButton btnNewButton_1 = new JButton("Back");

        btnNewButton_1.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                dispose(); } });

        btnNewButton_1.setBounds(145, 131, 127, 38);

        contentPane.add(btnNewButton_1);
    }
}

```

```
} }
```

## **Prevord .java**

```
package dbmsp;

import java.awt.EventQueue;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JTable;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class prevord extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTable table;
    DefaultTableModel model,model1;
    private JTable table_1;
    public static void screen() {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    prevord frame = new prevord();
                    frame.setTitle(login.name);
```

```

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

});

}

public prevord() {
    dispose();
    setBounds(100, 100, 607, 383);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    model=new DefaultTableModel();
    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setBounds(27, 11, 531, 126);
    contentPane.add(scrollPane);
    table = new JTable();
    scrollPane.setViewportView(table);
    Object column[]= {"sno","name","price","quantity","total"};
    Object row[]=new Object[5];
    table.setModel(model);
    scrollPane.setViewportView(table);
    JButton btnNewButton = new JButton("Back");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dispose();
        }
    });
    model.setColumnIdentifiers(column);
    btnNewButton.setBounds(229, 294, 103, 39);
    contentPane.add(btnNewButton);
    JScrollPane scrollPane_1 = new JScrollPane();

```

```

scrollPane_1.setBounds(42, 167, 480, 87);
contentPane.add(scrollPane_1);
Object column1[]= {"date of purchase","time of purchase","total Price"};
Object row1[]=new Object[3];
table_1 = new JTable();
scrollPane_1.setViewportViewView(table_1);
model1=new DefaultTableModel();
table_1.setModel(model1);
scrollPane_1.setViewportViewView(table_1);
model1.setColumnIdentifiers(column1);
try {
    String query="select * from prevorder where customer_id=?";
    String query1="select * from sales where customer_id=?";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","tilakavati","tilak");
    PreparedStatement ps=con.prepareStatement(query);
    PreparedStatement ps1=con.prepareStatement(query1);
    ps.setInt(1, login.cid);
    ps1.setInt(1, login.cid);
    ResultSet rs=ps.executeQuery();
    ResultSet rs1=ps1.executeQuery();
    while(rs.next() ) {
        if(rs.getInt(2)==1) {
model.insertRow(table.getRowCount(),new Object[]{"","","","",""});
        }
        row[0]=rs.getInt(2);
        row[1]=rs.getString(3);
        row[2]=rs.getInt(4);
        row[3]=rs.getInt(5);
        row[4]=rs.getInt(6);
        model.addRow(row);
    }
}

```



```

        while(rs1.next()) {
            row1[0]=rs1.getDate(3);
            row1[1]=rs1.getString(5);
            row1[2]=String.valueOf(rs1.getInt(4));
            model1.addRow(row1);
        }
    }

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

```

## **DATABASE TABLES**

```
SQL> desc employee;
Name                                         Null?      Type
-----
EID                                         NOT NULL   NUMBER(38)
EFNAME                                     NOT NULL   VARCHAR2(255)
ELNAME                                     NOT NULL   VARCHAR2(255)
JOB_TITLE                                  NOT NULL   VARCHAR2(255)
DOB                                         NOT NULL   DATE
DOJ                                         NOT NULL   DATE
SALARY                                     NOT NULL   NUMBER(38)
ADDRESS                                    NOT NULL   VARCHAR2(255)
EMAIL_ID                                   NOT NULL   VARCHAR2(255)
EMPPHONE_NO                               NOT NULL   VARCHAR2(255)

SQL> _
```

```
SQL> desc customer;
Name                                         Null?      Type
-----
CUSTOMER_ID                               NOT NULL   NUMBER(38)
CFNAME                                     NOT NULL   VARCHAR2(255)
CLNAME                                     NOT NULL   VARCHAR2(255)
EMAIL_ID                                   NOT NULL   VARCHAR2(255)
GENDER                                    NOT NULL   VARCHAR2(25)
AGE                                         NOT NULL   NUMBER(38)
PHONE_NO                                   VARCHAR2(10)
ADDRESS                                    NOT NULL   VARCHAR2(100)

SQL> desc grlogin;
Name                                         Null?      Type
-----
USERNAME                                  NOT NULL   VARCHAR2(15)
PASSWORD                                  NOT NULL   VARCHAR2(15)
CUSTOMER_ID                               NUMBER(38)
```

```
SQL> desc item;
Name                                         Null?      Type
-----
ITEM_ID                                   NOT NULL   NUMBER(38)
ITEM_NAME                                  NOT NULL   VARCHAR2(255)

SQL> desc vegetable;
Name                                         Null?      Type
-----
VID                                         NOT NULL   NUMBER(38)
ITEM_ID                                   NOT NULL   NUMBER(38)
VEGETABLE_NAME                             VARCHAR2(255)
AVAILABILITY                                NOT NULL   NUMBER(38)
PRICE                                       NOT NULL   NUMBER(38)

SQL> desc water;
Name                                         Null?      Type
-----
WID                                         NOT NULL   NUMBER(38)
LIQUID_ID                                   NOT NULL   NUMBER(38)
BRAND                                       NOT NULL   VARCHAR2(255)

SQL>
```

```
SQL> desc waterap;
Name                                         Null?      Type
-----
WID                                         NOT NULL   NUMBER(38)
AVAILABILITY                               NOT NULL   NUMBER(38)
PRICE                                      NOT NULL   NUMBER(38)
QUANTITY                                   NOT NULL   NUMBER(38)

SQL> desc milk;
Name                                         Null?      Type
-----
MILK_ID                                    NOT NULL   NUMBER(38)
LIQUID_ID                                  NOT NULL   NUMBER(38)
BRAND                                       NOT NULL   VARCHAR2(255)

SQL> desc milkap;
Name                                         Null?      Type
-----
MILK_ID                                    NOT NULL   NUMBER(38)
AVAILABILITY                               NOT NULL   NUMBER(38)
PRICE                                      NOT NULL   NUMBER(38)
QUANTITY                                   NOT NULL   NUMBER(38)

SQL> desc beverages;
Name                                         Null?      Type
-----
BEVERAGES_ID                              NOT NULL   NUMBER(38)
LIQUID_ID                                  NOT NULL   NUMBER(38)
BRAND                                       NOT NULL   VARCHAR2(255)

SQL> desc beveragesap;
Name                                         Null?      Type
-----
BEVERAGES_ID                              NOT NULL   NUMBER(38)
AVAILABILITY                               NOT NULL   NUMBER(38)
PRICE                                      NOT NULL   NUMBER(38)
QUANTITY                                   NOT NULL   NUMBER(38)
```

```
SQL> desc solid;
Name                                         Null?      Type
-----
SOLID_ID                                   NOT NULL   NUMBER(38)
ITEM_ID                                    NOT NULL   NUMBER(38)
SNAME                                       NOT NULL   VARCHAR2(255)
QUANTITY                                   NOT NULL   NUMBER(38)
PRICE                                      NOT NULL   NUMBER(38)

SQL> desc fruit;
Name                                         Null?      Type
-----
FRUIT_ID                                   NOT NULL   NUMBER(38)
ITEM_ID                                    NOT NULL   NUMBER(38)
NAME                                        NOT NULL   VARCHAR2(255)
AVAILABILITY                               NOT NULL   NUMBER(38)
PRICE                                      NOT NULL   NUMBER(38)

SQL> desc seasonalfruit;
Name                                         Null?      Type
-----
SFRUIT_ID                                  NOT NULL   NUMBER(38)
ITEM_ID                                    NOT NULL   NUMBER(38)
NAME                                        NOT NULL   VARCHAR2(255)
AVAILABILITY                               NOT NULL   NUMBER(38)
PRICE                                      NOT NULL   NUMBER(38)
DOM                                         DATE

