

Name:-Vashesh Jogani

Roll no:-24

Div:-A

Batch:-AI&DS

**Airpod:**

```
package SwingMpr;
```

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import java.awt.event.*;
```

```
import java.awt.event.ActionListener;
```

```
public class Airpod extends JFrame implements  
ActionListener,ItemListener{
```

```
    JLabel Airpod_choice,Airpod_model,Airpod_price;
```

```
    JRadioButton c1,c3,c4,c5,c6;
```

```
    JList list;
```

```
    ButtonGroup cg,cg1,cg2,cg3;
```

```
    JButton b1,b2,b3,b4;
```

```
    String model;
```

```
    double base_airpod;
```

```
    boolean bool=false;
```

```
    Airpod()
```

```

{

    this.setSize(450,400);

    this.setLayout(null);
    this.getContentPane().setBackground(new Color(168, 169,
173));
    this.setTitle("Airpods");


    Airpod_model = new JLabel("Please choose a model?");
    Airpod_model.setBounds(25,50,250,20);
    Airpod_model.setFont(new
Font("Arial",Font.PLAIN,16));
    this.add(Airpod_model);


    cg = new ButtonGroup();
    c1= new JRadioButton("AirPods (3rd generation)");
    c1.setBounds(25,90,200,17);
    c1.setBackground(new Color(168, 169, 173));
    c1.addItemListener(this);
    this.add(c1);


    c3= new JRadioButton("AirPods Max");
    c3.setBounds(25,110,200,17);

```

```
c3.setBackground(new Color(168, 169, 173));  
c3.addItemListener(this);  
this.add(c3);
```

```
c4= new JRadioButton("AirPods Pro");  
c4.setBounds(25,130,200,17);  
c4.setBackground(new Color(168, 169, 173));  
c4.addItemListener(this);  
this.add(c4);
```

```
c5= new JRadioButton("AirPods (2nd generation)");  
c5.setBounds(25,150,200,17);  
c5.setBackground(new Color(168, 169, 173));  
c5.addItemListener(this);  
this.add(c5);
```

```
c6= new JRadioButton("AirPods (1st generation)");  
c6.setBounds(25,170,200,17);  
c6.setBackground(new Color(168, 169, 173));  
c6.addItemListener(this);  
this.add(c6);  
cg.add(c1);  
cg.add(c3);  
cg.add(c4);  
cg.add(c5);
```

```
cg.add(c6);
```

```
    b1= new JButton("DONE");  
    b1.setBackground(Color.white);  
    b1.setForeground(Color.DARK_GRAY);  
    b1.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
    b1.setBounds(160,220,80,40);  
    b1.addActionListener(this);  
    this.add(b1);
```

```
    b2= new JButton("CART");  
    b2.setBackground(Color.BLUE);  
    b2.setForeground(Color.WHITE);  
    b2.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
    b2.setBounds(60,220,80,40);  
    b2.addActionListener(this);  
    this.add(b2);
```

```
    b3= new JButton("CLEAR");  
    b3.setBackground(Color.GREEN);  
    b3.setForeground(Color.WHITE);
```

```
        b3.setFont(new
Font("Arial",Font.BOLD|Font.ITALIC,12));

        b3.setBounds(260,220,80,40);

        b3.addActionListener(this);

        this.add(b3);


        b4 = new JButton("Back to store");

        b4.setBounds(300,300,100,25);

        b4.setFont(new
Font("Arial",Font.PLAIN|Font.ITALIC,10));

        b4.setForeground(Color.red);

        b4.addActionListener(this);

        this.add(b4);


        this.revalidate();

        this.repaint();

        this.setVisible(true);

    }
```

@Override

```
public void actionPerformed (ActionEvent e){

    if(e.getSource()==b1)

    {
```

```

        if(bool==true)
        {
            Airpod_choice.setText("Selected CHOICE:
"+model);

            Airpod_price.setText("\nPrice: "+base_airpod);
            System.out.println("Selected CHOICE"+" "+"
"+model+" "+" "+" "+" ");

            getcart ob = new getcart();
            new Cartdummy(base_airpod,"add");
            ob.set_Airpod(model,base_airpod);

        }
        else if(model==null){
            Airpod_choice = new JLabel("Please Select a
choice");

            Airpod_choice.setBounds(50, 280, 400,40);
            Airpod_choice.setFont(new
Font("Arial",Font.PLAIN,16));

            this.add(Airpod_choice);
            Airpod_price = new JLabel();
            Airpod_price.setBounds(100, 320, 250,40);
            Airpod_price.setFont(new
Font("Arial",Font.PLAIN,16));

            this.add(Airpod_price);
            model=null;

```

```

        base_airpod=0.0;
        bool=true;
    }

    else
    {
        getcart ob = new getcart();
        new Cartdummy(base_airpod,"add");
        ob.set_Airpod(model,base_airpod);
        Airpod_choice = new JLabel("Selected
CHOICE: "+model);
        Airpod_choice.setBounds(50, 280, 400,40);
        Airpod_choice.setFont(new
Font("Arial",Font.PLAIN,16));
        this.add(Airpod_choice);
        Airpod_price = new JLabel("\nPrice:
"+base_airpod);
        Airpod_price.setBounds(100, 320, 250,40);
        Airpod_price.setFont(new
Font("Arial",Font.PLAIN,16));
        this.add(Airpod_price);
        model=null;
        base_airpod=0.0;
        bool=true;
    }

```

```
        this.revalidate();  
        this.repaint();  
        this.setVisible(true);  
    }
```

```
    if(e.getSource()==b2)  
    {  
        this.dispose();  
        getcart oc= new getcart();  
        oc.create_Frame(getcart.id, getcart.ap, getcart.mc,  
getcart.ip, getcart.idcost, getcart.ipcost, getcart.mccost,  
getcart.apcost);  
  
    }
```

```
    if(e.getSource()==b3)  
    {  
        this.dispose();  
        new Airpod();  
    }
```

```
    if(e.getSource()==b4)  
    {  
        this.dispose();  
        new Store();  
    }
```



```
}
```

```
}
```

```
@Override
```

```
public void itemStateChanged(ItemEvent e) {
```

```
    if(this.c1.isSelected()==true)
```

```
    {
```

```
        model="AirPods (3rd generation)";
```

```
        base_airpod=18500;
```

```
    }
```

```
    if(this.c3.isSelected()==true)
```

```
    {
```

```
        model="AirPods Max";
```

```
        base_airpod=59900;
```

```
    }
```

```
    if(this.c4.isSelected()==true)
```

```
    {
```

```
        model="AirPods Pro";
```

```
        base_airpod=24900;
```

```
}

if(c5.isSelected()==true)
{

    model="AirPods (2nd generation)";
    base_airpod=12900;

}

if(c6.isSelected()==true)
{
    model="AirPods (1st generation)";
    base_airpod=8000;

}

}
```

```
public static void main(String[] args)
{
    new Airpod();
}
```

```
}
```

### **Apple Pay:**

```
package SwingMpr;
```

```
import java.awt.event.*;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLIntegrityConstraintViolationException;
```

```
import java.util.Vector;
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
public class Apple_Pay extends JFrame implements  
ActionListener{
```

```

        JLabel AP_Title, AP_UserId,AP_AuthPin, AP_Add1,
        AP_Add2, AP_Add3,N_pass, AP_OrderCnf,lbl_error;

        JTextField tf1, tf4,tf5,tf6;

        JPasswordField tf2, tf3;

        JButton b1,b2,b3;

        public static boolean fundcheck;

        static public double deduct=-1.0;

        static String finalorder=" ";

        static Vector <String> v = new Vector<String>(16);

        Apple_Pay(double cost)
        {
            this.setSize(600,800);

            this.setLayout(null);

            this.getContentPane().setBackground(new Color(0, 71,
171));

            this.setTitle("Newark Store");

            AP_Title = new JLabel("NEWARK PAY");

            AP_Title.setBounds(150, 80, 250,40);

            AP_Title.setForeground(new Color(255, 215, 0));

            AP_Title.setFont(new
Font("Monospaced",Font.BOLD,40));

            this.add(AP_Title);


            /*list = new List(3,false);

            list.add("M1");

```

```
list.add("Intel i7");  
list.add("Intel i9");  
list.setBounds(25,100,250,20);  
list.addActionListener(this);  
this.add(list);*/
```

```
AP_UserId = new JLabel(" Newark_Id:",  
SwingConstants.LEFT);  
AP_UserId.setBounds(10,175,250,20);  
AP_UserId.setForeground(new Color(255, 215, 0));  
AP_UserId.setFont(new  
Font("Monospaced",Font.BOLD | Font.ITALIC,26));  
this.add(AP_UserId);
```

```
tf1 = new JTextField();  
tf1.setFont(new  
Font("TimesRoman",Font.BOLD,24));  
tf1.setBounds(275,170,250,35);  
this.add(tf1);
```

```
/*cg = new CheckboxGroup();  
c1= new Checkbox("Air",cg,false);
```

```
c1.setBounds(25,170,50,12);  
c1.addItemListener(this);  
this.add(c1);*/  
  
/*c3= new Checkbox("Pro",cg,false);  
c3.setBounds(25,210,50,12);  
c3.addItemListener(this);  
this.add(c3);*/
```

```
N_pass=new JLabel("Newark_Password:",  
SwingConstants.LEFT);  
  
N_pass.setBounds(10,320,250,20);  
  
N_pass.setFont(new Font("Monospaced",Font.BOLD |  
Font.ITALIC,23));  
  
N_pass.setForeground(new Color(255, 225, 0));  
this.add(N_pass);
```

```
tf2 = new JPasswordField();  
tf2.setBounds(275,315,250,35);  
tf2.setFont(new  
Font("TimesRoman",Font.BOLD,24));  
this.add(tf2);
```

```
        AP_AuthPin=new JLabel("Newark_Pin:",
SwingConstants.LEFT);

        AP_AuthPin.setBounds(10,250,250,20);

        AP_AuthPin.setFont(new
Font("Monospaced",Font.BOLD | Font.ITALIC,26));

        AP_AuthPin.setForeground(new Color(255, 215,
0));

        this.add(AP_AuthPin);
```

```
        tf3 = new JPasswordField();

        tf3.setBounds(275,245,250,35);

        tf3.setFont(new Font("TimesRoman",Font.BOLD,24));

        this.add(tf3);
```

```
        lbl_error=new JLabel();

        lbl_error.setBounds(10,400,550,20);

        lbl_error.setFont(new
Font("Monospaced",Font.BOLD | Font.ITALIC,26));
```

```
lbl_error.setForeground(new Color(255, 215, 0));  
this.add(lbl_error);
```

```
/*AP_Add1=new JLabel("Address 1 :");  
AP_Add1.setBounds(10,320,150,20);  
AP_Add1.setFont(new Font("Monospaced",Font.BOLD |  
Font.ITALIC,26));  
AP_Add1.setForeground(new Color(255, 215, 0));  
this.add(AP_Add1);
```

```
tf3 = new JTextField();  
tf3.setBounds(175,315,250,35);  
tf3.setFont(new Font("TimesRoman",Font.BOLD,24));  
this.add(tf3);
```

```
AP_Add2=new JLabel("Address 2 :");  
AP_Add2.setBounds(10,390,150,20);  
AP_Add2.setFont(new Font("Monospaced",Font.BOLD |  
Font.ITALIC,26));  
AP_Add2.setForeground(new Color(255, 215, 0));  
this.add(AP_Add2);
```



```
tf4 = new JTextField();  
tf4.setBounds(175,385,250,35);  
tf4.setFont(new  
Font("TimesRoman",Font.BOLD,24));  
this.add(tf4);
```

```
AP_Add3=new JLabel("Address 3 :");  
AP_Add3.setBounds(10,460,150,20);  
AP_Add3.setFont(new Font("Monospaced",Font.BOLD |  
Font.ITALIC,26));  
AP_Add3.setForeground(new Color(255, 215, 0));  
this.add(AP_Add3);
```

```
tf5 = new JTextField();  
tf5.setBounds(175,455,250,35);  
tf5.setFont(new  
Font("TimesRoman",Font.BOLD,24));  
this.add(tf5);
```

```
AP_Add4=new JLabel("Address 4 :");  
AP_Add4.setBounds(10,530,150,20);
```

```
AP_Add4.setFont(new Font("Monospaced",Font.BOLD |  
Font.ITALIC,26));
```

```
AP_Add4.setForeground(new Color(255, 215, 0));
```

```
this.add(AP_Add4);
```

```
tf6 = new JTextField();
```

```
tf6.setBounds(175,525,250,35);
```

```
tf6.setFont(new  
Font("TimesRoman",Font.BOLD,24));
```

```
this.add(tf6);
```

```
*/
```

```
/*cg2 = new CheckboxGroup();
```

```
c5= new Checkbox("128GB",cg2,false);
```

```
c5.setBounds(25,290,50,12);
```

```
c5.addItemListener(this);
```

```
this.add(c5);
```

```
c6= new Checkbox("256GB",cg2,false);
```

```
c6.setBounds(25,320,50,12);
```

```
c6.addItemListener(this);
```

```
this.add(c6);
```

```
c7= new Checkbox("512GB",cg2,false);
```

```
c7.setBounds(25,350,50,12);
```

```
c7.addItemListener(this);  
this.add(c7);  
c8= new Checkbox("1 TB",cg2,false);  
c8.setBounds(25,380,50,12);  
c8.addItemListener(this);  
this.add(c8);*/
```

```
/*MB_color=new JLabel("Choose a Color ");  
MB_color.setBounds(25,410, 250,20);  
MB_color.setFont(new  
Font("Arial",Font.PLAIN,16));  
this.add(MB_color);  
cg3 = new CheckboxGroup();  
c9= new Checkbox("SILVER",cg3,false);  
c9.setBounds(25,440,80,12);  
c9.addItemListener(this);  
this.add(c9);  
c10 = new Checkbox("ROSE-GOLD",cg3,false);  
c10.setBounds(25,470,120,12);  
c10.addItemListener(this);  
this.add(c10);  
c11= new Checkbox("SPACE GREY",cg3,false);
```

```
c11.setBounds(25,500,120,12);  
c11.addItemListener(this);  
this.add(c11);  
//c12= new Checkbox("SPACE GREY",cg3,false);  
//c12.setBounds(25,530,80,12);  
//c12.addItemListener(this);  
//this.add(c12);*/
```