SQL> desc cereal;
Name                                         Null?      Type
-----
CEREAL_ID                                  NOT NULL   NUMBER(38)
ITEM_ID                                    NOT NULL   NUMBER(38)
NAME                                        NOT NULL   VARCHAR2(255)
AVAILABILITY                               NOT NULL   NUMBER(38)
PRICE                                      NOT NULL   NUMBER(38)
```

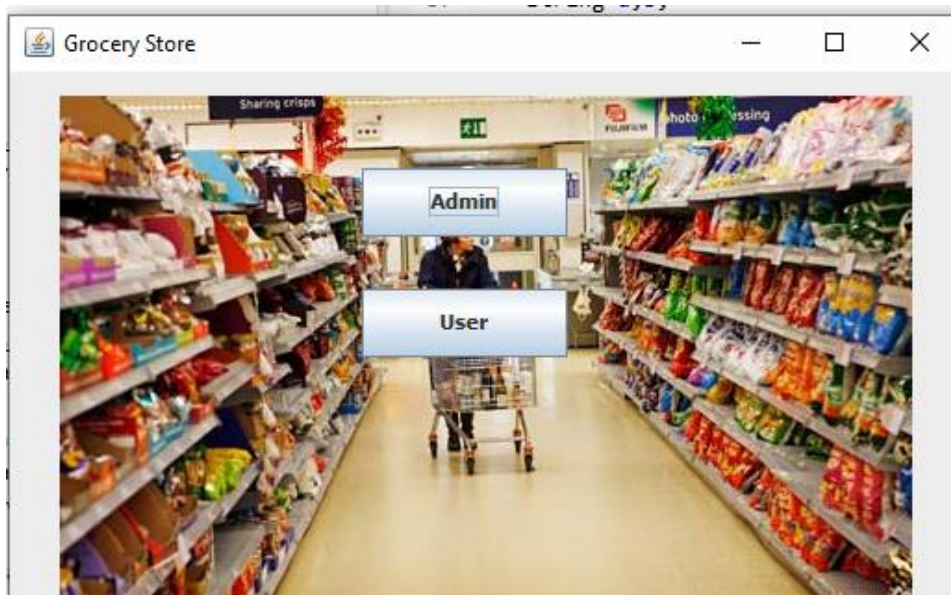
```
SQL> desc bill;
Name                                     Null?    Type
-----
ORDER_ID                               NOT NULL NUMBER(38)
CUSTOMER_ID                             NOT NULL NUMBER(38)
SALES_ID                                NOT NULL NUMBER(38)
TOTAL_BILL                              NOT NULL NUMBER(38)
DOP                                      NOT NULL DATE
MOT                                      NOT NULL VARCHAR2(255)
AMOUNT_PAYABLE                          NOT NULL NUMBER(38)

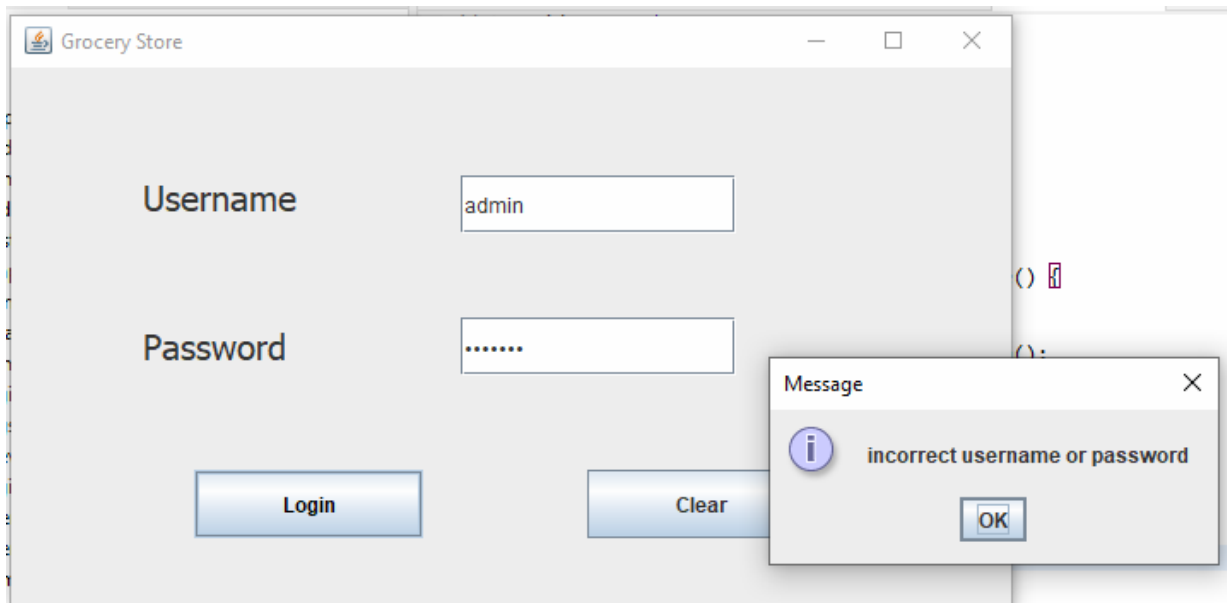
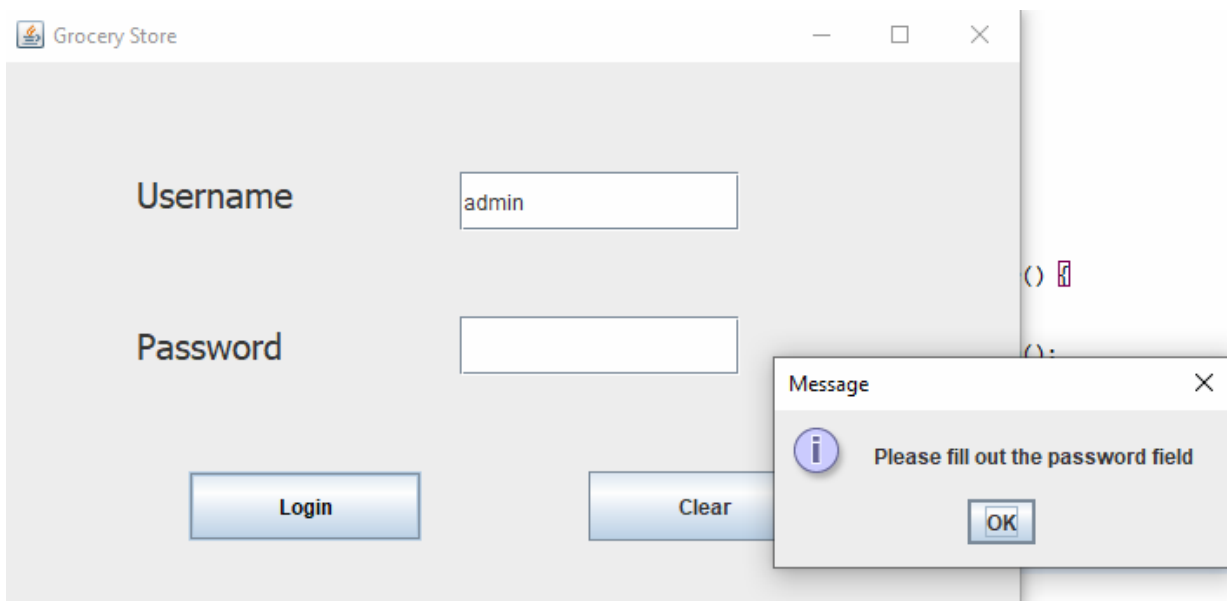
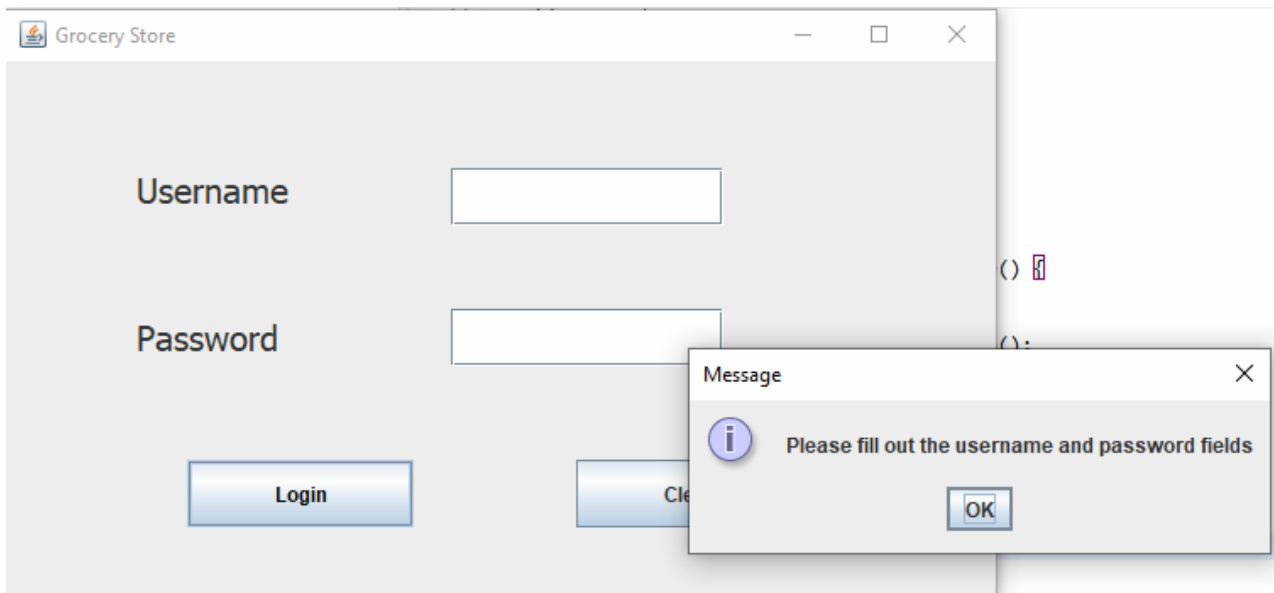
SQL> desc prevorder;
Name                                     Null?    Type
-----
CUSTOMER_ID                             NOT NULL NUMBER(38)
SNO                                      NOT NULL NUMBER(38)
NAME                                     NOT NULL VARCHAR2(25)
PRICE                                   NOT NULL NUMBER(38)
QUANTITY                                NOT NULL NUMBER(38)
TOTAL                                   NOT NULL NUMBER(38)