```
b1= new JButton("PAY "+cost);  
b1.setBackground(new Color(50, 205, 50));  
b1.setForeground(Color.WHITE);  
b1.setFont(new  
Font("TimesRoman",Font.BOLD,20));  
b1.setBounds(80,620,350,40);  
b1.setOpaque(true);  
b1.addActionListener(this);  
this.add(b1);
```

```
b3=new JButton("Back to Cart");  
b3.setForeground(Color.white);  
b3.setBackground(new Color(0, 71, 171));
```

```
        b3.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
        b3.setBounds(100,520,180,25);  
        b3.addActionListener(this);  
        b3.setOpaque(true);  
        this.add(b3);
```

```
        this.revalidate();  
        this.repaint();  
        this.setVisible(true);
```

```
    }
```

```
@Override
```

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

```
    /*if(e.getSource()==list)
```

```
    {
```

```
        edition=list.getSelectedIndex();
```

```
        c2= new Checkbox(edition,cg,false);  
        c2.setBounds(25,190,100,12);  
        c2.addItemListener(this);  
        this.add(c2);  
    }*/
```

```
        if(e.getSource()==b1)  
        {  
            if (!tf1.getText().equals("") &&  
!String.valueOf(tf2.getPassword()).equals("") &&  
!String.valueOf(tf3.getPassword()).equals(""))  
            {  
                String UID = tf1.getText();  
                String UPASS =  
String.valueOf(tf2.getPassword());  
                String UPIN =  
String.valueOf(tf3.getPassword());
```

```
                String id="";  
                String d="";  
                String pin="";  
                FirstJDBC ob = new FirstJDBC();
```

```

Connection con = null;
PreparedStatement p = null;
ResultSet rs = null;

con = ob.connectDB();
try
{
    String sql = "select * from
appleid_database where Apple_ID='"+UID+"'";
    System.out.println(sql);
    p = con.prepareStatement(sql);
    rs= p.executeQuery();

    if(rs.next())
    {
        d =
rs.getString("Apple_password");
        System.out.println(d);
        pin = rs.getString("Newark_pin");
        System.out.println(pin);
        id = rs.getString("Apple_ID");
    }
}

```

```

        }

    }

    catch(Exception sql)
    {

        sql.printStackTrace();

//
        JOptionPane.showMessageDialog(this, "Recheck your
credential");

    }


    if(d.equals(UPASS) && pin.equals(UPIN))
    {

        System.out.println("Done");
        getBalance();
        if(Apple_Pay.deduct== -1.0)
        {

```



```

        b1.setVisible(false);
    }
    else
    {
        orderlist();
        addorder();
        pay_deduct(Login_Page.maild);

JOptionPane.showMessageDialog(this, "Payment sucessful");
        this.dispose();
        new Thank_you();
//        System.out.println("Payment
sucessful");

    }

}
else
{
    if(id.equals(""))
    {

JOptionPane.showMessageDialog(this, "No Account");

```

```
lbl_error.setText("NO ACCOUNT
!");

this.revalidate();
this.repaint();
this.setVisible(true);
}
else
{
System.out.println(id);
JOptionPane.showMessageDialog(this,
"Incorrect Password");
lbl_error.setText("INCORRECT
PASSWORD !");

this.revalidate();
this.repaint();
this.setVisible(true);
}
}

}

else {

JOptionPane.showMessageDialog(this,
"Please fill all fields");

lbl_error.setText("Please fill all fields");
```

```

        this.revalidate();
        this.repaint();
        this.setVisible(true);
    }
}
if(e.getSource()==b3)
{
    this.dispose();
    getcart oc= new getcart();
    oc.create_Frame(getcart.id, getcart.ap,
getcart.mc, getcart.ip, getcart.idcost, getcart.ipcost,
getcart.mccost, getcart.apcost);
}

}

```

```

/*public void itemStateChanged(ItemEvent e) {

```

```
if(c1.getState()==true)
```

```
{
```

```
    model="AIR";
```

```
    base_macbook=70000;
```

```
}
```

```
if(c2.getState()==true)
```

```
{
```

```
    model="13";
```

```
    base_macbook=90000;
```

```
}
```

```
if(c3.getState()==true)
```

```
{
```

```
    model="Pro";
```

```
    base_macbook=100000;
```

```
}
```

```
if(c1.getState()==false && c2.getState()==false &&  
c3.getState()==false)
```

```
{
```

```
}  
if(c5.getState()==true)  
{  
  
    memory="128GB";  
    memory_macbook=0;  
  
}  
if(c6.getState()==true)  
{  
    memory="256GB";  
    memory_macbook=5000;  
  
}  
if(c7.getState()==true)  
{  
    memory="512GB";  
    memory_macbook=10000;  
  
}  
if(c8.getState()==true)  
{
```

```
memory="1 TB";
memory_macbook=20000;
}

if(c9.getState()==true)
{

    Mac_color="SILVER";

}

if(c10.getState()==true)
{
    Mac_color="ROSE-GOLD";

}

if(c11.getState()==true)
{
    Mac_color="SPACE GREY";

}

//if(c12.getState()==true)
//{
//    Mac_color="BLUE";
```

```
//}
```

```
price=base_macbook+memory_macbook;
```

```
}
```

```
*/
```

```
public static void pay_deduct(String ID)
```

```
{
```

```
    FirstJDBC obc = new FirstJDBC();
```

```
    Connection con =obc.connectDB();
```

```
    PreparedStatement p = null;
```

```
    String q ="UPDATE  bank_database SET Balance=?  
where email=?";
```

```
    //ResultSet rs = null;
```

```
    try
```

```
    {
```

```
        p = con.prepareStatement(q);
```

```
p.setDouble(1,Apple_Pay.deduct);  
p.setString(2,ID);
```

```
p.executeUpdate();  
}  
catch (SQLIntegrityConstraintViolationException o)  
{  
    Sign_Up l = new Sign_Up() ;  
    l.errorofrepeat();  
  
    // TODO: handle exception  
}  
catch(Exception sql)  
{  
    sql.printStackTrace();  
}
```

```
public void getBalance()  
{  
    double balance=0.0;  
    FirstJDBC ob = new FirstJDBC();
```



```
Connection con = null;  
PreparedStatement p = null;  
ResultSet rs = null;
```

```
    con = ob.connectDB();  
    try  
    {  
        String sql = "select * from bank_database where  
email='"+Login_Page.maild+"'";  
        System.out.println(sql);  
        p = con.prepareStatement(sql);  
        rs= p.executeQuery();  
  
        if(rs.next())  
        {    balance = rs.getDouble("Balance");  
        System.out.println(balance);  
  
        }  
    }
```

```
}  
  
    catch(Exception sql)  
    {  
        sql.printStackTrace();  
    }  
  
if(balance<Cartdummy.gtotal)  
{  
    lbl_error.setText("Insuffiecient funds");  
    fundcheck=false;  
  
}  
else  
{  
  
    Apple_Pay.deduct=balance-Cartdummy.gtotal;  
    System.out.println(Apple_Pay.deduct);
```

```

        fundcheck= true;
    }
}

public static void orderlist()
{

    for(int i = 0;i<4;i++)
    {
        if(getcart.ipcost[i]!=0.0)
        {
            v.add(getcart.ip[i]+" "+getcart.ipcost[i]);
        }
    }
    for(int i=0; i<4 ;i++)
    {
        if(getcart.idcost[i]!=0.0)
        { v.add(getcart.id[i]+"
"+getcart.idcost[i]);}
    }
    for(int i=0; i<4 ;i++)
    {
        if(getcart.apcost[i]!=0.0)
        { v.add(getcart.ap[i]+"
"+getcart.apcost[i]);}
    }
}

```

```

        for(int i=0; i<4 ;i++)
        {
            if(getcart.mccost[i]!=0.0)
                { v.add(getcart.mc[i]+"
"+getcart.mccost[i]);}
        }

```

```

        finalorder = v.toString();
        System.out.print(v.toString());

```

```

    }

```

```

public static void addorder()
{

```

```

    FirstJDBC obc = new FirstJDBC();
    Connection con =obc.connectDB();

```

```

    PreparedStatement p = null;
    String q ="INSERT INTO order_database "
        + "VALUES(?,?,?,?)";

```

```

    //ResultSet rs = null;

```

```

    try

```

```
{  
  
    p = con.prepareStatement(q);  
    Emails u = new Emails();  
    p.setString(1,Login_Page.maild);  
        p.setString(2,finalorder);  
        p.setString(3,Integer.toString(u.generate()));  
        p.setDouble(4,Cartdummy.gtotal);  
  
    p.executeUpdate();  
}
```

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

```
public static void main(String[] args)  
{  
    new Apple_Pay(52.0);  
}
```

```
}
```

### **Cartdummy:**

```
package SwingMpr;  
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.*;  
import java.util.Arrays;  
import javax.swing.*;  
  
public class Cartdummy extends JFrame implements  
ActionListener{  
  
    JLabel[] ipad_items = new JLabel[5];
```

```
JLabel[] iphone_items = new JLabel[5];
```

```
JLabel[] macbook_items = new JLabel[5];
```

```
JLabel[] airpod = new JLabel[5];
```

```
JButton[] deletion = new JButton[20];
```

```
JButton proceed;
```

```
JLabel error;
```

```
int main_counter =0;
```

```
double [] cart = new double[20];
```

```
public static double gtotal=0.0;
```

```
JButton b1;
```

```
JButton b2;

JLabel total,display;

//TextField tf1;

    Cartdummy(String[]a, String[]b, String[]c,String[]d,
double[]ipad_cost,
double[]iphone_cost,double[]macbook_cost,double[]airpod_c
ost)
    {

        this.setVisible(true);


        this.setSize(1500,1500);
        this.setLayout(null);
        this.setTitle("Cart");
        int x=50;
        int y=0;
        error = new JLabel("Cant select more than 4 of each
items");
        error.setVisible(false);
        error.setBounds(25, 520, 100, 20);
        this.add(error);


        /*tf1 = new TextField(2);
        tf1.setBounds(1010,50,25,20);
```



```
this.add(tf1);*/
```

```
/* b2 = new JButton("REMOVE");
```

```
b2.setBounds(920,50,80,20);
```

```
this.add(b2);
```

```
b2.addActionListener(this);*/
```

```
y=y+30;
```

```
JLabel header0 = new JLabel("iPads in your cart");
```

```
header0.setFont(new Font("Arial",Font.BOLD,12));
```

```
header0.setBounds(25,y,780,20);
```

```
this.add(header0);
```

```
for(int i=0;i<4;i++,x=x+50)
```

```
{
```

```
    y=x;
```

```
    //try
```

```
    //{
```

```
        ipad_items[i]= new JLabel();
```

```
        ipad_items[i].setBounds(25,x, 780, 20);
```

```
ipad_items[i].setBackground(Color.LIGHT_GRAY);
```

```
System.out.println(i+" "+x);
```

```
        ipad_items[i].setFont(new  
Font("Arial",Font.BOLD,12));
```

```
        deletion[i]= new JButton("DELETE");  
        deletion[i].setFont(new  
Font("Arial",Font.PLAIN,12));  
        deletion[i].setBounds(830,x,100,18);  
        deletion[i].addActionListener(this);  
        this.add(deletion[i]);
```

```
        if(ipad_cost[i]!=0.00)  
        {  
            ipad_items[i].setText(i+" "+a[i]+"  
"+"Price"+" "+ "Rs "+ipad_cost[i]);  
        }  
        else  
        {    deletion[i].setVisible(false);  
            ipad_items[i].setText(null);  
            ipad_items[i].setVisible(false);
```

```

    }

    this.add(ipad_items[i]);

//}

//catch(ArrayIndexOutOfBoundsException e)
//{

    error.setVisible(false);

//}

}

JLabel header1 = new JLabel("iPhones in your
cart");

header1.setFont(new Font("Arial",Font.BOLD,12));
header1.setBounds(25,y+30,780,20);
this.add(header1);


for(int i=0;i<4;i++,x=x+50)
{
    y=x;

    System.out.println("hey");

    System.out.println(getcart.mc[1]);

```

```
        iphone_items[i]= new JLabel();
        iphone_items[i].setBounds(25,x, 780, 20);

iphone_items[i].setBackground(Color.LIGHT_GRAY);
        deletion[i+4]= new JButton("DELETE");
        deletion[i+4].setFont(new
Font("Arial",Font.PLAIN,12));
        deletion[i+4].setBounds(830,x,100,18);
        deletion[i+4].addActionListener(this);
        this.add(deletion[i+4]);
        if(iphone_cost[i]!=0.00)
        {
            iphone_items[i].setText((i+4)+" "+"
+d[i]+" "+"Price"+" "+"Rs "+" "+iphone_cost[i]);
        }
        else
        {
            deletion[i+4].setVisible(false);
            iphone_items[i].setText(null);
            iphone_items[i].setVisible(false);
        }

        iphone_items[i].setFont(new
Font("Arial",Font.BOLD,12));
        System.out.println(i+" "+x);
```

```

        this.add(iphone_items[i]);
    }
    JLabel header2 = new JLabel("MacBooks in your
cart");
    header2.setFont(new Font("Arial",Font.BOLD,12));
    header2.setBounds(25,y+30,780,20);
    this.add(header2);
    for(int i=0;i<4;i++,x=x+50)
    {
        y=x;
        System.out.println(getcart.mc[0]);
        macbook_items[i]= new JLabel();
        macbook_items[i].setBounds(25,x,780, 20);

        macbook_items[i].setBackground(Color.LIGHT_GRAY)
;
        macbook_items[i].setFont(new
Font("Arial",Font.BOLD,12));
        deletion[i+8]= new JButton("DELETE");
        deletion[i+8].setFont(new
Font("Arial",Font.PLAIN,12));
        deletion[i+8].setBounds(830,x,100,18);
        deletion[i+8].addActionListener(this);
        this.add(deletion[i+8]);
    }
}

```