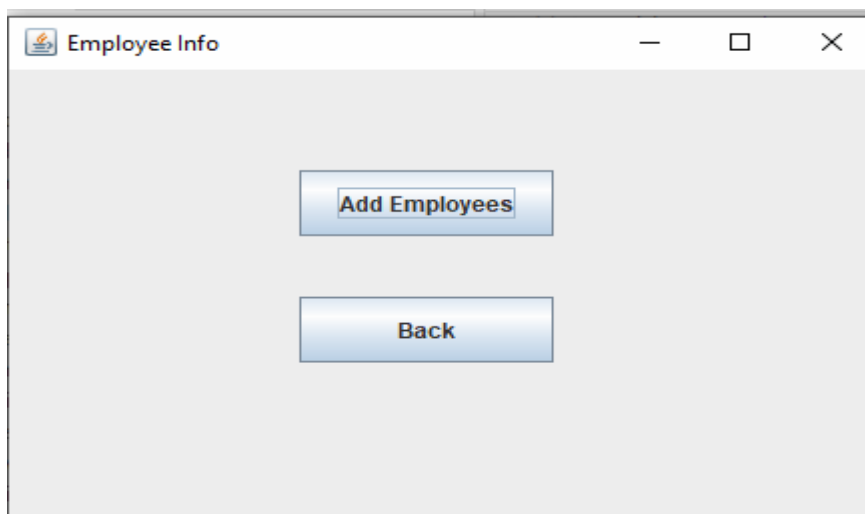
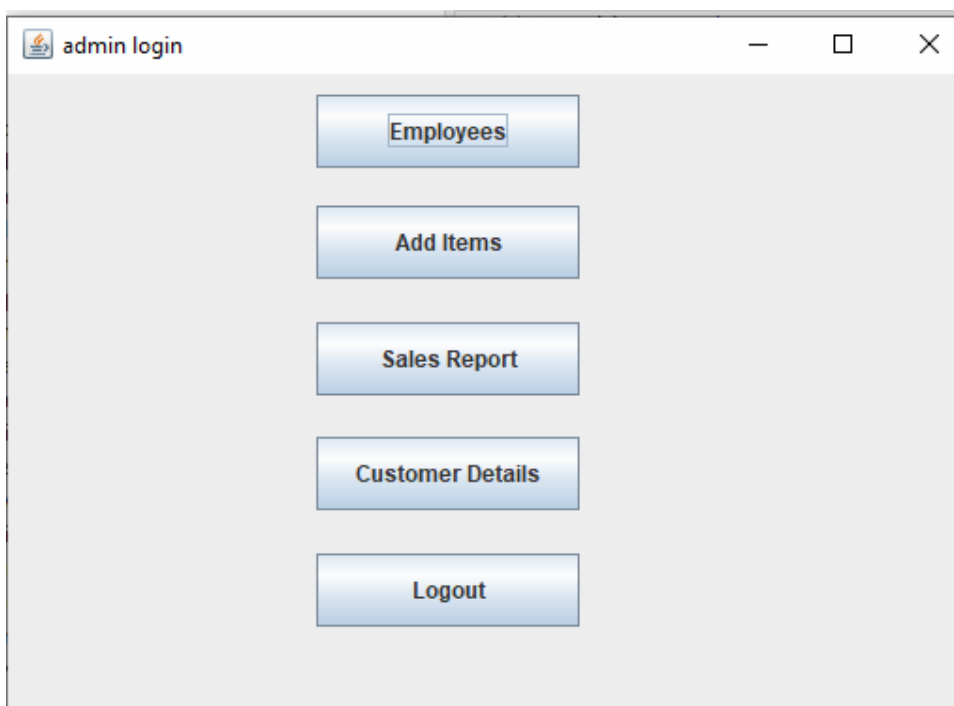
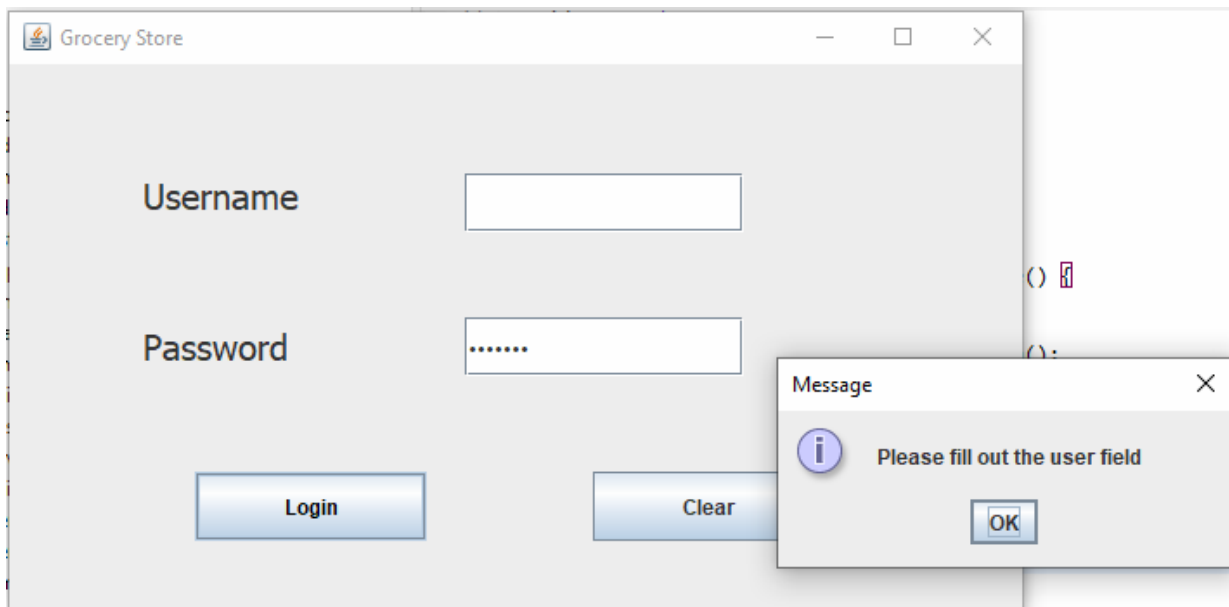
SQL> desc sales;
Name                                     Null?    Type
-----
SALES_ID                                NOT NULL NUMBER(38)
CUSTOMER_ID                             NOT NULL NUMBER(38)
DOP                                      NOT NULL DATE
FINAL_PRICE                             NOT NULL NUMBER(38)
TOP                                      NOT NULL VARCHAR2(20)
```

## SCREENSHOTS

### Admin:







Employees

First Name

Sally

Last Name

Jane

Job Title

Senior Analyst

Date of Birth

12-Apr-1975

Date of Joining

19-Jul-2011

Salary

75000

Address

New York

Email ID

llyjane21@gmail.com

Mobile Number

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
----	------------	-----------	-------------	-----	-----	--------	---------	----------	----------

Message

Please fill the phone number

OK

Activate Windows  
Go to Settings to activate Windows.

Employees

First Name

Sally

Last Name

Jane

Job Title

Senior Analyst

Date of Birth

12-Apr-1975

Date of Joining

19-Jul-2011

Salary

75000

Address

Email ID

llyjane21@gmail.com

Mobile Number

7849823774

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
----	------------	-----------	-------------	-----	-----	--------	---------	----------	----------

Message

Please fill the address field

OK

Activate Windows  
Go to Settings to activate Windows.

Employees

First Name

Sally

Last Name

Jane

Job Title

Senior Analyst

Date of Birth

12-Apr-1975

Date of Joining

19-Jul-2011

Salary

Address

New York

Email ID

llyjane21@gmail.com

Mobile Number

7849823774

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
<div> <div>Message</div> <div> <div>Please fill the salary field</div> <div>OK</div> </div> </div>									

Activate Windows

Go to Settings to activate Windows.

Employees

First Name

Sally

Last Name

Job Title

Senior Analyst

Date of Birth

12-Apr-1975

Date of Joining

19-Jul-2011

Salary

75000

Address

New York

Email ID

llyjane21@gmail.com

Mobile Number

7849823774

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
<div> <div>Message</div> <div> <div>Please fill the last name field</div> <div>OK</div> </div> </div>									

Activate Windows

Go to Settings to activate Windows.

Employees

First Name

Last Name

Jane

Job Title

Senior Analyst

Date of Birth

12-Apr-1975

Date of Joining

19-Jul-2011

Salary

75000

Address

New York

Email ID

llyjane21@gmail.com

Mobile Number

7849823774

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
<div> <div>Message</div> <div> <div>Please fill the first name field</div> <div>OK</div> </div> </div>									

Activate Windows

Go to Settings to activate Windows.



Employees

First Name

Sally

Last Name

Jane

Job Title

Senior Analyst

Date of Birth

12-Apr-1975

Date of Joining

19-Jul-2011

Salary

75000

Address

New York

Email ID

Mobile Number

7849823774

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
<div> <div>Message</div> <div> <div>Please enter valid email</div> <div>OK</div> </div> </div>									

Activate Windows

Go to Settings to activate Windows.

Employees

First Name

Last Name

Job Title

Manager

Date of Birth

Date of Joining

Salary

Address

Email ID

Mobile Number

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
10001	Sally	Jane	Senior Analyst	1975-04-12	2011-07-19	75000	New York	sallyjane21@...	7849823774

Message

Saved

OK

Activate Windows

Go to Settings to activate Windows.

Employees

First Name

Last Name

Job Title

Manager

Date of Birth

Date of Joining

Salary

Address

Email ID

Mobile Number

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
10001	Sally	Jane	Senior Analyst	1975-04-12	2011-07-19	75000	New York	sallyjane21@...	7849823774
10002	Murvin	Drake	Analyst	1984-06-14	2016-09-06	25000	California	murvinwer@...	9482398712
10003	Madison	Phillips	Analyst	1984-02-22	2016-12-07	25000	Berlin	madison3@g...	8874982384

Message

To update click on view all button and then select a row to update

OK

Activate Windows

Go to Settings to activate Windows.

Employees

First Name

Madison

Last Name

Phillips

Job Title

Analyst

Date of Birth

22-Feb-1984

Date of Joining

07-Dec-2016

Salary

25000

Address

Bogota

Email ID

madison3@gmail.com

Mobile Number

8874982384

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
10001	Sally	Jane	Senior Analyst	1975-04-12	2011-07-19	75000	New York	sallyjane21@...	7849823774
10002	Murvin	Drake	Analyst	1984-06-14	2016-09-06	25000	California	murvinwer@...	9482398712
10003	Madison	Phillips	Analyst	1984-02-22	2016-12-07	25000	Bogota	madison3@g...	8874982384

Activate Windows

Go to Settings to activate Windows.

Employees

First Name

Madison

Last Name

Phillips

Job Title

Analyst

Date of Birth

22-Feb-1984

Date of Joining

07-Dec-2016

Salary

25000

Address

Bogota

Email ID

madison3@gmail.com

Mobile Number

8874982384

Save

Update

Clear

Delete

View all

Back

id	first name	last name	designation	dob	doj	salary	address	email-id	phone no
10001	Sally	Jane	Senior Analyst	1975-04-12	2011-07-19	75000	New York	sallyjane21@...	7849823774
10002	Murvin	Drake	Analyst	1984-06-14	2016-09-06	25000	California	murvinwer@...	9482398712

Message

successfully deleted

OK

Activate Windows

Go to Settings to activate Windows.

admin login

Employees

Add Items

Sales Report

Customer Details

Logout

Items Info

Vegetables

1

Beans

?

Select an Option

Are you sure you want to save

Yes

No

Cancel

Name

brinjal

Save

Quantity

250

Clear

Price

240

Delete

Availability

Back

Select Category

Vegetables

Stock

Liquid T

ID	Item ID
	2
	2