```

        System.out.println(macbook_cost[i]);
        if(macbook_cost[i]!=0.00)
        {
            macbook_items[i].setText((i+8)+")"+" "+
"+c[i]+"Price"+ " "+ "Rs "+ " "+macbook_cost[i]);
        }
        else
        {
            deletion[i+8].setVisible(false);
            macbook_items[i].setText(null);
            macbook_items[i].setVisible(false);
        }
        this.add(macbook_items[i]);
    }
    JLabel header3 = new JLabel("Airpods in your
cart");
    header3.setFont(new Font("Arial",Font.BOLD,12));
    header3.setBounds(25,y+30,780,20);
    this.add(header3);
    for(int i=0;i<4;i++,x=x+50)
    {
        airpod[i]= new JLabel();
        airpod[i].setBounds(25,x,780, 20);

        airpod[i].setBackground(Color.LIGHT_GRAY);
    }
}

```

```

        airpod[i].setFont(new
Font("Arial",Font.BOLD,12));
        System.out.println(b[i]);
        deletion[i+12]= new JButton("DELETE");
        deletion[i+12].setFont(new
Font("Arial",Font.PLAIN,12));
        deletion[i+12].setBounds(830,x,100,18);
        deletion[i+12].addActionListener(this);
        this.add(deletion[i+12]);
        if(airpod_cost[i]!=0.00)
        {
            airpod[i].setText((i+12)+")"+" "+b[i]+"
"+"Price"+" " +"Rs " +" "+airpod_cost[i]);
        }
        else
        {
            deletion[i+12].setVisible(false);
            airpod[i].setText(null);
            airpod[i].setVisible(false);
        }
        this.add(airpod[i]);
    }

    b1 = new JButton("BACK TO Store");
    b1.setBounds(940,560,130,20);
    b1.addActionListener(this);

```

```
this.add(b1);
```

```
JLabel ORDERHEADING= new JLabel("ORDER  
DETAILS");
```

```
ORDERHEADING.setFont(new  
Font("serif",Font.BOLD,30));
```

```
ORDERHEADING.setBounds(940,25,250,40);
```

```
this.add(ORDERHEADING);
```

```
total= new JLabel("Order Total: "+gtotal);
```

```
total.setBounds(960,70, 160, 20);
```

```
this.add(total);
```

```
proceed = new JButton("PROCEED TO PAY");
```

```
proceed.setFont(new Font("serif",Font.BOLD,25));
```

```
proceed.setBackground(Color.GREEN);
```

```
proceed.setForeground(Color.WHITE);
```

```
proceed.setBounds(980,120,280, 40);
```

```
proceed.addActionListener(this);
```



```
this.add(proceed);  
this.revalidate();  
this.repaint();  
this.setVisible(true);
```

```
}  
Cartdummy(double t,String s)  
{  
    if(s.equals("add"))  
    {    gtotal=gtotal+t;}  
    else if (s.equals("remove"))  
    {    if(gtotal!=0)  
        gtotal=gtotal-t;  
    else  
    {  
        System.out.println("Nothing to be removed");  
    }  
}
```

```
}
```

```
@Override
```

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

```
    if(e.getSource()==b1)
```

```
    {
```

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

```
        new Store();
```

```
    }
```

```
    /*if(e.getSource()==b2)
```

```
    {
```

```
        getcart ob = new getcart();
```

```
        ob.delete(Integer.parseInt(tf1.getText()));
```

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

```
    }*/  
    if(e.getSource()==deletion[0])  
    {  
        getcart ob = new getcart();  
        ob.delete(0);  
        this.dispose();  
  
    }  
    if(e.getSource()==deletion[1])  
    {  
        getcart ob = new getcart();  
        ob.delete(1);  
        this.dispose();  
  
    }  
    if(e.getSource()==deletion[2])  
    {  
        getcart ob = new getcart();  
        ob.delete(2);  
        this.dispose();  
  
    }
```

```
if(e.getSource()==deletion[3])
{
    getcart ob = new getcart();
    ob.delete(3);
    this.dispose();

}
```

```
if(e.getSource()==deletion[4])
{
    getcart ob = new getcart();
    ob.delete(4);
    this.dispose();

}
```

```
if(e.getSource()==deletion[5])
{
    getcart ob = new getcart();
    ob.delete(5);
    this.dispose();

}
```

```
if(e.getSource()==deletion[6])
```

```
{  
    getcart ob = new getcart();  
    ob.delete(6);  
    this.dispose();  
  
}  
if(e.getSource()==deletion[7])  
{  
    getcart ob = new getcart();  
    ob.delete(7);  
    this.dispose();  
  
}  
if(e.getSource()==deletion[8])  
{  
    getcart ob = new getcart();  
    ob.delete(8);  
    this.dispose();  
  
}  
if(e.getSource()==deletion[9])  
{  
    getcart ob = new getcart();
```

```
        ob.delete(9);
        this.dispose();

    }
    if(e.getSource()===deletion[10])
    {
        getcart ob = new getcart();
        ob.delete(10);
        this.dispose();

    }
    if(e.getSource()===deletion[11])
    {
        getcart ob = new getcart();
        ob.delete(11);
        this.dispose();

    }
    if(e.getSource()===deletion[12])
    {
        getcart ob = new getcart();
        ob.delete(12);
        this.dispose();
    }
```

```
}  
if(e.getSource()==deletion[13])  
{  
    getcart ob = new getcart();  
    ob.delete(13);  
    this.dispose();  
  
}  
if(e.getSource()==deletion[14])  
{  
    getcart ob = new getcart();  
    ob.delete(14);  
    this.dispose();  
  
}  
if(e.getSource()==deletion[15])  
{  
    getcart ob = new getcart();  
    ob.delete(15);  
    this.dispose();  
  
}
```

```
if(e.getSource()==proceed)
{
    this.dispose();
        new Apple_Pay(gtotal);
        String forder=Arrays.toString(getcart.ap);
        System.out.println(forder);
}
```

```
}
```

```
public void remove()
{
```

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

```
}
```

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



}

}

### **Chatbox:**

```
package SwingMpr;
```

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import java.awt.event.*;
```

```
public class Chatbox extends JFrame implements  
ActionListener
```

```
{
```

```
    JLabel label_hello, label_greeting, label_greeting2;
```

```
    JLabel[] label_chat= new JLabel[20];
```

```
    JLabel[] temp;
```

```
    String s=" ";
```

```
    String a=" ";
```

```
    boolean bool=true;
```

```
    int v=0;
```

```
//    ScrollPane scrollB1, scrollB2;
```

```
    JButton b;
```

```
    JTextField User_Textfield;
```

```
    private static int complaint_id=121314151;
```

```
    private int comp_x=25, user_x=300, y=100,  
x_width=375, y_width=25;
```

```
    int index_pos=0;
```

```

static int array_size=20;
public void create_label(String arg)
{
if (index_pos==array_size)
    {
        temp = new JLabel[array_size];
        for(int i=0;i<index_pos;i++)
            {
                label_chat[i].setVisible(false);
                //temp[i]=label_chat[i];
            }
        label_chat = temp;
        y=100;
        index_pos=0;
    }

    label_chat[index_pos] = new JLabel(arg,
SwingConstants.RIGHT);

    label_chat[index_pos].setFont(new Font("Times New
Roman",Font.BOLD,16));

    label_chat[index_pos].setForeground(Color.blue);

    label_chat[index_pos].setBounds(user_x,y,x_width,y_wi
dth);

    y+=27;

    label_chat[index_pos].setVisible(true);

```

```
this.add(label_chat[index_pos]);
```

```
index_pos++;
```

```
this.revalidate();
```

```
this.repaint();
```

```
this.setVisible(true);
```

```
}
```

```
public void create_comp_label(String arg)
```

```
{
```

```
if (index_pos==array_size)
```

```
{
```

```
temp = new JLabel[array_size];
```

```
for(int i=0;i<index_pos;i++)
```

```
{
```

```
if (label_chat[i]!=null)
```

```
{label_chat[i].setVisible(false);}
```

```
//temp[i]=label_chat[i];
```

```
}
```

```
label_chat = temp;
```

```
index_pos=0;
```

```

        y=100;
    }

    label_chat[index_pos] = new JLabel(arg,
SwingConstants.LEFT);

    label_chat[index_pos].setFont(new Font("Times New
Roman",Font.BOLD,16));

    label_chat[index_pos].setForeground(Color.GREEN);
    label_chat[index_pos].setBounds(comp_x,y,x_width,y_
width);
    y+=27;
    label_chat[index_pos].setVisible(true);

    this.add(label_chat[index_pos]);

    index_pos++;
    this.revalidate();
    this.repaint();
    this.setVisible(true);
}

Chatbox()
{
    this.setLayout(null);
    this.getContentPane().setBackground(Color.black);

```

```
this.setSize(700,700);  
this.setTitle("Newark Helpdesk");
```

```
label_hello = new JLabel("Newark Helpdesk",  
SwingConstants.CENTER);
```

```
label_hello.setFont(new Font("Times New  
Roman",Font.BOLD,19));
```

```
label_hello.setForeground(Color.BLACK);
```

```
label_hello.setBackground(Color.YELLOW);
```

```
label_hello.setBounds(0,25, 700, 50);
```

```
label_hello.setVisible(true);
```

```
label_hello.setOpaque(true);
```

```
this.add(label_hello);
```

```
//label_hello.revalidate();
```

```
//Add Username from database in the String
```

```
String S = new String("Customer");
```

```
create_comp_label("Hi "+S+",Thank you for  
reaching out to us.");
```

```
create_comp_label("Type the product you want to  
buy or have ");
```

```
create_comp_label("issues with?");
```

```
User_Textfield = new JTextField();
User_Textfield.setFont(new
Font("Tahoma",Font.BOLD,18));
User_Textfield.setBounds(10,620,550,25);
this.add(User_Textfield);
User_Textfield.setOpaque(true);
User_Textfield.setVisible(true);

b=new JButton("Send");
b.setForeground(Color.WHITE);
b.setBackground(new Color(34,139,34));
b.setFont(new
Font("Serif",Font.BOLD|Font.ITALIC,20));
b.setBounds(570,620,110,25);
b.addActionListener(this);
this.add(b);
b.setVisible(true);
b.setOpaque(true);
/* Scrollbar scrollB1 = new
Scrollbar(Scrollbar.HORIZONTAL, 0,10,0,1000);
scrollB1.setBounds(0,585,600,15);
scrollB1.setVisible(true);
```

```
        scrollB1.addAdjustmentListener(new
HorizontalAdjustmentListener());
        this.add(scrollB1, BorderLayout.EAST);

        Scrollbar scrollB2 = new
Scrollbar(Scrollbar.VERTICAL, 0,10,0,1000);
        scrollB2.setBounds(585,0,600,600);
        scrollB2.setVisible(true);
        scrollB2.setLineIncrement(5);
        scrollB2.addAdjustmentListener(new
VerticalAdjustmentListener());
        this.add(scrollB2, BorderLayout.SOUTH);
*/

        this.revalidate();
        this.repaint();
        this.setVisible(true);
    }
```

@Override

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



```

if (e.getSource()==b)
{
    if (!User_Textfield.getText().equals("")){
        String UserString = User_Textfield.getText();
        create_label(UserString);
        String[] arrOfstr = UserString.split(" ");
        if (bool==false) {
            for(int k=0;k<arrOfstr.length;k++)
            {
                if (v==0)
                    {v++;}
                else {
                    a=arrOfstr[k];
                }
                switch(a){
                case "y":
                    switch(s){
                    case "airpods":
                    case "airpod":
                        this.dispose();
                        new Airpod();
                        break;
                    case "iphone":

```

```
        this.dispose();
        new Iphone();
        break;
    case "ipad":
        this.dispose();
        new Ipad();
        break;
    case "macbook":
        this.dispose();
        new MacBook();
        break;
    case "login":
        if(Login_Page.bt==true)
        {
            this.dispose();
            new Login_Page();
        }
        else
        {
            create_comp_label("You
are already logged in,STUPID!");

            create_comp_label("Anyways do you want to register
your friends too ??y/n");
```

```
                s= "register";
            }
            break;
        case "register":
            this.dispose();
            new Sign_Up();
            break;
        case "store":
            this.dispose();
            new Store();
            break;
        case "cart":
            this.dispose();
            getcart oc= new getcart();
            oc.create_Frame(getcart.id,
getcart.ap, getcart.mc, getcart.ip, getcart.idcost, getcart.ipcost,
getcart.apcost, getcart.apcost);
            break;
        case "airpods_warranty":
            replacement_label("airpods");
            s=" ";
            break;
        case "iphone_warranty":
```

```
        replacement_label("iphone");
        s=" ";
        break;
    case "ipad_warranty":
        replacement_label("ipad");
        s=" ";
        break;
    case "macbook_warranty":
        replacement_label("macbook");
        s=" ";
        break;
    }
    break;
case "n":
    switch(s){
    case "register":
        create_comp_label("Sorry you
don't have an option, you have to register a friend");
        create_comp_label("Haha just
joking");

        this.dispose();
        new Store();
        break;
```

```
case "airpods":
case "airpod":
    create_comp_label("Okay, is
your airpods in warranty?y/n");
    s="airpods_warranty";
    break;
case "airpods_warranty":
    complaint_label();
    s = " ";
    break;
case "iphone":
    create_comp_label("Okay, is
your iphone in warranty?y/n");
    s="iphone_warranty";

    break;
case "iphone_warranty":
    complaint_label();
    s = " ";
    break;
case "ipad":
    create_comp_label("Okay, is
your ipad in warranty?y/n");
    bool=false;
```

```
s="ipad_warranty";

break;
case "ipad_warranty":
    complaint_label();
    s = " ";
    break;

case "macbook":
    create_comp_label("Okay, is
your macbook in warranty?y/n");
    s="macbook_warranty";

    break;
case "macbook_warranty":
    complaint_label();
    s = " ";
    break;
case "store":
    create_comp_label("Type in
your request again.");

    break;
case "cart":
```

```

        create_comp_label("Type in
your request again.");
        break;
    }
    break;
}
}
}
else {
    for(int k=0;k<arrOfstr.length;k++)
    {
        switch(arrOfstr[k].toLowerCase()){
            case "airpod":
            case "airpods":
                create_comp_label("Do you
want to buy Airpods?y/n");
                s=new String("airpods");
                break;
            case "iphone":
                create_comp_label("Do you
want to buy Iphone?y/n");
                s=new String("iphone");
                break;
            case "ipad":

```

```
        create_comp_label("Do you  
want to buy Ipad?y/n");  
        s=new String("ipad");  
        break;  
    case "macbook":  
        create_comp_label("Do you  
want to buy Macbook?y/n");  
        s=new String("macbook");  
        break;  
    case "login":  
        create_comp_label("Do you  
want to go to Login Page?y/n");  
        s=new String("login");  
        break;  
  
    case "store":  
        create_comp_label("Do you  
want to go to Store Home Page?y/n");  
        s=new String("store");  
        break;  
    case "cart":  
        create_comp_label("Do you  
want to go proceed to cart?y/n");  
        s=new String("cart");
```



```
        break;
    case "n":
        a="n";
        bool=false;
        create_comp_label("Describe
your issue");
```

```
        break;
    case "y":
        a="y";
        switch(s){
        case "airpods":
        case "airpod":
            this.dispose();
            new Airpod();
            break;
        case "iphone":
            this.dispose();
            new Iphone();
            break;
        case "ipad":
            this.dispose();
            new Ipad();
            break;
```

```
case "macbook":
    this.dispose();
    new MacBook();
    break;
case "login":
    if(Login_Page.bt==true)
    {
        this.dispose();
        new Login_Page();
    }
    else
    {
```

```
        create_comp_label("You are already logged
in,STUPID!");
```

```
        create_comp_label("Anyways do you want to register
your friends too ??y/n");
```

```
        s= "register";
    }
    break;
case "register":
    this.dispose();
    new Sign_Up();
```

```

        break;
    case "store":
        this.dispose();
        new Store();
        break;
    case "cart":
        this.dispose();
        getcart oc= new getcart();
        oc.create_Frame(getcart.id,
getcart.ap, getcart.mc, getcart.ip, getcart.idcost, getcart.ipcost,
getcart.apcost, getcart.apcost);

        break;
    case "airpods_warranty":

replacement_label("airpods");

        s=" ";
        break;
    case "iphone_warranty":

replacement_label("iphone");

        s=" ";
        break;
    case "ipad_warranty":
        replacement_label("ipad");

```

```

                s=" ";
                break;
            case "macbook_warranty":

replacement_label("macbook");

                s=" ";
                break;
            }
            break;
        }
    }

    }
    User_Textfield.setText("");
    }
}
}

```

```

public void replacement_label(String t){

        create_comp_label("A complaint has
been generated.");

        create_comp_label("Your complain
will be verified by us");

```

```
        create_comp_label("and if faulty,  
your airpods will be replaced");  
        create_comp_label("by us within 3-7  
working days.");  
        create_comp_label("The following is  
your complaint_id "+complaint_id);  
        complaint_id++;  
        v=0;  
        bool=true;  
    }
```

```
    public void complaint_label(){  
        create_comp_label("A complaint has  
been generated.");  
        create_comp_label("Our Support  
Executive will get in touch");  
        create_comp_label("with you within  
the next 24 hours");  
        create_comp_label("The following is  
your complaint_id "+complaint_id);  
        complaint_id++;  
        v=0;  
        bool=true;  
    }
```

```
public static void main(String[] Args)
{
    new Chatbox();
}
}
```

### **Emailsend:**

```
package SwingMpr;
```

```
import java.util.Properties;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
import javax.mail.Authenticator;
```

```
import javax.mail.Message;
```

```
import javax.mail.PasswordAuthentication;
```

```
import javax.mail.Session;
```

```
import javax.mail.internet.InternetAddress;
```

```
import javax.mail.internet.MimeMessage;
import javax.mail.Transport;
```

```
public class Emailsend {
```

```
    public int b;
```

```
    public static int verify[]= new int[25];
```

```
    public static int o=0;
```

```
    int check=1;
```

```
    public static void sendMail(String receipient,int number)
    throws Exception
```

```
    {
```

```
        System.out.println("Preparing to send message");
```

```
        String host="smtp.gmail.com";
```

```
        //get the system properties
```

```
        Properties properties = System.getProperties();
```

```
        System.out.println("PROPERTIES "+properties);
```

```
        properties.put("mail.smtp.host", host);
```

```
        properties.put("mail.smtp.port","465");
```

```
properties.put("mail.smtp.ssl.enable","true");  
properties.put("mail.smtp.auth","true");
```

```
String myAccountEmail =  
"newarkstoresindia@gmail.com";
```

```
String password ="Qwerty@1@2";
```

```
Session session= Session.getInstance(properties,  
new Authenticator() {
```

```
    protected PasswordAuthentication  
    getPasswordAuthentication() {
```

```
        return new  
        PasswordAuthentication(myAccountEmail, password);  
    }
```

```
    });
```

```
Message message =  
prepareMessage(session,myAccountEmail,receipient,number);
```

```
Transport.send(message);
```

```
System.out.println("Message send sucessfully");
```



```
}
```

```
    private static Message prepareMessage(Session
session,String myAccountEmail,String receipient,int number)
{
    try
    {
        Message message = new
MimeMessage(session);
        message.setFrom(new
InternetAddress(myAccountEmail));

        message.setRecipient(Message.RecipientType.TO, new
InternetAddress(receipient));

        message.setSubject("One Time Password");
        System.out.println(number);
        message.setText(Integer.toString(number));
        o++;
        System.out.print(o);
        if(o>=25)
        {
```

```

        o=0;
    }
    return message;

}
catch(Exception ex)
{   System.out.println("Incorrect email");

    Logger.getLogger(Emailsend.class.getName()).log(Level
.SEVERE,null,ex);
}
return null;

}

public int generate()
{
    int min = 2000;
    int max = 4000;
    double a = Math.random()*(max-min+1)+min;
    b = (int)(Math.random()*(max-min+1)+min);
    System.out.println(b);
    verify[o]=b;
    System.out.println(verify[o]);

```

```
return b;
```

```
}
```

```
public int verified (String s )
```

```
{
```

```
    System.out.println(Integer.parseInt(s));
```

```
    for(int j=0;j<25;j++)
```

```
    {   System.out.println("a"+verify[j]);
```

```
        if(verify[j]==Integer.parseInt(s))
```

```
        {
```

```
            check=0;
```

```
        }
```

```
    }
```

```
    return check;
```

```
}
```

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

### **FirstJDBC:**

```
package SwingMpr;  
import java.sql.*;  
public class FirstJDBC {  
  
    final String url  
="jdbc:mysql://localhost:3306/Newarkstore";  
    final String username="root";
```

```

public Connection connectDB() {

    Connection con=null;

    try
    {
        Class.forName("com.mysql.cj.jdbc.Driver");

        String password ="B@rcelona123";

        con=
DriverManager.getConnection(url,username,password);
        if(con.isClosed())
        {
            System.out.println("Connection
closes");
        }
        else

```

```

        {
            System.out.println("Connection
done");
        }

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

```

### **ForgotPassword:**

```

package SwingMpr;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.*;

```

```
public class ForgotPassword extends JFrame implements  
ActionListener{
```

```
    JLabel email_f,email_en,email_v,label_error4;
```

```
    JTextField text_email_en;
```

```
    JPasswordField text_email_v;
```

```
    JButton b1,b2;
```

```
    String y;
```

```
    public static String reset_email;
```

```
    static int k=0;
```

```
    Emailsend u = new Emailsend();
```

```
    ForgotPassword(String login)
```

```
    {
```

```
        this.setLayout(null);
```

```
        this.getContentPane().setBackground(Color.black);
```

```
        this.setSize(800,600);
```

```
        this.setTitle("Forget Password");
```

```
        email_f =new JLabel("Email Verification");
```

```
email_f.setFont(new  
Font("Tahoma",Font.BOLD,25));  
email_f.setForeground(Color.BLACK);  
email_f.setBackground(Color.WHITE);  
email_f.setBounds(10,50,800,50);  
email_f.setVisible(true);  
email_f.setOpaque(true);  
this.add(email_f);
```

```
email_en =new JLabel("Enter Email");  
email_en.setFont(new  
Font("Tahoma",Font.BOLD,25));  
email_en.setForeground(Color.white);  
email_en.setBounds(50,160,200,50);  
email_en.setVisible(true);  
  
this.add(email_en);
```

```
text_email_en = new JTextField();  
text_email_en.setFont(new  
Font("TimesRoman",Font.BOLD,24));
```



```
text_email_en.setBounds(275,170,250,35);  
this.add(text_email_en);
```

```
email_v =new JLabel("Email Code");  
    email_v.setFont(new  
Font("Tahoma",Font.BOLD,25));  
    email_v.setForeground(Color.white);  
    email_v.setBounds(50,230,200,50);  
    email_v.setVisible(true);  
    this.add(email_v);
```

```
    text_email_v = new JPasswordField();  
    text_email_v.setFont(new  
Font("TimesRoman",Font.BOLD,24));  
    text_email_v.setBounds(275,230,250,35);  
    this.add(text_email_v);
```

```
label_error4 =new JLabel();  
    label_error4.setFont(new  
Font("Tahoma",Font.BOLD,20));  
    label_error4.setForeground(Color.white);  
    label_error4.setBounds(50,280,400,50);  
    label_error4.setVisible(false);
```

```
this.add(label_error4);
```

```
b1=new JButton("Verify Email");  
    b1.setForeground(Color.white);  
    b1.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
    b1.setBounds(575,180,150,25);  
    b1.setOpaque(true);  
    b1.setBackground(Color.black);  
    b1.addActionListener(this);  
  
this.add(b1);
```

```
b2=new JButton("Proceed");  
    b2.setForeground(Color.white);  
    b2.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
    b2.setBounds(575,280,150,25);  
    b2.setOpaque(true);
```

```
        b2.setBackground(Color.black);
        b2.addActionListener(this);
        //b2.setVisible(false);
        this.add(b2);
        this.revalidate();
        this.repaint();
        this.setVisible(true);
    }
```

@Override

```
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    if(e.getSource()==b1)
    {
        if(text_email_en.getText().equals(""))==true)
        {
            label_error4.setVisible(true);
            label_error4.setText("Enter your email");
        }
        else
        {
            label_error4.setVisible(false);
            int l=u.generate();
            System.out.println(l);
        }
    }
}
```

```

        System.out.println("one");
        y=text_email_en.getText();
        try {
            System.out.print("Pika"+k++);
            sendmail(y,l);
        } catch (Exception e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
    this.revalidate();
    this.repaint();
    this.setVisible(true);
}

```

```

if(e.getSource()==b2)
{

```

```

    if(String.valueOf(text_email_v.getPassword()).equals(""))
    ==true)
    {
        label_error4.setVisible(true);
    }
}

```

```

        label_error4.setText("Enter your
password");

        this.revalidate();
        this.repaint();
        this.setVisible(true);
    }
    else
    if(u.verified(String.valueOf(text_email_v.getPassword()))==0
    )

    {
        System.out.print(getFocusTraversalKeysEnabled());

        String d="";
        String email="";

        FirstJDBC ob = new FirstJDBC();
        String s=text_email_en.getText();

        Connection con = null;
        PreparedStatement p = null;
        ResultSet rs = null;

        con = ob.connectDB();
        try
        {

```

```

        String sql = "select * from
customer_id where email_id='"+s+"'";

        System.out.println(sql);
        p = con.prepareStatement(sql);
        rs= p.executeQuery();

        if(rs.next())
        {
            d = rs.getString("password");
            System.out.println(d);
            email = rs.getString("email_id");
            System.out.println(email);
            System.out.println();
        }

    }

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

```

```
label_error4.setText("Invalid  
Credentials");  
  
}
```

```
if(email.equals(""))  
{ System.out.print("F");  
    label_error4.setVisible(true);  
    label_error4.setText("NO  
ACCOUNT");  
  
    this.revalidate();  
    this.repaint();  
    this.setVisible(true);  
}  
else {
```

```
    reset_email=text_email_en.getText();  
    this.dispose();  
    new ResetPassword();  
}
```

```
}
```

```
}
```

```
}
```

```
public static void sendmail(String y,int l) throws Exception{  
    Emailsend.sendMail(y,l);
```

```
    System.out.println(k);
```

```
    k++;
```

```
}
```

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    new ForgotPassword("he");  
}
```

```
}
```



### **Getcart:**

```
package SwingMpr;  
import java.util.Arrays;
```

```
public class getcart {
```

```
    static public String ip[]= new String[5];  
    static public double ipcost[]= new double[5];  
    static public String id[]= new String[5];
```

```
static public double idcost[]= new double[5];
static public String mc[]= new String[5];
static public double mccost[]= new double[5];
static public String ap[]= new String[5];
static public double apcost[]= new double[5];
static int g=-1,f=-1,e=-1,h=-1;
```

```
static
{
    for(int l=0;l<5;l++)
    {
        ip[l]="";
        ipcost[l]=0.00;
        id[l]="";
        idcost[l]=0.00;
        mc[l]="";
        mccost[l]=0.00;
        ap[l]="";
        apcost[l]=0.00;
    }
}
```

```

public void set_Ipad(String s,Double d)
{
    if(g<3)
    {
        g++;
        id[g]=s;
        idcost[g]=d;
    }

    else
    g=-1;

    //new Cartdummy(id, ap, mc,
ip,idcost,ipcost,mccost,apcost);

}

public void set_Iphone(String s,Double d)
{
    if(f<3)
    {
        f++;
        ip[f]=s;
        ipcost[f]=d;
    }
}

```

```
}
```

```
else
```

```
f=-1;
```

```
    //new Cartdummy(id, ap, mc,  
ip,idcost,ipcost,mccost,apcost);
```

```
}
```

```
public void set_Macbook(String s,Double d)
```

```
{    if(e<3)
```

```
{
```

```
    e++;
```

```
    mc[e]=s;
```

```
    mccost[e]=d;
```

```
}
```

```
else
```

```
e=-1;
```

```
    //new Cartdummy(id, ap, mc,  
ip,idcost,ipcost,mccost,apcost);
```

```

    }

    public void set_Airpod(String s,Double d)
    {
        if(h<3)
        {
            h++;
            ap[h]=s;
            apcost[h]=d;
        }

        else
        h=-1;

        //new Cartdummy(id, ap, mc,
ip,idcost,ipcost,mccost,apcost);
    }

    public void create_Frame(String[]l, String[]m,
String[]n,String[]o, double[]p, double[]q,double[]r,double[]s)
    {
        new Cartdummy(l,m,n,o,p,q,r,s);

    }

```

```

public void delete(int index)
{
    if(index>=0 && index<4)
    {
        int u;

        int a_index=index;
        id[a_index]=" ";
        // getcart uy = new getcart();

        new Cartdummy(idcost[a_index],"remove");
        idcost[a_index]=0.0;
        for(u=a_index;u<4;u++)
        {
            id[u]=id[u+1];
            idcost[u]=idcost[u+1];
        }
        if(g== -1)
        {
            g=4;
        }
        else
        {
            g--;
        }
    }
}

```

```
}
```

```
        create_Frame(getcart.id, getcart.ap, getcart.mc,  
getcart.ip, getcart.idcost, getcart.ipcost, getcart.mccost,  
getcart.apcost);
```