	2

Items Info

Name

brinjal

Save

Quantity

250

Clear

Price

240

Availability

Select Category

Vegetables

Message

i

Saved

OK

Items Info

Name:

Quantity:

Price:

Availability:

Select Category:

Save

Select an Option

? Are you sure you want to save

Yes No Cancel

Back

Items Info

Name:

Quantity:

Price:

Availability:

Select Category:

Delete

Back

Select an Option

? Are you sure you want to save

Yes No Cancel

Items Info

Name:

Quantity:

Price:

Availability:

Select Category:

Save

Back

Select an Option

? Are you sure you want to save

Yes No Cancel

Items Info

Name


Quantity

Price

Availability

Select Category

Select an Option

 Are you sure you want to delete

Items Info

Name


Quantity

Price

Availability

Select Category

Select an Option

 Are you sure you want to delete

Items Info

Name


Quantity

Price

Availability

Select Category  ▼

Message

 Item name doesnt exist

Items Info

Name


Quantity

Price

Availability

Select Category  ▼

Message

 Deleted

Customer Details

Select to view the details of customer

[View Customer](#)

[View all customers](#)

[Back](#)

customer_id	first name	last name	email_id	gender	age	Phone no	address
-------------	------------	-----------	----------	--------	-----	----------	---------

Customer Details

Select to view the details of customer

[View Customer](#)

[View all customers](#)

[Back](#)

Input

?

Enter the customer id

3

[OK](#) [Cancel](#)

customer_id	first name	last name	email_id	gender	age	Phone no	address
-------------	------------	-----------	----------	--------	-----	----------	---------

Customer Details

Select to view the details of customer

[View Customer](#)

[View all customers](#)

[Back](#)

customer_id	first name	last name	email_id	gender	age	Phone no	address
3	antonio	clark	antoniectr@gmail.com	male	45	8980089800	New York

Customer Details

Select to view the details of customer

[View Customer](#)

[View all customers](#)

[Back](#)

customer_id	first name	last name	email_id	gender	age	Phone no	address
-------------	------------	-----------	----------	--------	-----	----------	---------

Input

Enter the customer id

4

[OK](#) [Cancel](#)

Customer Details

Select to view the details of customer

[View Customer](#)

[View all customers](#)

[Back](#)

customer_id	first name	last name	email_id	gender	age	Phone no	address
4	suzie	hawk	suzhawk@gmail.com	female	34	9989990902	Houston



Customer Details

Select to view the details of customer

View Customer

View all customers

Back

customer_id	first name	last name	email_id	gender	age	Phone no	address
4	suze	hawk	suzhawk@gmail.com	female	34	9989990902	Houston
3	antonio	clark	antonieclr@gmail.com	male	45	8980089800	New York
6	sherley	ashbaughn	shereyash@gmail.com	female	23	9979870093	Georgia

admin login

Employees

Add Items

Sales Report