```
}
```

```
if(index>3 && index<8)
```

```
{
```

```
    int a_index=index-4;
```

```
    int u;
```

```
    ip[a_index]=" ";
```

```
    // getcart uy = new getcart();
```

```
        new Cartdummy(ipcost[a_index],"remove");
```

```
    ipcost[a_index]=0.0;
```

```
    for(u=a_index;u<4;u++)
```

```
{
```

```
        ip[u]=ip[u+1];
```

```
        ipcost[u]=ipcost[u+1];
```

```

    }
    if(f==-1)
    {
        f=4;
    }
    else
    {
        f--;
    }

```

```

        create_Frame(getcart.id, getcart.ap, getcart.mc,
getcart.ip, getcart.idcost, getcart.ipcost, getcart.mccost,
getcart.apcost);

```

```

    }
    if(index>7 && index<12)
    {   int u;

        int a_index=index-8;
        mc[a_index]=" ";
        // getcart uy = new getcart();

```

```

        new
Cartdummy(mccost[a_index],"remove");
        mccost[a_index]=0.0;
        for(u=a_index;u<4;u++)

```



```

{
    mc[u]=mc[u+1];
    mccost[u]=mccost[u+1];
}
if(e==-1)
{
    e=4;
}
else
{
    e--;
}

```

```

        create_Frame(getcart.id, getcart.ap, getcart.mc,
getcart.ip, getcart.idcost, getcart.ipcost, getcart.mccost,
getcart.apcost);

```

```

    }
    if(index>11 && index<16)
    {   int u;
        int a_index=index-12;
        ap[a_index]=" ";
        // getcart uy = new getcart();
    }
}

```

```
new Cartdummy(apcost[a_index],"remove");  
apcost[a_index]=0.0;
```

```
for(u=a_index;u<4;u++)  
{  
    ap[u]=ap[u+1];  
    apcost[u]=apcost[u+1];  
}
```

```
if(h==-1)
```

```
{  
    h=4;  
}
```

```
else
```

```
{  
    h--;  
}
```

```
create_Frame(getcart.id, getcart.ap, getcart.mc,  
getcart.ip, getcart.idcost, getcart.ipcost, getcart.mccost,  
getcart.apcost);
```

```
}
```

```
}
```

```
public static void main(String[] args) {  
    getcart og=new getcart();  
    System.out.println(getcart.ip[1]);  
    og.send_Label(id,ap,mc,ip);  
  
}  
  
}
```

### **Ipad:**

```
package SwingMpr;
```

```
import java.awt.Button;  
import java.awt.Checkbox;  
import java.awt.CheckboxGroup;  
import java.awt.Color;  
import java.awt.Font;  
import java.awt.Frame;
```

```
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import javax.swing.*;

public class Ipad extends JFrame implements
ActionListener,ItemListener{
```

```
    JLabel
Ip_edition,Ip_model,Ip_memory,Ip_color,output,acesories;

    JRadioButton c5,c6,c7,c8,c9,c10,c11,c12,c13;
    JCheckBox c1,c2,c3,c4;
    List list;
    ButtonGroup cg,cg1,cg2,cg3;
    JButton b1,b2,b3,b4;
    String Idcolor,Idmemory,Idedition,acc_final;
    String[] acc = new String[4];

    static String[] idbill = new String[5];
    static int z=0;
    double fprice;
```

```
double  
base_ipad=0,a_price1=0,base_memory=0,a_price2=0,a_price  
3=0,a_price4=0,base_price=0;
```

```
Ipad(int z,String exceed)  
{  
    this.setSize(450,800);  
    this.setVisible(true);  
    this.setLayout(null);  
    this.getContentPane().setBackground(new  
Color(168, 169, 173));  
    Label l1= new Label("Exceeding:ITEM LIMIT  
4");  
    new Store();  
    l1.setFont(new Font("Tahoma",Font.PLAIN,15));  
    l1.setBounds(25, 50, 250,40);  
    this.add(l1);  
  
}
```

```
Ipad()
```

```
{ this.setSize(650,900);

    this.setLayout(null);
    this.getContentPane().setBackground(new Color(168,
169, 173));
    this.setTitle("Ipad");
        Ip_edition = new JLabel("Please choose a Ipad?");
        Ip_edition.setBounds(25, 50, 250,40);
        Ip_edition.setFont(new
Font("Arial",Font.PLAIN,16));
        this.add(Ip_edition);

        b4 = new JButton("Back to store");
        b4.setBounds(300,740,200,25);
        b4.setFont(new
Font("Arial",Font.PLAIN|Font.ITALIC,16));
        b4.setForeground(Color.red);
        b4.addActionListener(this);
        this.add(b4);

        list = new List(3,false);
        list.add("ipad mini");
        list.add("ipad (entry level)");
```

```
list.add("ipad Air");  
list.add("Ipad Pro");  
list.setBounds(25,100,250,20);  
list.addActionListener(this);  
this.add(list);
```

```
Ip_model = new JLabel("Please choose Your  
Acessories ");
```

```
Ip_model.setBounds(25,130,250,20);  
Ip_model.setFont(new  
Font("Arial",Font.PLAIN,16));  
this.add(Ip_model);
```

```
c1= new JCheckBox("magic keyboard",false);  
c1.setBounds(25,170,250,12);  
c1.addItemListener(this);  
c1.setBackground(new Color(168, 169, 173));  
this.add(c1);
```

```
c2= new JCheckBox("apple pencil 2nd  
generation",false);  
c2.setBounds(25,190,200,12);
```

```
c2.addItemListener(this);  
c2.setBackground(new Color(168, 169, 173));  
this.add(c2);
```

```
c3= new JCheckBox("protective case",false);  
c3.setBounds(25,210,250,12);  
c3.addItemListener(this);  
c3.setBackground(new Color(168, 169, 173));  
this.add(c3);  
c4= new JCheckBox("Screen guard",false);  
c4.setBounds(25,230,260,12);  
c4.addItemListener(this);  
c4.setBackground(new Color(168, 169, 173));  
this.add(c4);
```

```
Ip_memory=new JLabel("Please choose a memory  
option?");  
Ip_memory.setBounds(25,260,250,20);  
Ip_memory.setFont(new  
Font("Arial",Font.PLAIN,16));  
this.add(Ip_memory);
```



```
cg2 = new ButtonGroup();
c5= new JRadioButton("32GB");
c5.setBounds(25,290,80,12);
c5.setBackground(new Color(168, 169, 173));
c5.addItemListener(this);
this.add(c5);
c6= new JRadioButton("64GB");
c6.setBounds(25,320,80,12);
c6.setBackground(new Color(168, 169, 173));
c6.addItemListener(this);
this.add(c6);
c7= new JRadioButton("128GB");
c7.setBounds(25,350,80,12);
c7.setBackground(new Color(168, 169, 173));
c7.addItemListener(this);
this.add(c7);
c8= new JRadioButton("512GB");
c8.setBounds(25,380,80,12);
c8.addItemListener(this);
c8.setBackground(new Color(168, 169, 173));
this.add(c8);
cg2.add(c5);
cg2.add(c6);
```

```
cg2.add(c7);
```

```
cg2.add(c8);
```

```
Ip_color=new JLabel("Chose a Color ");
```

```
Ip_color.setBounds(25,410, 250,20);
```

```
Ip_color.setFont(new  
Font("Arial",Font.PLAIN,16));
```

```
this.add(Ip_color);
```

```
cg3 = new ButtonGroup();
```

```
c9= new JRadioButton("RED");
```

```
c9.setBounds(25,440,180,12);
```

```
c9.addItemListener(this);
```

```
c9.setBackground(new Color(168, 169, 173));
```

```
this.add(c9);
```

```
c10 = new JRadioButton("ROSE-GOLD");
```

```
c10.setBounds(25,470,180,12);
```

```
c10.addItemListener(this);
```

```
c10.setBackground(new Color(168, 169, 173));
```

```
this.add(c10);
```

```
c11= new JRadioButton("BLUE");
```

```
c11.setBounds(25,500,180,12);
```

```
c11.addItemListener(this);
```

```
c11.setBackground(new Color(168, 169, 173));  
this.add(c11);  
c12= new JRadioButton("SPACE GREY");  
c12.setBounds(25,530,180,12);  
c12.addItemListener(this);  
c12.setBackground(new Color(168, 169, 173));  
this.add(c12);  
cg3.add(c9);  
cg3.add(c10);  
cg3.add(c11);  
cg3.add(c12);
```

```
b1= new JButton("DONE");  
b1.setBackground(Color.white);  
b1.setForeground(Color.DARK_GRAY);  
b1.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b1.setBounds(160,570,80,40);  
b1.addActionListener(this);  
this.add(b1);  
  
b2= new JButton("CLEAR");
```

```
b2.setBackground(Color.GREEN);  
b2.setForeground(Color.WHITE);  
b2.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b2.setBounds(260,570,80,40);  
b2.addActionListener(this);  
this.add(b2);
```

```
b3= new JButton("CART");  
b3.setBackground(Color.BLUE);  
b3.setForeground(Color.WHITE);  
b3.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b3.setBounds(60,570,80,40);  
b3.addActionListener(this);  
this.add(b3);
```

```
output=new JLabel();  
output.setBounds(25,640,600,20);  
output.setFont(new Font("Arial",Font.PLAIN,16));  
this.add(output);  
accesories=new JLabel();  
accesories.setBounds(25,680,600,20);
```

```
        accessories.setFont(new  
Font("Arial",Font.PLAIN,16));  
        this.add(accessories);  
        this.revalidate();  
        this.repaint();  
        this.setVisible(true);  
  
    }
```

@Override

```
public void actionPerformed(ActionEvent e) {  
  
    if(e.getSource()==list)  
  
    {  
        if(list.getSelectedItem()==null)  
        {  
            output.setText("Please enter a Ipad Model ");  
        }  
    }  
}
```

```
else
{
    Idedition=list.getSelectedItemAt();
    if(Idedition.equals("ipad mini"))
    {
        base_price=40000.0;

    }
    else if(Idedition.equals("ipad (entry
level)"))
    {
        base_price=30000.0;

    }
    else if(Idedition.equals("ipad Air"))
    {
        base_price=50000.0;

    }
    else if(Idedition.equals("Ipad Pro"))
    {
        base_price=90000.0;

    }
}
```

```

        }

        this.revalidate();
        this.repaint();
        this.setVisible(true);

    }

    if(e.getSource()==b1)
    {

        if(base_price!=0.0 && base_memory!=0.0 )
        {

            //idbill[j]= Idedition+" "+Idmemory+"
            "+Idcolor+" "+fprice;

            fprice=
            base_memory+base_price+a_price1+a_price2+a_price3+a_pr
            ice4;

            output.setText(Idedition+" "+Idmemory+"
            "+Idcolor+" "+fprice);

            accesories.setText(acc[0]+acc[1]+acc[2]+acc[3]);

            //Icount a =new Icount();

```

```

//Icount.price[1]=fprice;
getcart ob = new getcart();
new Cartdummy(fprice,"add");
ob.set_Ipad(Idedition+" "+Idmemory+"
"+Idcolor+"("+acc[0]+acc[1]+acc[2]+acc[3]+")", fprice);

z++;
if(z>4)
{
    this.dispose();
    new Ipad(z,"Ipad");

}

//a.setPrice(fprice,Idedition+" "+Idmemory+"
"+Idcolor+"("+acc[0]+acc[1]+acc[2]+acc[3]+")");

//a.getPrice();y.artoutput(Idedition+"
"+Idmemory+" "+Idcolor+"("+acc[0]+acc[1]+acc[2]+acc[3],
fprice);

}

else
{

```



```
output.setText("select all options");
```

```
}
```

```
this.revalidate();
```

```
this.repaint();
```

```
this.setVisible(true);
```

```
}
```

```
if(e.getSource()==b2)
```

```
{
```

```
    int o = list.getSelectedIndex();
```

```
    list.deselect(o);
```

```
    c1.setSelected(rootPaneCheckingEnabled);
```

```
    c2.setSelected(false);
```

```
    c3.setSelected(false);
```

```
    c4.setSelected(false);
```

```
    c5.setSelected(false);
```

```
    c6.setSelected(false);
```

```
    c7.setSelected(false);
```

```
    c8.setSelected(false);
```

```
    c9.setSelected(false);
```

```
    c10.setSelected(false);
```

```
        c11.setSelected(false);  
        c12.setSelected(false);  
        this.dispose();  
        new Ipad();  
    }
```

```
    if(e.getSource()==b3)  
    {  
        this.dispose();  
        getcart oc= new getcart();  
        oc.create_Frame(getcart.id, getcart.ap,  
getcart.mc, getcart.ip, getcart.idcost, getcart.ipcost,  
getcart.mccost, getcart.apcost);  
    }
```

```
    if(e.getSource()==b4)  
    {  
        this.dispose();  
        new Store();  
    }  
}
```

```
@Override
public void itemStateChanged(ItemEvent e) {

    if(c1.isSelected() == true)
    {

        acc[0] = "Magic keyboard ";
        a_price1 = 7000;
    }
    if(c1.isSelected() == false)
    {

        acc[0] = " ";
        a_price1 = 0;
    }
}
```

```
if(c2.isSelected()==true)
{
    acc[1]="apple pencil 2nd generation ";
    a_price2=8000;

}
if(c2.isSelected()==false)
{

    acc[1]=" ";
    a_price2=0;
}
if(c3.isSelected()==true)
{
    acc[2]="protective Case ";
    a_price3=1000;

}
if(c3.isSelected()==false)
{

    acc[2]=" ";
    a_price3=0;
```

```
}  
if(c4.isSelected()==true)  
{  
    acc[3]="Screen Guard";  
    a_price4=1000;  
}  
if(c4.isSelected()==false)  
{  
  
    acc[3]=" ";  
    a_price4=0;  
}  
if(c1.isSelected()==false && c2.isSelected()==false  
&& c3.isSelected()==false && c4.isSelected()==false)  
{  
    accesories.setText("Acessories: None");  
}  
  
if(c5.isSelected()==true)  
{  
  
    Idmemory="32GB";
```

```
        base_memory=0.01;

    }
    if(c6.isSelected()==true)
    {
        Idmemory="64GB";
        base_memory=5000.0;

    }
    if(c7.isSelected()==true)
    {
        Idmemory="128GB";
        base_memory=10000.0;

    }
    if(c8.isSelected()==true)
    {
        Idmemory="512GB";
        base_memory=20000.0;
    }
```

```
        /*if(c8.isSelected()==false &&
c7.isSelected()==false && c6.isSelected()==false &&
c5.isSelected()==false)
        {
            output.setText("select all options");

        }*/
        if(c9.isSelected()==true)
        {

            Idcolor="RED";