Customer Details

Logout

Sales Info

Choose the date for sales,which you want to generate

☒ Date

12-Jul-2021

3905

Get date Back Clear

Sales Info

Choose the date for sales,which you want to generate

☒ Date

06-Jul-2021

0

Get date Back Clear

Sales Info

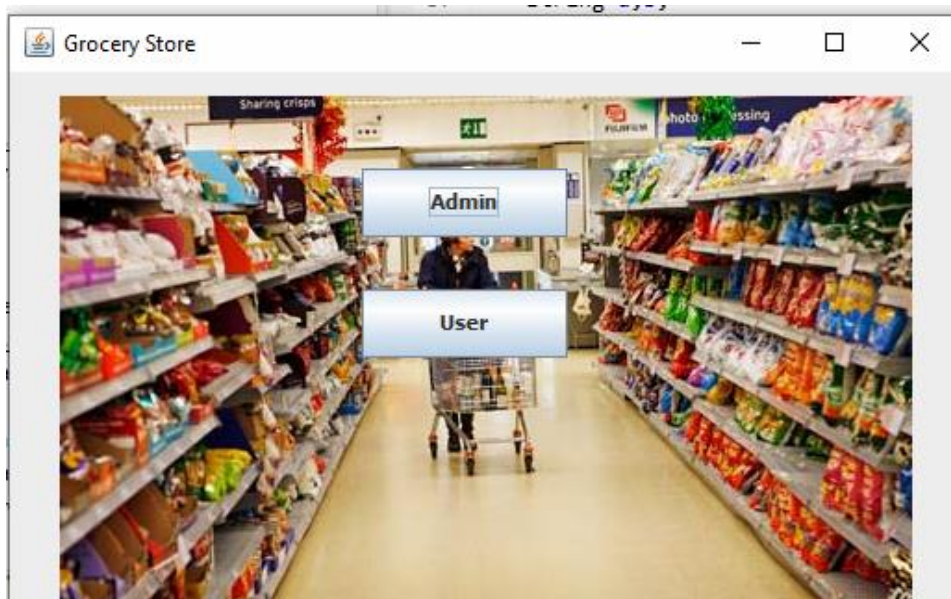
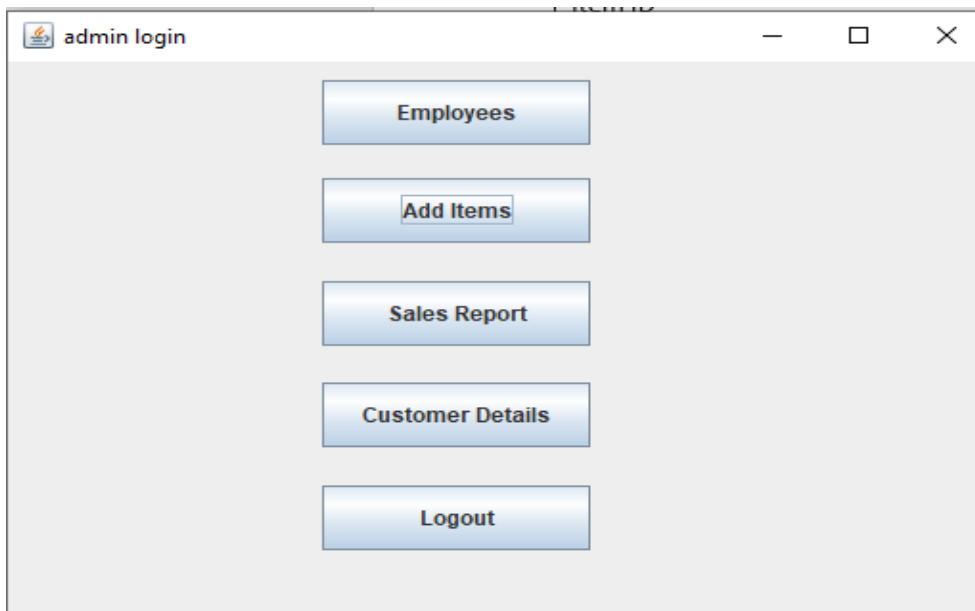
Choose the date for sales,which you want to generate

☒ Date

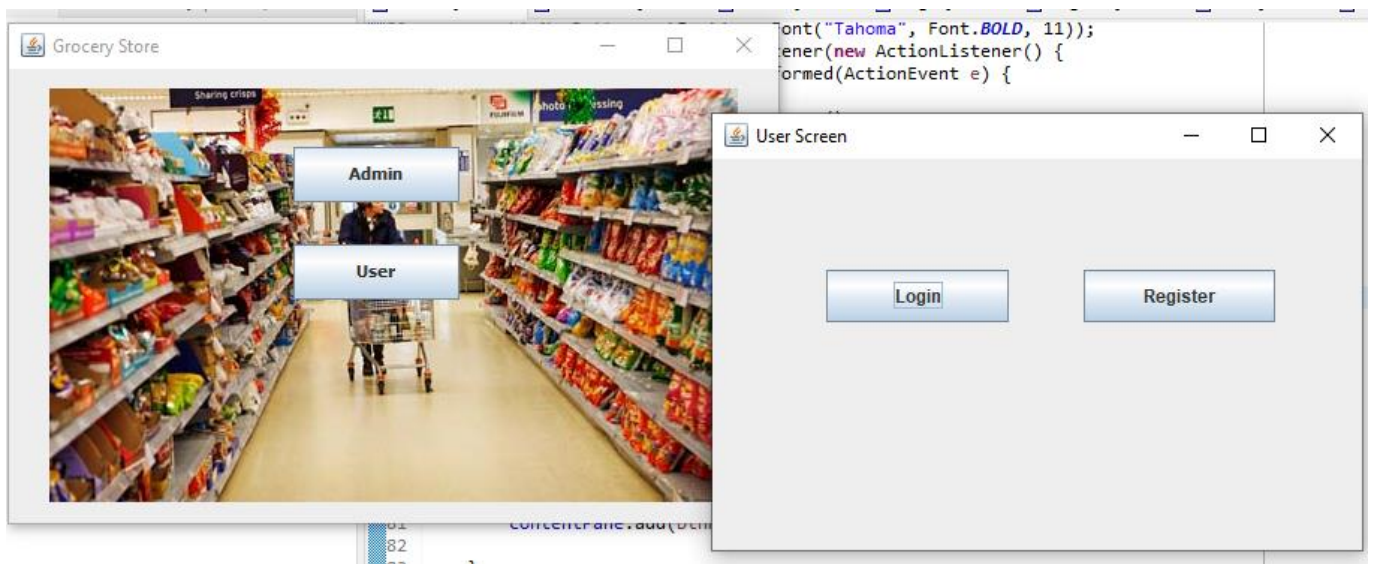
12-Jul-2021

3905

Get date Back Clear



**User**



The 'User Registration Form' window is a Java Swing dialog box with a light gray background. It features a title bar with the text 'User Registration Form'. The main content area is titled 'Register Form'. It contains several input fields and a set of radio buttons for gender selection. At the bottom, there are two buttons: 'Save' and 'Clear'.

First Name	<input type="text"/>
Last Name	<input type="text"/>
Email ID	<input type="text"/>
Gender	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Others
Address	<input type="text"/>
Age	<input type="text"/>
Phone Number	<input type="text"/>

antonio

First Name

Last Name

Email ID


Gender ☒ Male ☐

Address

Age

Phone Number

Message

 Successfully saved

Password Confirmation

UserName

Password

Confirm Password


Password Confirmation

UserName

Password

Confirm Password

Message

 passwords must be 8 character long


Password Confirmation

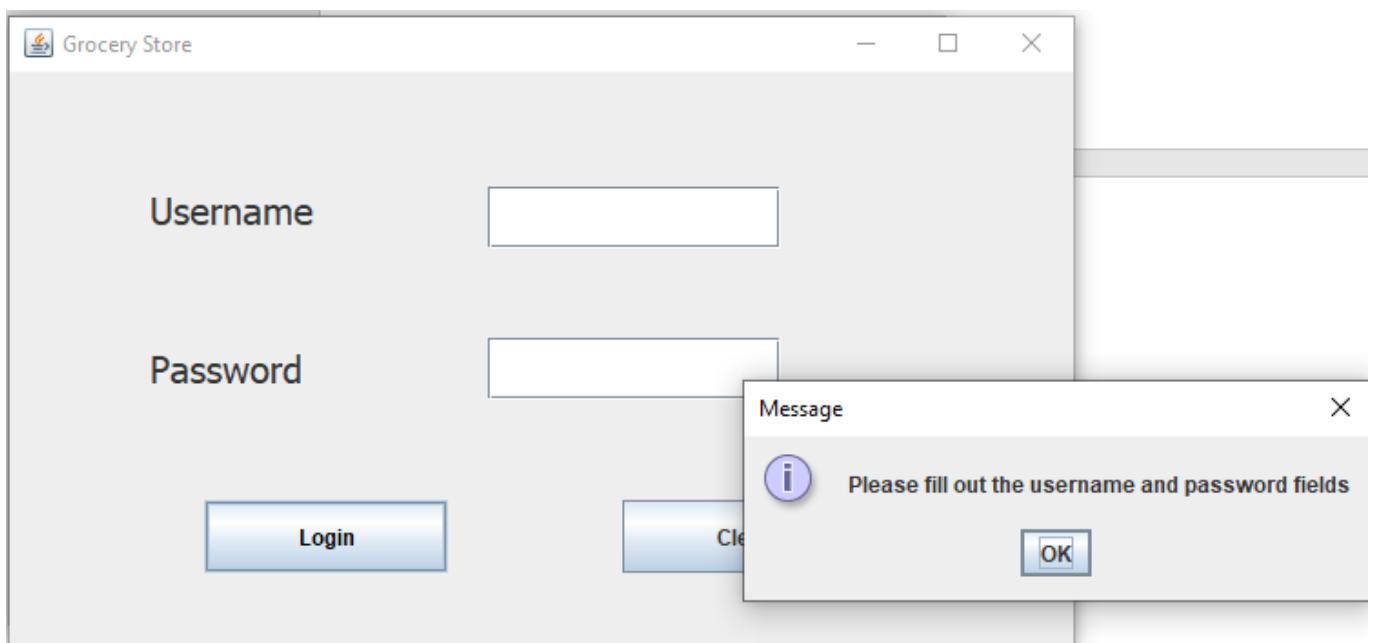
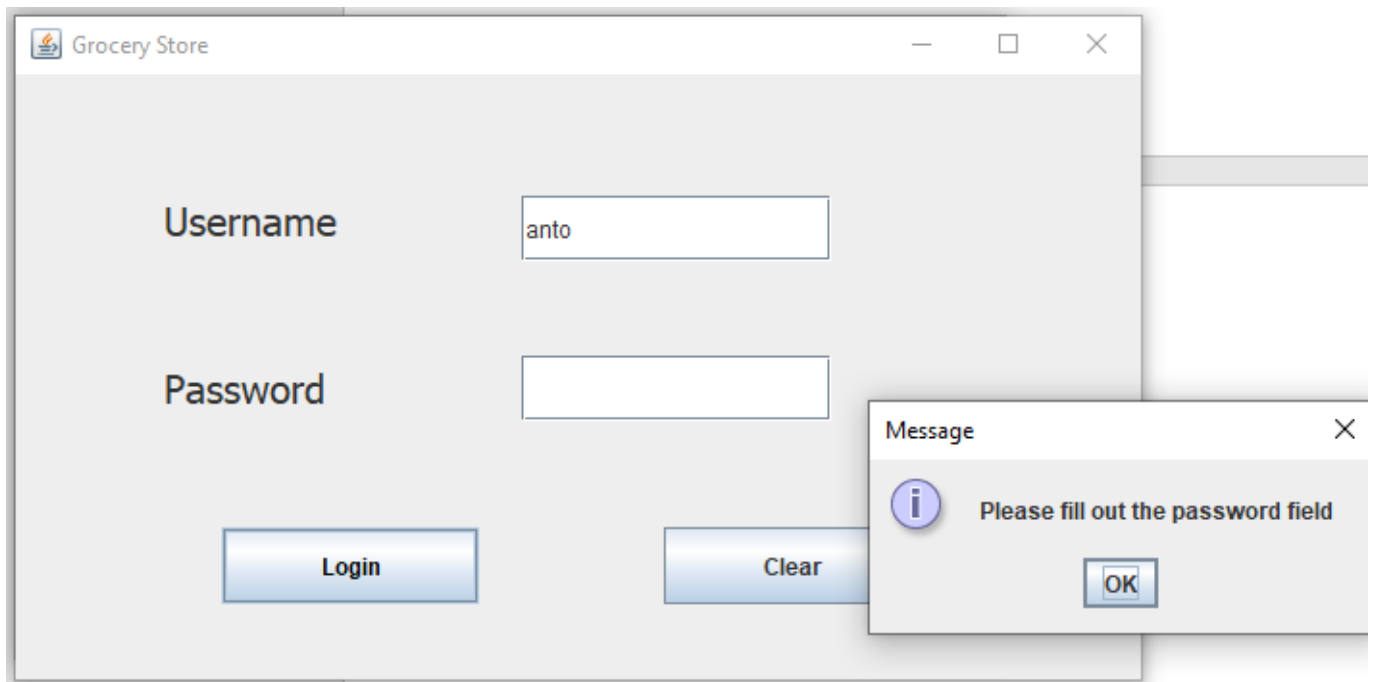
UserName

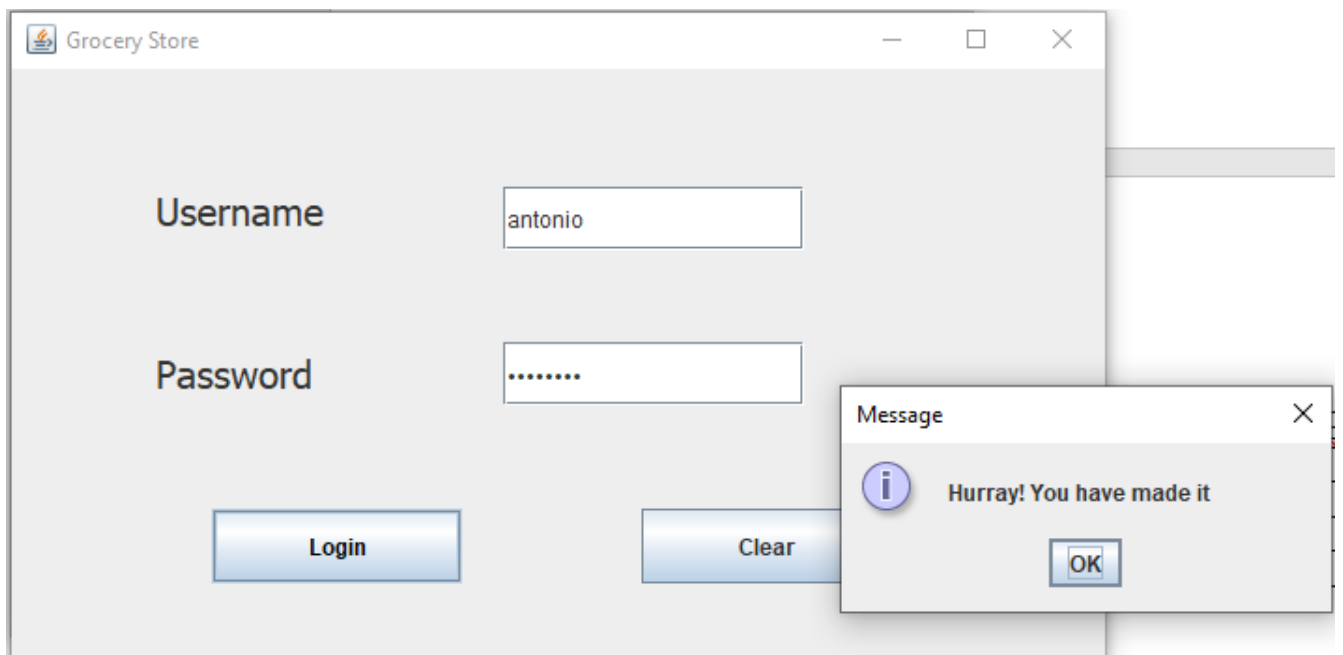
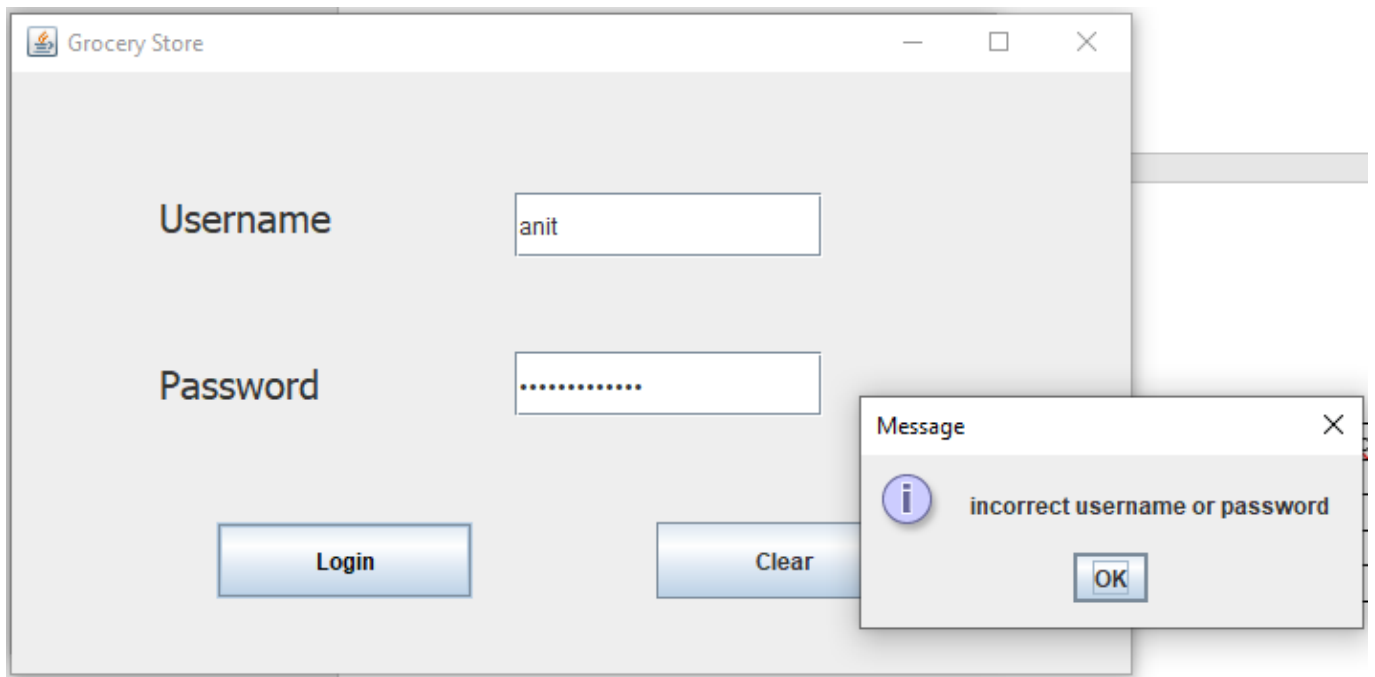
Password

Confirm Password


Message

 passwords must match








 antonio

Buy

Previous Orders

Edit

Logout

 antonio

Vegetables

Item

Tomato

Price

100

Quantity

3

Add To Bill

Clear

Back

Buy

vid	item_id	vegetable_name	availability	price
3007	1	Tomato	1	100
4007	1	Potato	1	100
5007	1	Bottle Guard	1	100
6007	1	Beans	1	100
1007	1	brinjal	250	240
2007	1	lady's finger	300	249

## Bill

id	Item	price	quantity	total
1	Tomato	100	3	300

antonio

Bill

Water

Item

Bisleri

Price

200

Quantity

Add To Bill

Clear

Back

Buy

id	Item	price	quantity	total
1	Tomato	100	3	300

Message

Please enter the quantity

OK

wid	liquid_id	brand	availability	price	quantity
1000	2	Oxygen	250	100	1
1000	2	Oxygen	300	400	5
2000	2	Bisleri	400	200	2
2000	2	Bisleri	450	150	1
3000	2	Kinley	200	180	1
3000	2	Kinley	300	450	2

antonio

Bill

Cereals

Item

Corn Flakes

Price

300

Quantity

2

Add To Bill

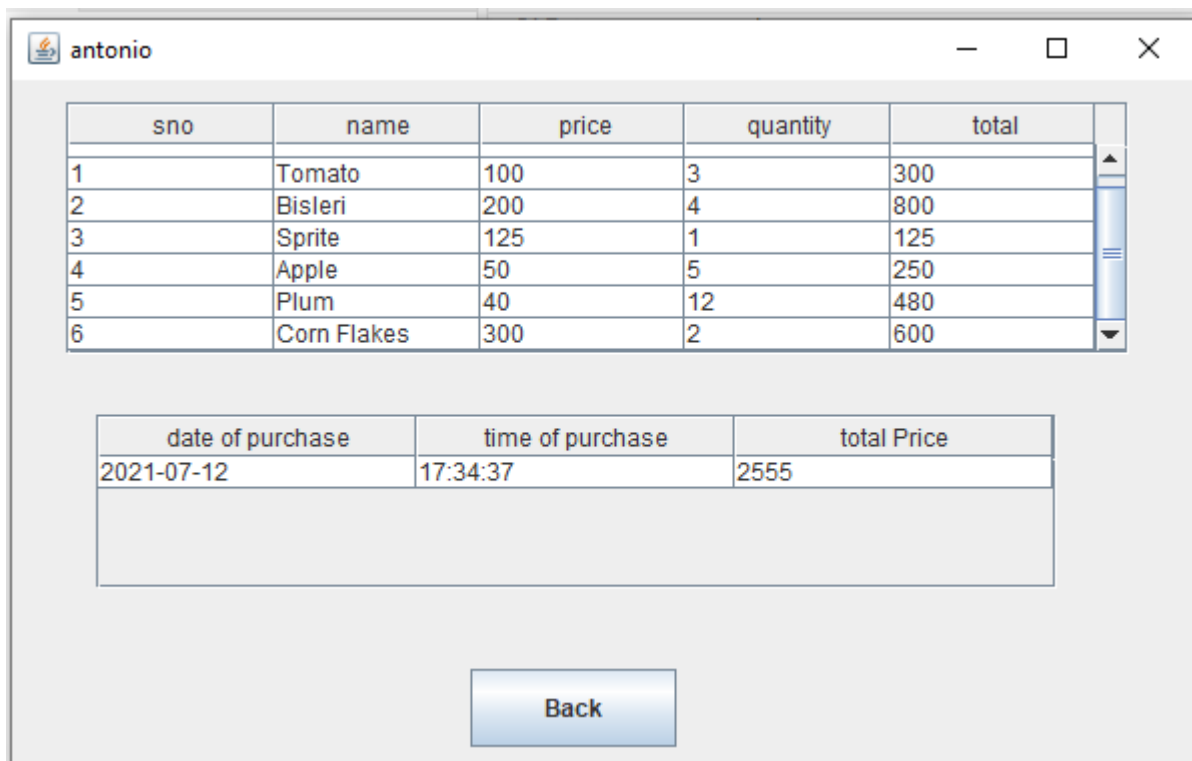
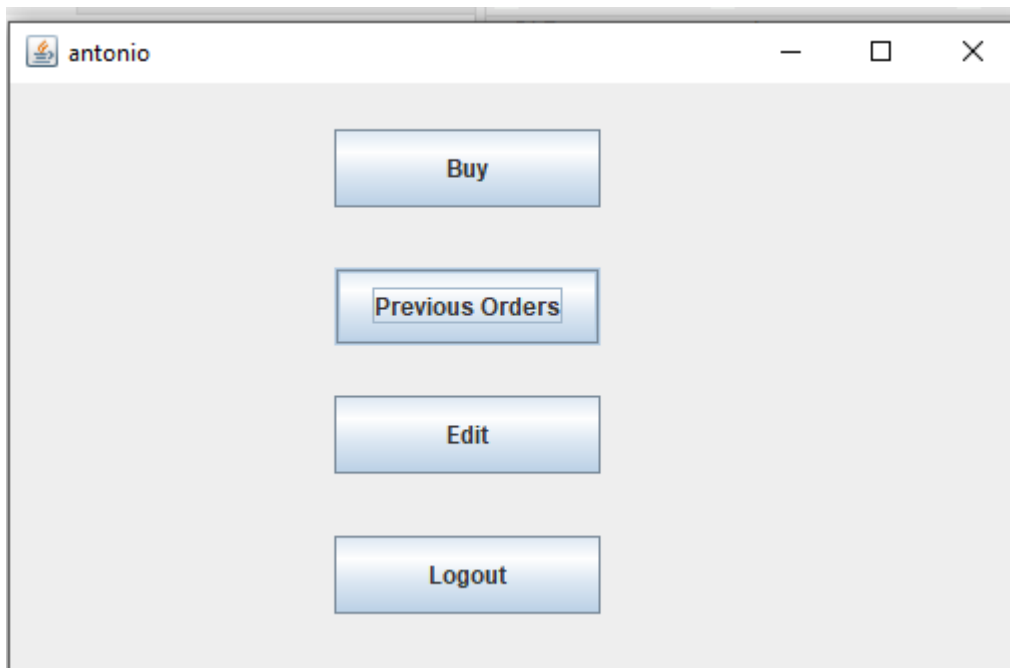
Clear

Back

Buy

id	Item	price	quantity	total
1	Tomato	100	3	300
2	Bisleri	200	4	800
3	Sprite	125	1	125
4	Apple	50	5	250
5	Plum	40	12	480
6	Corn Flakes	300	2	600

cereal_id	item_id	name	availability	price
8006	6	Oats	150	250
9006	6	Honey Rings	50	100
6006	6	Corn Flakes	500	300