        }
        if(c10.isSelected()==true)
        {
            Idcolor="ROSE-GOLD";

        }
        if(c11.isSelected()==true)
        {
            Idcolor="SPACE GREY";

        }
```

```
        if(c12.isSelected()==true)
        {
            Idcolor="BLUE";

        }

        /*if(c9.isSelected()==false &&
c10.isSelected()==false && c11.isSelected()==false &&
c12.isSelected()==false)
        {
            output.setText("Select all options");
        }*/

    }
```

```
public static void main(String[] args)
{
    new Ipad();
}
```



```
}
```

### **Iphone:**

```
package SwingMpr;
```

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import java.awt.event.*;
```

```
import java.awt.event.ActionListener;
```

```
public class Iphone extends JFrame implements  
ActionListener,ItemListener {
```

```

        JLabel
Ip_edition,Ip_model,Ip_memory,Ip_color,Ip_output,Ip_price;

        JRadioButton
c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13;

        List list;

        ButtonGroup cg,cg1,cg2,cg3;

        JButton b1,b2,b3,b4;

        String model,Icolor,memory,edition;

        double base_iphone,memory_iphone;

        Iphone()
        {   this.setSize(450,800);

            this.setTitle("Iphone");


            this.setLayout(null);

            this.getContentPane().setBackground(new Color(168,
169, 173));

            Ip_edition = new JLabel("Please choose a
edition?");

            Ip_edition.setBounds(25, 50, 250,40);

            Ip_edition.setFont(new
Font("Arial",Font.PLAIN,16));

            this.add(Ip_edition);

```

```
list = new List(3,false);  
list.add("iphone 12");  
list.add("iphone 13");  
list.setBounds(25,100,250,20);  
list.addActionListener(this);  
this.add(list);
```

```
Ip_model = new JLabel("Please choose a model?");  
Ip_model.setBounds(25,130,250,20);  
Ip_model.setFont(new  
Font("Arial",Font.PLAIN,16));  
this.add(Ip_model);
```

```
cg = new ButtonGroup();  
c1= new JRadioButton("Mini");  
c1.setBounds(25,170,75,12);  
c1.setBackground(new Color(168, 169, 173));  
c1.addItemListener(this);  
this.add(c1);
```

```
c3= new JRadioButton("Pro");  
c3.setBounds(25,210,75,12);
```

```
c3.setBackground(new Color(168, 169, 173));  
c3.addItemListener(this);  
this.add(c3);  
c4= new JRadioButton("Pro max");  
c4.setBounds(25,230,75,12);  
c4.setBackground(new Color(168, 169, 173));  
c4.addItemListener(this);  
this.add(c4);
```

```
Ip_memory=new JLabel("Please choose a memory  
option?");
```

```
Ip_memory.setBounds(25,260,250,20);  
Ip_memory.setFont(new  
Font("Arial",Font.PLAIN,16));  
this.add(Ip_memory);  
cg2 = new ButtonGroup();  
c5= new JRadioButton("128GB");  
c5.setBounds(25,290,75,12);  
c5.setBackground(new Color(168, 169, 173));  
c5.addItemListener(this);  
this.add(c5);
```

```
c6= new JRadioButton("256GB");
c6.setBounds(25,320,75,12);
c6.setBackground(new Color(168, 169, 173));
c6.addItemListener(this);
this.add(c6);

c7= new JRadioButton("512GB");
c7.setBounds(25,350,75,12);
c7.setBackground(new Color(168, 169, 173));
c7.addItemListener(this);
this.add(c7);

c8= new JRadioButton("1 TB");
c8.setBounds(25,380,75,12);
c8.setBackground(new Color(168, 169, 173));
c8.addItemListener(this);
this.add(c8);
```

```
Ip_color=new JLabel("Chose a Color ");
Ip_color.setBounds(25,410, 250,20);
Ip_color.setFont(new
Font("Arial",Font.PLAIN,16));
this.add(Ip_color);
cg3 = new ButtonGroup();
```

```
c9= new JRadioButton("RED");
c9.setBounds(25,440,80,12);
c9.setBackground(new Color(168, 169, 173));
c9.addItemListener(this);
this.add(c9);

c10 = new JRadioButton("ROSE-GOLD");
c10.setBounds(25,470,110,12);
c10.setBackground(new Color(168, 169, 173));
c10.addItemListener(this);
this.add(c10);

c11= new JRadioButton("BLUE");
c11.setBounds(25,500,80,12);
c11.setBackground(new Color(168, 169, 173));
c11.addItemListener(this);
this.add(c11);

c12= new JRadioButton("SPACE GREY");
c12.setBounds(25,530,110,12);
c12.setBackground(new Color(168, 169, 173));
c12.addItemListener(this);
this.add(c12);

b1= new JButton("DONE");
```

```
b1.setBackground(Color.white);  
b1.setForeground(Color.DARK_GRAY);  
b1.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b1.setBounds(160,570,80,40);  
b1.addActionListener(this);  
this.add(b1);
```

```
b2= new JButton("CLEAR");  
b2.setBackground(Color.GREEN);  
b2.setForeground(Color.WHITE);  
b2.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b2.setBounds(260,570,80,40);  
b2.addActionListener(this);  
this.add(b2);
```

```
b3= new JButton("CART");  
b3.setBackground(Color.BLUE);  
b3.setForeground(Color.WHITE);  
b3.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b3.setBounds(60,570,80,40);  
b3.addActionListener(this);
```

```
this.add(b3);
```

```
b4 = new JButton("Back to store");
```

```
b4.setBounds(320,700,100,25);
```

```
b4.setFont(new  
Font("Arial",Font.PLAIN|Font.ITALIC,10));
```

```
b4.setForeground(Color.red);
```

```
b4.addActionListener(this);
```

```
this.add(b4);
```

```
Ip_output = new JLabel();
```

```
Ip_output.setBounds(25, 630,600,40);
```

```
Ip_output.setFont(new  
Font("Arial",Font.PLAIN,16));
```

```
this.add(Ip_output);
```

```
Ip_price = new JLabel();
```

```
Ip_price.setBounds(25, 680,600,40);
```

```
Ip_price.setFont(new  
Font("Arial",Font.PLAIN,16));
```

```
this.add(Ip_price);
```

```
this.revalidate();
```

```
this.repaint();
```



```
        this.setVisible(true);  
  
    }
```

@Override

```
public void actionPerformed(ActionEvent e) {  
  
    if(e.getSource()==list)  
    {  
        edition=list.getSelectedItemAt()  
        c2= new JRadioButton(edition);  
        c2.setBounds(25,190,100,12);  
        c2.setBackground(new Color(168, 169, 173));  
        c2.addItemListener(this);  
        this.add(c2);  
        this.revalidate();  
        this.repaint();  
        this.setVisible(true);  
    }  
    if(e.getSource()==b1)
```

```

        {
            if(edition==null || model==null ||
memory==null || Icolor==null)
            {
                System.out.println("Please select all the
options");
            }
            else
            { double iptotal=
base_iphone+memory_iphone;
                getcart ob = new getcart();
                new Cartdummy(iptotal,"add");

ob.set_Iphone(edition+model+memory+Icolor,iptotal);
                Ip_output.setText("SELECTED
CHOICE:"+edition+model+memory+Icolor);
                Ip_price.setText("Rs "+iptotal);

            }

            this.revalidate();
            this.repaint();
            this.setVisible(true);
        }

```

```
        if(e.getSource()==b2)
        {

            this.dispose();
            new Iphone();
        }
        if(e.getSource()==b3)
        {

            this.dispose();
            getcart oc= new getcart();
            oc.create_Frame(getcart.id, getcart.ap,
getcart.mc, getcart.ip, getcart.idcost, getcart.ipcost,
getcart.mccost, getcart.apcost);

        }
        if(e.getSource()==b4)
        {

            this.dispose();
            new Store();
        }

    }
```

@Override

public void itemStateChanged(ItemEvent e) {

    if(c1.isSelected()==true)

    {

        model="Mini";

        base\_iphone=70000;

    }

    if(c2.isSelected()==true)

    {

        model="13";

        base\_iphone=90000;

    }

    if(c3.isSelected()==true)

    {

        model="Pro";

```
        base_iphone=100000;

    }
    if(c4.isSelected()==true)
    {
        model="Pro Max";
        base_iphone=120000;
    }

    if(c5.isSelected()==true)
    {

        memory="128GB";
        memory_iphone=0;

    }
    if(c6.isSelected()==true)
    {
        memory="256GB";
        memory_iphone=5000;

    }
    if(c7.isSelected()==true)
```

```
{
    memory="512GB";
    memory_iphone=10000;

}
if(c8.isSelected()==true)
{
    memory="1 TB";
    memory_iphone=20000;
}

if(c9.isSelected()==true)
{

    Icolor="RED";

}
if(c10.isSelected()==true)
{
    Icolor="ROSE-GOLD";

}
if(c11.isSelected()==true)
```

```
{  
    Icolor="SPACE GREY";  
  
}  
if(c12.isSelected()==true)  
{  
    Icolor="BLUE";  
  
}  
  
}
```

```
public static void main(String args[])  
{  
    new Iphone();  
}  
  
}
```

### **Login Page:**

```
package SwingMpr;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import javax.swing.*;
```



```
public class Login_Page extends JFrame implements  
ActionListener
```

```
{
```

```
    public static boolean bt =true;
```

```
    JLabel lbl_username, lbl_password, lbl_heading, lbl_error;
```

```
    JTextField tx1;
```

```
    JPasswordField tx2;
```

```
    JButton b1,b2,b3,b4;
```

```
    public static String maild="";
```

```
    //public static Frame f = new Frame();
```

```
    Login_Page()
```

```
{
```

```
        this.setLayout(null);
```

```
        this.getContentPane().setBackground(Color.black);
```

```
        this.setSize(600,600);
```

```
        this.setTitle("Newark Store Login");
```

```
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    E);
```

```
        /*f.setSize(450,800);  
f.setVisible(true);  
f.setLayout(null);  
f.setBackground(new Color(168, 169, 173));*/
```

```
        lbl_heading =new JLabel("Newark  
Store",JLabel.CENTER);  
        lbl_heading.setFont(new  
Font("Serif",Font.BOLD,25));  
        lbl_heading.setForeground(Color.BLACK);  
        lbl_heading.setBackground(Color.WHITE);  
        lbl_heading.setBounds(0,15,600,50);  
        lbl_heading.setVisible(true);  
        lbl_heading.setOpaque(true);
```

```
        lbl_username= new JLabel("Email ID");  
        lbl_username.setFont(new  
Font("Serif",Font.BOLD,22));
```

```
lbl_username.setForeground(Color.white);  
lbl_username.setBounds(100,200,150,25);  
lbl_username.setVisible(true);
```

```
tx1=new JTextField(10);  
tx1.setFont(new Font("Serif",Font.BOLD,18));  
tx1.setBounds(300,200,150,25);
```

```
lbl_password= new JLabel("Password");  
lbl_password.setFont(new  
Font("Serif",Font.BOLD,22));  
lbl_password.setForeground(Color.white);  
lbl_password.setBounds(100,250,150,25);  
lbl_password.setVisible(true);
```

```
tx2=new JPasswordField(10);  
tx2.setFont(new Font("Serif",Font.BOLD,18));  
tx2.setBounds(300,250,150,25);
```

```
lbl_error= new JLabel();
```

```
        lbl_error.setFont(new  
Font("Serif",Font.BOLD,15));  
        lbl_error.setForeground(Color.white);  
        lbl_error.setBounds(150,350,150,25);
```

```
        this.add(lbl_heading);  
        this.add(lbl_username);  
        this.add(lbl_password);  
        this.add(tx1);  
        this.add(tx2);  
        this.add(lbl_error);
```

```
        b1=new JButton("LOGIN");  
        b1.setForeground(Color.white);  
        b1.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
        b1.setBounds(400,300,100,25);  
        b1.setBackground(Color.black);  
        b1.addActionListener(this);
```

```
this.add(b1);
```

```
    b2=new JButton("Create Account");  
    b2.setForeground(Color.white);  
    b2.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
    b2.setBounds(100,410,180,25);  
    b2.addActionListener(this);  
    b2.setBackground(Color.black);  
this.add(b2);
```

```
    b4=new JButton("Forget Password");  
    b4.setForeground(Color.white);  
    b4.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
    b4.setBounds(300,410,180,25);  
    b4.addActionListener(this);  
    b4.setBackground(Color.black);  
this.add(b4);
```

```
this.revalidate();
```

```
    this.repaint();
```

```
this.setVisible(true);
```

```
}
```

```
@Override
```

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

```
    if(e.getSource()==b1)
```

```
    {    String d="";
```

```
        String email="";
```

```
        FirstJDBC ob = new FirstJDBC();
```

```
        String s=tx1.getText();
```

```
        Connection con = null;
```

```
        PreparedStatement p = null;
```

```
        ResultSet rs = null;
```

```
        con = ob.connectDB();
```

```

try
{
    String sql = "select * from customer_id
where email_id='"+s+"'";

    System.out.println(sql);
    p = con.prepareStatement(sql);
    rs= p.executeQuery();

    if(rs.next())
    {
        d = rs.getString("password");
        System.out.println(d);
        email = rs.getString("email_id");
        System.out.println(email);
    }

}

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

```

```
}
```

```
        if(email.equals(tx1.getText()) &&  
d.equals(String.valueOf(tx2.getPassword())) &&  
!tx1.getText().equals("")) &&  
!String.valueOf(tx2.getPassword()).equals(""))
```

```
{
```

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

```
        Login_Page.maild=email;
```

```
        Store ps = new Store();
```

```
        bt=false;
```

```
}
```

```
else
```

```
{
```

```
        if(String.valueOf(tx2.getPassword()).equals("") ||  
tx1.getText().equals("")){
```

```
                lbl_error.setText("Please fill all  
fields");
```

```
}
```



```
else if(email.equals(""))
{
    lbl_error.setText("NO ACCOUNT
!");
}
else
{
    System.out.println(email);
    lbl_error.setText("INCORRECT
PASSWORD !");
}
this.revalidate();
this.repaint();
this.setVisible(true);
}

}
```

```
if(e.getSource()== b2)
{
```

```
        this.dispose();
        new Sign_Up();
    }
    if(e.getSource()==b4)
    {
        this.dispose();
        new ForgotPassword(Login_Page.maild);
    }

}
```