7006	6	Chocos	750	200



## Register Form

First Name Last Name Email ID Gender ☐ OthersAddress Age Phone Number 

Message

×



Please fill the last name field

## Register Form

First Name Last Name Email ID Gender ☐ OthersAddress Age Phone Number 

Message

×



Please fill age field

User Registration Form

## Register Form

First Name

Last Name

Email ID

Gender ☐ Others

Address

Age

Phone Number

Message

Please fill the first name field

User Registration Form

## Register Form

First Name

Last Name

Email ID

Gender ☐ Others


Address

Age

Phone Number

Message

Please fill phone number field

 User Registration Form

Register Form

First Name

suzie

Last Name

hawk

Email ID

uzhawk@gmail.com

Gender

☐ Male

☒ Female

☐ Others

Address

Houston

Age


34

Phone Number

9989990902

Save

Clear

 Password Confirmation

UserName

Password

Confirm Password

Save

Clear

Back

suzie

sno	name	price	quantity	total

date of purchase	time of purchase	total Price

Back

suzie

Bill

Vegetables ▼

Item

Bottle Guard

Price

100

Quantity

2

id	Item	price	quantity	total
1	Bottle Guard	100	2	200

Add To Bill

Clear


Back

Buy

vid	item_id	vegetable_name	availability	price
3007	1	Tomato	1	100
4007	1	Potato	1	100
5007	1	Bottle Guard	1	100
6007	1	Beans	1	100
1007	1	brinjal	250	240
2007	1	lady's finger	300	249

[illegible]




suzie
— □ ×

Milk ▼

Item Jersey

Price 150

Quantity 12

Add To Bill

Clear


Back

Buy

### Bill

id	Item	price	quantity	total
1	Bottle Guard	100	2	200

milk_id	liquid_id	brand	availability	price	quantity
1001	7	Amul	230	100	250
1001	7	Amul	408	400	500
2001	7	Jersey	390	150	250
2001	7	Jersey	430	200	500
3001	7	Arogya	140	450	500
3001	7	Arogya	200	180	250


suzie
— □ ×

Milk ▼

Item Jersey

Price 150

Quantity 12

Add To Bill

Clear

Back

Buy

### Bill

id	Item	price	quantity	total
1	Bottle Guard	100	2	200
2	Jersey	150	12	1800

milk_id	liquid_id	brand	availability	price	quantity
1001	7	Amul	230	100	250
1001	7	Amul	408	400	500
2001	7	Jersey	390	150	250
2001	7	Jersey	430	200	500
3001	7	Arogya	140	450	500
3001	7	Arogya	200	180	250

## Bill

Cereals ▼

Item Honey Rings

Price 100

Quantity 3

id	Item	price	quantity	total
1	Bottle Guard	100	2	200
2	Jersey	150	12	1800
3	Honey Rings	100	3	300


Add To Bill

Clear

Back

Buy

cereal_id	item_id	name	availability	price
8006	6	Oats	150	250
9006	6	Honey Rings	50	100
6006	6	Corn Flakes	500	300
7006	6	Chocos	750	200


suzie
—
□
×

Cereals ▼

Item Honey Rings

Price 100

Quantity 3

Add To Bill

Clear

Back

Buy

### Bill

id	Item	price	quantity	total
1	Bottle Guard	100	2	200
2	Jersey	150	12	1800
3	Honey Rings	100	3	300

cereal_id	item_id	name	availability	price
8006	6	Oats	150	250
9006	6	Honey Rings	50	100
6006	6	Corn Flakes	500	300
7006	6	Chocos	750	200

## Bill

Item

Price

Quantity

id	Item	price	quantity	total
1	Bottle Guard	100	2	200