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

    new Login_Page();
}

}
```

## **MacBook:**

```
package SwingMpr;
```

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import java.awt.event.*;
```

```
import java.awt.event.ActionListener;
```

```
public class MacBook extends JFrame implements  
ActionListener,ItemListener{
```

```

        JLabel
MB_edition,MB_model,MB_memory,MB_color,MB_selecte
d_choice,MB_price;

        JRadioButton
c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13;

        List list;

        ButtonGroup cg,cg1,cg2,cg3;

        JButton b1,b2,b3,b4;

        String model,Mac_color,memory,edition;

        double base_macbook,memory_macbook,price;

        MacBook()
        {
            this.setSize(500,800);

            this.setTitle("Macbook");

            this.setLayout(null);

            this.getContentPane().setBackground(new Color(168,
169, 173));

            MB_edition = new JLabel("Please choose a
edition");

            MB_edition.setBounds(25, 50, 250,40);

            MB_edition.setFont(new
Font("Arial",Font.PLAIN,16));

            this.add(MB_edition);

            list = new List(3,false);

```

```
list.add("M1");  
list.add("Intel i7");  
list.add("Intel i9");  
list.setBounds(25,100,250,20);  
list.addActionListener(this);  
this.add(list);
```

```
MB_model = new JLabel("Choose a model");  
MB_model.setBounds(25,130,250,20);  
MB_model.setFont(new  
Font("Arial",Font.PLAIN,16));  
this.add(MB_model);
```

```
cg = new ButtonGroup();  
c1= new JRadioButton("Air");  
c1.setBounds(25,170,50,12);  
c1.setBackground(new Color(168, 169, 173));  
c1.addItemListener(this);  
this.add(c1);
```

```
c3= new JRadioButton("Pro");  
c3.setBounds(25,210,50,12);
```

```
c3.setBackground(new Color(168, 169, 173));  
c3.addItemListener(this);  
this.add(c3);
```

```
MB_selected_choice= new JLabel();  
MB_selected_choice.setBounds(25,650, 400,20);  
this.add(MB_selected_choice);
```

```
MB_memory=new JLabel("Choose a memory  
option");  
MB_memory.setBounds(25,260,250,20);  
MB_memory.setFont(new  
Font("Arial",Font.PLAIN,16));  
this.add(MB_memory);  
cg2 = new ButtonGroup();  
c5= new JRadioButton("128GB");  
c5.setBounds(25,290,75,12);  
c5.setBackground(new Color(168, 169, 173));
```

```
c5.addItemListener(this);
this.add(c5);

c6= new JRadioButton("256GB");
c6.setBounds(25,320,75,12);
c6.setBackground(new Color(168, 169, 173));
c6.addItemListener(this);
this.add(c6);

c7= new JRadioButton("512GB");
c7.setBounds(25,350,75,12);
c7.setBackground(new Color(168, 169, 173));
c7.addItemListener(this);
this.add(c7);

c8= new JRadioButton("1 TB");
c8.setBounds(25,380,75,12);
c8.setBackground(new Color(168, 169, 173));
c8.addItemListener(this);
this.add(c8);
```

```
MB_color=new JLabel("Choose a Color ");
MB_color.setBounds(25,410, 250,20);

MB_color.setFont(new
Font("Arial",Font.PLAIN,16));
```

```
this.add(MB_color);
cg3 = new ButtonGroup();
c9= new JRadioButton("SILVER");
c9.setBounds(25,440,80,12);
c9.setBackground(new Color(168, 169, 173));
c9.addItemListener(this);
this.add(c9);
c10 = new JRadioButton("ROSE-GOLD");
c10.setBounds(25,470,120,12);
c10.setBackground(new Color(168, 169, 173));
c10.addItemListener(this);
this.add(c10);
c11= new JRadioButton("SPACE GREY");
c11.setBounds(25,500,120,12);
c11.setBackground(new Color(168, 169, 173));
c11.addItemListener(this);
this.add(c11);
//c12= new Checkbox("SPACE GREY",cg3,false);
//c12.setBounds(25,530,80,12);
//c12.addItemListener(this);
//this.add(c12);
```



```
b1= new JButton("DONE");  
b1.setBackground(Color.white);  
b1.setForeground(Color.DARK_GRAY);  
b1.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b1.setBounds(160,570,80,40);  
b1.addActionListener(this);  
this.add(b1);
```

```
b2= new JButton("CART");  
b2.setBackground(Color.BLUE);  
b2.setForeground(Color.WHITE);  
b2.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b2.setBounds(60,570,80,40);  
b2.addActionListener(this);  
this.add(b2);
```

```
b3= new JButton("CLEAR");  
b3.setBackground(Color.GREEN);  
b3.setForeground(Color.WHITE);  
b3.setFont(new  
Font("Arial",Font.BOLD|Font.ITALIC,12));  
b3.setBounds(260,570,80,40);
```

```
        b3.addActionListener(this);
        this.add(b3);

        b4 = new JButton("Back to store");
        b4.setBounds(320,700,100,25);
        b4.setFont(new
Font("Arial",Font.PLAIN|Font.ITALIC,10));
        b4.setForeground(Color.red);
        b4.addActionListener(this);
        this.add(b4);

        this.revalidate();
        this.repaint();
        this.setVisible(true);

    }
```

@Override

```
public void actionPerformed(ActionEvent e) {

    if(e.getSource()==list)
```

```

{
    edition=list.getSelectedItemAt();
    c2= new JRadioButton(edition);
    c2.setBounds(25,190,100,12);
    c2.setBackground(new Color(168, 169, 173));
    c2.addItemListener(this);
    this.add(c2);
}
if(e.getSource()==b1)
{
    if(edition==null || model==null ||
memory==null || Mac_color==null)
    {
        MB_selected_choice.setText("Please
select all the options ");
        //MB_selected_choice.setBounds(25,650,
400,20);
        MB_selected_choice.setFont(new
Font("Arial",Font.PLAIN,16));
        //this.add(MB_selected_choice);
    }
    else

```

```

        { System.out.println("Selected Choice :
Macbook"+" "+edition+" "+model+" "+memory+"
"+Mac_color+price);

        MB_selected_choice.setText("Selected
Choice : Macbook"+" "+edition+" "+model+" "+memory+"
"+Mac_color);

        //MB_selected_choice.setBounds(25,650,
450,20);

        System.out.println("Selected Choice :
Macbook"+" "+edition+" "+model+" "+memory+"
"+Mac_color);

        MB_selected_choice.setFont(new
Font("Arial",Font.PLAIN,16));

        //this.add(MB_selected_choice);

        double mactotal=
base_macbook+memory_macbook;

        getcart ob = new getcart();
        new Cartdummy(price,"add");
        ob.set_Macbook(edition+" "+model+"
"+memory+" "+Mac_color+" ",price);

        MB_price=new JLabel("Price : "+price);
        MB_price.setBounds(25,700, 450,20);
        MB_price.setFont(new
Font("Arial",Font.PLAIN,16));

        this.add(MB_price);

```

```
}  
}
```

```
if(e.getSource()==b2)  
{  
    getcart oc= new getcart();  
    oc.create_Frame(getcart.id, getcart.ap,  
getcart.mc, getcart.ip, getcart.idcost, getcart.ipcost,  
getcart.mccost, getcart.apcost);
```

```
}
```

```
if(e.getSource()==b3)
```

```
{
```

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

```
    new MacBook();
```

```
}
```

```
if(e.getSource()==b4)
```

```
{
```

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

```
    new Store();
```

```
}
```

```
//this.revalidate();
```

```
//this.repaint();
```

```
        //this.setVisible(true);  
  
    }
```

```
@Override  
public void itemStateChanged(ItemEvent e) {  
  
    if(c1.isSelected()==true)  
    {  
  
        model="AIR";  
        base_macbook=70000;  
    }  
    if(c2.isSelected()==true)  
    {  
        model="M1";  
        base_macbook=90000;  
  
    }  
    if(c3.isSelected()==true)
```

```
{  
    model="Pro";  
    base_macbook=100000;  
}
```

```
if(c5.isSelected()==true)  
{
```

```
    memory="128GB";  
    memory_macbook=0;
```

```
}
```

```
if(c6.isSelected()==true)
```

```
{
```

```
    memory="256GB";  
    memory_macbook=5000;
```

```
}
```

```
if(c7.isSelected()==true)
```

```
{
```

```
    memory="512GB";
```

```
memory_macbook=10000;

}
if(c8.isSelected()==true)
{
    memory="1 TB";
    memory_macbook=20000;

}

if(c9.isSelected()==true)
{

    Mac_color="SILVER";

}
if(c10.isSelected()==true)
{
    Mac_color="ROSE-GOLD";

}
if(c11.isSelected()==true)
{
```



```
Mac_color="SPACE GREY";
```

```
}
```

```
//if(c12.getState()==true)
```

```
//{
```

```
//  Mac_color="BLUE";
```

```
//}
```

```
price=base_macbook+memory_macbook;
```

```
}
```

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

```
{
```

```
    new MacBook();
```

```
}
```

```
}
```

**Pay Apple ID:**

```
package SwingMpr;
```

```
import java.util.regex.Pattern;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import javax.swing.*;
```

```
public class Pay_Apple_ID extends JFrame implements
ActionListener
{

    JLabel label_id,label_email,label_upi_id,
label_password,label_confirmpassword,label_heading,label_e
rror,label_error2, label_error3, label_pin;

    JTextField text_id,text_email,text_upi_id;

    JPasswordField text_password, text_confirmpassword,
text_pin;

    JButton b, b2;

    int a;

    String text;

    Pay_Apple_ID()
    {

        this.setLayout(null);

        this.getContentPane().setBackground(Color.black);

        this.setSize(600,600);

        this.setTitle("Newark Store Payment Gateway Part
2");
```

```
        label_heading =new JLabel("Set Up Newark  
Payment",JLabel.CENTER);  
        label_heading.setFont(new  
Font("Tahoma",Font.BOLD,25));  
        label_heading.setForeground(Color.BLACK);  
        label_heading.setBackground(Color.WHITE);  
        label_heading.setBounds(0,25,600,50);  
        label_heading.setOpaque(true);  
        label_heading.setVisible(true);  
  
        this.add(label_heading);  
  
        label_id= new JLabel("Newark Id");  
        label_id.setFont(new  
Font("Tahoma",Font.BOLD,22));  
        label_id.setForeground(Color.white);  
        label_id.setBounds(100,100,150,25);  
        label_id.setVisible(true);  
  
        this.add(label_id);  
  
        text_id = new JTextField(20);  
        text_id.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_id.setBounds(325,100,150,25);
```

```
this.add(text_id);
```

```
label_password= new JLabel("Set Password");
```

```
label_password.setFont(new Font("Times New  
Roman",Font.BOLD,22));
```

```
label_password.setForeground(Color.white);
```

```
label_password.setBounds(100,150,150,25);
```

```
label_password.setVisible(true);
```

```
this.add(label_password);
```

```
text_password = new JPasswordField(20);
```

```
text_password.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_password.setBounds(325,150,150,25);
```

```
//text_password.setEchoChar('*');
```

```
this.add(text_password);
```

```
label_confirmpassword = new JLabel("Confirm  
Password");
```

```
label_confirmpassword.setFont(new Font("Times  
New Roman",Font.BOLD,22));
```

```
label_confirmpassword.setForeground(Color.white);  
label_confirmpassword.setBounds(100,200,225,25);  
label_confirmpassword.setVisible(true);
```

```
this.add(label_confirmpassword);
```

```
text_confirmpassword = new JPasswordField(20);  
text_confirmpassword.setFont(new  
Font("Tahoma",Font.BOLD,18));  
text_confirmpassword.setBounds(325,200,150,25);  
//text_confirmpassword.setEchoChar('*');
```

```
this.add(text_confirmpassword);
```

```
label_pin= new JLabel("Set Pin");  
label_pin.setFont(new Font("Times New  
Roman",Font.BOLD,22));  
label_pin.setForeground(Color.white);  
label_pin.setBounds(100,250,150,25);  
label_pin.setVisible(true);
```

```
this.add(label_pin);
```

```
text_pin = new JPasswordField(20);
```

```
text_pin.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_pin.setBounds(325,250,150,25);
```

```
this.add(text_pin);
```

```
b=new JButton("Register");
```

```
b.setForeground(Color.black);
```

```
b.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));
```

```
b.setBounds(200,300,150,25);
```

```
b.addActionListener(this);
```

```
b.setBackground(Color.white);
```

```
this.add(b);
```

```
label_error= new JLabel();
```

```
label_error.setFont(new Font("Times New  
Roman",Font.ITALIC,22));
```

```
label_error.setForeground(Color.white);
```

```
label_error.setBounds(100,350,450,25);  
label_error.setVisible(true);
```

```
this.add(label_error);
```

```
label_error2 = new JLabel("uppercase letter, one  
lowercase letter, one number and one ");
```

```
label_error2.setFont(new Font("Times New  
Roman",Font.ITALIC,15));
```

```
label_error2.setForeground(Color.white);
```

```
label_error2.setBounds(100,375,450,25);
```

```
label_error3 = new JLabel("special character");
```

```
label_error3.setFont(new Font("Times New  
Roman",Font.ITALIC,15));
```

```
label_error3.setForeground(Color.white);
```

```
label_error3.setBounds(100,400,450,25);
```

```
this.revalidate();
```

```
this.repaint();
```

```
this.setVisible(true);
```

```
}
```



```

@Override

public void actionPerformed(ActionEvent e) {

    if(e.getSource()==b)
    {
        label_error.setText("");
        boolean a=false;
        label_error2.setVisible(false);
        label_error3.setVisible(false);
        if (text_id.getText().equals("") ||
String.valueOf(text_password.getPassword()).equals("") ||
String.valueOf(text_confirmpassword.getPassword()).equals("") ||
String.valueOf(text_pin.getPassword()).equals(""))
        {
            label_error.setText("Please fill all fields");

        }
        else
        if(String.valueOf(text_password.getPassword()).equals(String
.valueOf(text_confirmpassword.getPassword())) &&
!String.valueOf(text_password.getPassword()).equals(""))
        {
            boolean strong_password =
Pattern.matches("(?=[A-Z])(?=.*?[a-z])(?=.*?[0-

```

```

9])(?=.*?[#?!@$%^&*~]).{8,}$",
String.valueOf(text_password.getPassword()));

        if (strong_password)
        {

            text =
String.valueOf(text_pin.getPassword());

            boolean Pin = Pattern.matches("[0-
9]{4}", String.valueOf(text_pin.getPassword()));

            if(Pin)
            {

                create_AppleId(text_id.getText(),
String.valueOf(text_password.getPassword()),
String.valueOf(text_pin.getPassword()));

                text_id.setText("");

                JOptionPane.showMessageDialog(this, "Account
Created Successfully");

                this.dispose();

                new Login_Page();

                a=true;

            }

            else

            {