2	Jersey	150	12	1800
3	Honey Rings	100	3	300

Add To Bill

Clear

Back

Buy

cereal_id	item_id	name	availability	price
8006	6	Oats	150	250
9006	6	Honey Rings	50	100
6006	6	Corn Flakes	500	300
7006	6	Chocos	750	200

## Bill

Cereals ▼

Item

Corn Flakes

Price

300

Quantity

3

id	Item	price	quantity	total
1	Bottle Guard	100	2	200
2	Jersey	150	12	1800
3	Honey Rings	100	3	300
4	Corn Flakes	300	3	900


Add To Bill

Clear

Back

Buy

cereal_id	item_id	name	availability	price
8006	6	Oats	150	250
9006	6	Honey Rings	50	100
6006	6	Corn Flakes	500	300
7006	6	Chocos	750	200

 suzie

Cereals ▼

Item: Corn Flakes

Price: 300

Quantity: 3

Add To Bill

Clear


Back

Buy

### Bill

id	Item	price	quantity	total
1	Bottle Guard	100	2	200
2	Jersey	150	12	1800
3	Honey Rings	100	3	300
4	Corn Flakes	300	3	900

Message

 Your total is 3200

OK

cereal_id	item_id	name	availability	price
8006	6	Oats	150	250
9006	6	Honey Rings	50	100
6006	6	Corn Flakes	500	300
7006	6	Chocos	750	200

 suzie

sno	name	price	quantity	total
1	Bottle Guard	100	2	200
2	Jersey	150	12	1800
3	Honey Rings	100	3	300
4	Corn Flakes	300	3	900

date of purchase	time of purchase	total Price
2021-07-12	22:03:09	3200

Back

User Registration Form

## Register Form

First Name

Last Name

Email ID

Gender ☐ Male ☒ Female ☐ Others

Address

Age

Phone Number


Password Confirmation

UserName

Password

Confirm Password

Message

 passwords must be 8 character long

reyash@gmail.com	M
reyash@gmail.c	F
reyash@gmail.	F
reyash@gmail.c	F



— □ ×


Password Confirmation

UserName

Password

Confirm Password

Username already exists ×

 Error message

sherley

Buy

Previous Orders

Edit

Logout

sherley

First Name: sherley

Last Name: ashbaughn

Email ID: reyash@gmail.com

Gender: ☐ Male ☒ Female ☐ Others

Address: Atlanta

Age: 23

Phone Number: 9979870093

Save Back

First Name

Last Name

Email ID

Gender ☐ Male ☒ Female ☐ Others

Address

Age

Phone Number

First Name

Last Name

Email ID

Gender ☐ Male ☒ Female ☐ Others

Address

Age

Phone Number

sherley

First Name

Last Name

Email ID

Gender ☒ Male ☐ Others

Address

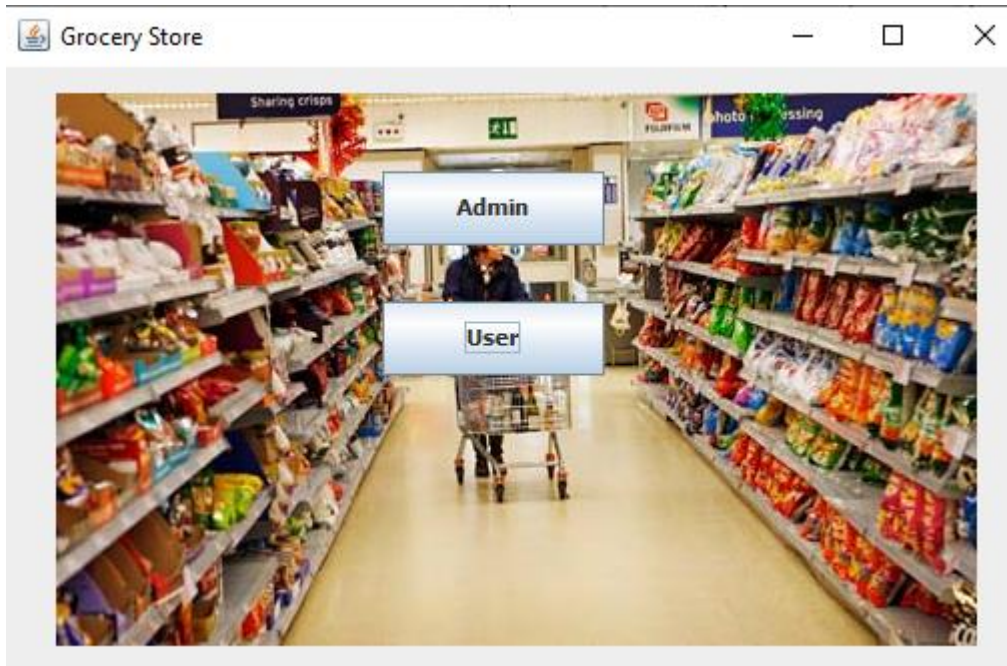
Age

Phone Number

Message

Successfully saved

sherley



## Testing

The code written for building GUI and connecting with database ensures that the values entered by the user are of correct data types. It prompts an error message if the values entered are not of the specified data types.

## RESULT

1. Connection with database is established
2. The values given for tables in the GUI components by the user are saved in the database.

## REFERENCES

<https://docs.oracle.com/javase/7/docs/api/>

<https://www.geeksforgeeks.org/establishing-jdbc-connection-in-java/>

<https://www.javatpoint.com/java-swing>

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.tutorialride.com](http://www.tutorialride.com)