```

```

        label_error.setText("Invalid Pin.
Enter 4-digit Pin");
    }
}
else{
    label_error.setText("Please enter a
strong 8-digit password containing at least one ");
    label_error.setFont(new Font("Times
New Roman",Font.ITALIC,15));

    label_error2.setVisible(true);
    label_error3.setVisible(true);

    this.add(label_error2);
    this.add(label_error3);

}
}
else
{
    label_error.setText("Password
Mismatch");
}
text_password.setText("");

```

```

        text_confirmpassword.setText("");
        text_pin.setText("");
        if(a==false) {
            this.revalidate();
            this.repaint();
            this.setVisible(true);
        }
    }

    public static void create_AppleId(String AppleId,String
    Password, String PIN)
    {
        FirstJDBC obc = new FirstJDBC();
        Connection con =obc.connectDB();

        PreparedStatement p = null;

        String q ="INSERT INTO
appleid_database(Apple_ID,Apple_password,Newark_pin,em
ail_id) "

            + "VALUES(?,?,?,?)";

        //ResultSet rs = null;

        try
        {

```

```
p = con.prepareStatement(q);
```

```
p.setString(1,AppleId);
```

```
    p.setString(2>Password);
```

```
    p.setString(3,PIN);
```

```
    p.setString(4,Sign_Up.registration_mail);
```

```
p.executeUpdate();
```

```
}
```

```
catch( Exception sql)
```

```
{
```

```
    sql.printStackTrace();
```

```
}
```

```
}
```

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

```
    // TODO Auto-generated method stub
```

```
    new Pay_Apple_ID();
```

```
}
```

```
}
```

### **Pay Sign Up:**

```
package SwingMpr;

import java.util.regex.Pattern;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.*;
```

```
public class Pay_Sign_Up extends JFrame implements  
ActionListener
```

```
{
```

```
    JLabel label_acctname,label_email,label_upi_id,  
    label_password,label_upi_pass,label_heading,label_error,label_error2, label_error3, label_pin;
```

```
    JTextField text_acctname,text_email,text_upi_id;
```

```
    JPasswordField text_password, text_upi_pass, text_pin;
```

```
    JButton b, b2, b3;
```

```
    int a;
```

```
    String text;
```

```
Pay_Sign_Up()
```

```
{
```

```
    this.setLayout(null);
```

```
    this.getContentPane().setBackground(Color.black);
```

```
    this.setSize(700,600);
```

```
this.setTitle("Newark Store Payment Gateway");  
this.setVisible(true);
```

```
label_heading =new JLabel("Set Up Newark  
Payment",JLabel.CENTER);
```

```
label_heading.setFont(new  
Font("Tahoma",Font.BOLD,25));
```

```
label_heading.setForeground(Color.BLACK);
```

```
label_heading.setBackground(Color.WHITE);
```

```
label_heading.setBounds(0,25,700,50);
```

```
label_heading.setVisible(true);
```

```
label_heading.setOpaque(true);
```

```
this.add(label_heading);
```

```
label_acctname= new JLabel("Account Name");
```

```
label_acctname.setFont(new  
Font("Tahoma",Font.BOLD,22));
```

```
label_acctname.setForeground(Color.white);
```

```
label_acctname.setBounds(100,100,200,25);
```

```
label_acctname.setVisible(true);
```

```
this.add(label_acctname);
```



```
text_acctname = new JTextField(20);
text_acctname.setFont(new
Font("Tahoma",Font.BOLD,18));
text_acctname.setBounds(325,100,150,25);

this.add(text_acctname);

/*label_email= new JLabel("Email");
label_email.setFont(new Font("Times New
Roman",Font.BOLD,22));
label_email.setForeground(Color.white);
label_email.setBounds(100,150,150,25);
label_email.setVisible(true);

this.add(label_email);

text_email = new JTextField(20);
text_email.setFont(new
Font("Tahoma",Font.BOLD,18));
text_email.setBounds(325,150,150,25);

this.add(text_email);*/

label_upi_id = new JLabel("UPI id");
```

```
label_upi_id.setFont(new Font("Times New  
Roman",Font.BOLD,22));
```

```
label_upi_id.setForeground(Color.white);
```

```
label_upi_id.setBounds(100,200,225,25);
```

```
label_upi_id.setVisible(true);
```

```
this.add(label_upi_id);
```

```
text_upi_id = new JTextField(20);
```

```
text_upi_id.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_upi_id.setBounds(325,200,150,25);
```

```
this.add(text_upi_id);
```

```
label_password= new JLabel("Account Password");
```

```
label_password.setFont(new Font("Times New  
Roman",Font.BOLD,22));
```

```
label_password.setForeground(Color.white);
```

```
label_password.setBounds(100,150,200,25);
```

```
label_password.setVisible(true);
```

```
this.add(label_password);
```

```
text_password = new JPasswordField(20);
text_password.setFont(new
Font("Tahoma",Font.BOLD,18));
text_password.setBounds(325,150,150,25);
//text_password.setEchoChar('*');

this.add(text_password);

label_upi_pass = new JLabel("UPI PIN");
label_upi_pass.setFont(new
Font("Serif",Font.BOLD,22));
label_upi_pass.setForeground(Color.white);
label_upi_pass.setBounds(100,250,200,25);
label_upi_pass.setVisible(true);

this.add(label_upi_pass);

text_upi_pass = new JPasswordField(20);
text_upi_pass.setFont(new
Font("Tahoma",Font.BOLD,18));
text_upi_pass.setBounds(325,250,150,25);
text_upi_pass.setEchoChar('*');
```

```
this.add(text_upi_pass);

/*label_pin = new JLabel("Email Code");
label_pin.setFont(new
Font("Serif",Font.BOLD,22));
label_pin.setForeground(Color.white);
label_pin.setBounds(100,350,200,25);
label_pin.setVisible(true);

this.add(label_pin);

text_pin = new JTextField(20);
text_pin.setFont(new
Font("Tahoma",Font.BOLD,18));
text_pin.setBounds(325,350,150,25);

this.add(text_pin);

/*b2=new JButton("Verify Email");
b2.setForeground(Color.white);
b2.setFont(new
Font("Serif",Font.BOLD|Font.ITALIC,20));
b2.setBounds(480,150,150,25);
b2.addActionListener(this);
```

```
this.add(b2);*/
```

```
b=new JButton("Register");  
b.setForeground(Color.white);  
b.setBackground(Color.black);  
b.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
b.setBounds(200,400,150,25);  
b.setOpaque(true);  
b.setVisible(true);  
b.addActionListener(this);  
this.add(b);  
  
label_error= new JLabel();  
label_error.setFont(new Font("Times New  
Roman",Font.ITALIC,22));  
label_error.setForeground(Color.white);  
label_error.setBounds(100,450,450,25);  
label_error.setVisible(true);  
  
this.add(label_error);
```

```
        label_error2 = new JLabel("uppercase letter, one  
lowercase letter, one number and one ");
```

```
        label_error2.setFont(new Font("Times New  
Roman",Font.ITALIC,15));
```

```
        label_error2.setForeground(Color.white);
```

```
        label_error2.setBounds(100,475,450,25);
```

```
        label_error3 = new JLabel("special character");
```

```
        label_error3.setFont(new Font("Times New  
Roman",Font.ITALIC,15));
```

```
        label_error3.setForeground(Color.white);
```

```
        label_error3.setBounds(100,500,450,25);
```

```
        this.revalidate();
```

```
        this.repaint();
```

```
        this.setVisible(true);
```

```
    }
```

```
@Override
```

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

```
    if(e.getSource()==b)
```

```
    {
```

```

        boolean a=false;

        if (text_acctname.getText().equals("") ||
String.valueOf(text_password.getPassword()).equals("") ||
String.valueOf(text_upi_pass.getPassword()).equals("") ||
String.valueOf(text_upi_id.getText()).equals(""))
        {
            label_error.setText("Please fill all fields");
            JOptionPane.showMessageDialog(this,
"Please fill all fields");
        }
        else
        {
            String name="";
            String password="";
            String upi_pin="";
            String upi_id="";
            double balance=0;

            FirstJDBC ob = new FirstJDBC();
            String s= text_acctname.getText();

            Connection con = null;
            PreparedStatement p = null;
            ResultSet rs = null;

            con = ob.connectDB();

```

```

        try
        {
            String sql = "select * from
bank_database where Account_name='"+s+"'";
//            System.out.println(sql);
            p = con.prepareStatement(sql);
            rs= p.executeQuery();

            if(rs.next())
            {
                name =
rs.getString("Account_name");
                password =
rs.getString("account_password");
//            System.out.println(password);
                upi_id=rs.getString("upi_id");
                upi_pin =
rs.getString("upi_pin");
                System.out.println(upi_pin);
                balance =
rs.getDouble("Balance");
            }

```



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

```
    if(name!="")  
    {
```

```
        if(password.equals(String.valueOf(text_password.getPas  
sword()))) &&  
        upi_id.equals(String.valueOf(text_upi_id.getText())) &&  
        upi_pin.equals(String.valueOf(text_upi_pass.getPassword()))
```

```
    {
```

```
//
```

```
    System.out.println("Verified"+balance);
```

```
        label_error.setText("Verified");
```

```
    JOptionPane.showMessageDialog(this, "Verified");
```

```

        this.dispose();
        new Pay_Apple_ID();
        a=true;
    }
    else
    {

//                System.out.println("Recheck your
credential");

                label_error.setText("Recheck your
credential");

                System.out.println(name);

        JOptionPane.showMessageDialog(this, "Recheck your
credential");

    }

}

    else
    {

//                System.out.println("No bank account
found contact your bank");

                label_error.setText("No bank
account found contact your bank");

```

```
JOptionPane.showMessageDialog(this, "No bank  
account found. Contact your bank");
```

```
}
```

```
}
```

```
if (a==false) {
```

```
    this.revalidate();
```

```
    this.repaint();
```

```
    this.setVisible(true);
```

```
}
```

```
}
```

```
}
```

```
//public boolean(String acc_name, String , String c,  
String d , )
```

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

```
    // TODO Auto-generated method stub
```

```
    new Pay_Sign_Up();
```

```
}
```

}

### **ResetPassword:**

```
package SwingMpr;
```

```
import javax.swing.*;
```

```
import java.awt.Color;
```

```
import java.awt.Font;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.SQLIntegrityConstraintViolationException;
```

```
import java.util.regex.Pattern;
```

```
public class ResetPassword extends JFrame implements  
ActionListener {  
  
    JLabel  
password_f,password_en,password_v,label_error4,label_error  
,label_error2,label_error3;  
  
    JPasswordField text_password_en,text_password_v;  
  
    JButton b2;  
  
    String y;  
  
    public static String email="";  
  
    ResetPassword()  
    {  
  
        this.setLayout(null);  
  
        this.getContentPane().setBackground(Color.black);  
  
        this.setSize(800,600);  
  
        this.setTitle("Forget Password");  
  
  
        password_f =new JLabel("Reset Password",  
SwingConstants.CENTER);  
  
        password_f.setFont(new  
Font("Tahoma",Font.BOLD,25));  
  
        password_f.setForeground(Color.BLACK);
```

```
password_f.setBackground(Color.WHITE);  
password_f.setBounds(0,50,800,50);  
password_f.setOpaque(true);  
password_f.setVisible(true);
```

```
this.add(password_f);
```

```
password_en =new JLabel("New Password",  
SwingConstants.LEFT);
```

```
password_en.setFont(new  
Font("Tahoma",Font.BOLD,20));
```

```
password_en.setForeground(Color.white);  
password_en.setBounds(50,160,200,50);  
password_en.setVisible(true);
```

```
this.add(password_en);
```

```
text_password_en = new JPasswordField();
```

```
text_password_en.setFont(new  
Font("TimesRoman",Font.BOLD,24));
```

```
text_password_en.setBounds(275,170,250,35);
```

```
this.add(text_password_en);
```

```
password_v = new JLabel("Confirm Password",  
SwingConstants.LEFT);
```

```
password_v.setFont(new  
Font("Tahoma",Font.BOLD,20));
```

```
password_v.setForeground(Color.white);
```

```
password_v.setBounds(50,230,200,50);
```

```
password_v.setVisible(true);
```

```
this.add(password_v);
```

```
text_password_v = new JPasswordField();
```

```
text_password_v.setFont(new  
Font("TimesRoman",Font.BOLD,24));
```

```
text_password_v.setBounds(275,230,250,35);
```

```
this.add(text_password_v);
```

```
label_error4 = new JLabel();
```

```
label_error4.setFont(new  
Font("Tahoma",Font.BOLD,25));
```

```
label_error4.setForeground(Color.white);
```

```
label_error4.setBounds(50,380,400,50);
```

```
label_error4.setVisible(true);
```

```
this.add(label_error4);
```

```
    b2=new JButton("Proceed");  
    b2.setForeground(Color.white);  
    b2.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
    b2.setBounds(575,280,150,25);  
    b2.addActionListener(this);  
    b2.setBackground(Color.black);  
    b2.setOpaque(true);  
    b2.setVisible(true);  
    this.add(b2);
```

```
    this.revalidate();  
    this.repaint();  
    this.setVisible(true);
```

```
}
```

```
@Override
```

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



```

        if(e.getSource()==b2)
        {

            if(String.valueOf(text_password_en.getPassword()).equals(String.valueOf(text_password_v.getPassword()))){

                boolean strong_password =
                Pattern.matches("^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*~]).{8,}$",
                String.valueOf(text_password_en.getPassword()));

                if (strong_password){

                    label_error4.setVisible(false);

                    reset(String.valueOf(text_password_v.getPassword()));

                    this.dispose();

                    new Login_Page();

                }

                else{

                    label_error4.setVisible(true);

                    label_error4.setText("Please enter a strong
password");

                    text_password_en.setText("");
                    text_password_v.setText("");

                    this.revalidate();

                    this.repaint();

```

```

        this.setVisible(true);
    }
}

// if
(!String.valueOf(text_password_en.getPassword()).equals(String.valueOf(text_password_v.getPassword())))
    else{
        label_error4.setVisible(true);
        label_error4.setText("Password
Mismatch");

        text_password_en.setText("");
        text_password_v.setText("");
        this.revalidate();
        this.repaint();
        this.setVisible(true);
    }

}

}

public static void reset(String pass)
{
    FirstJDBC obc = new FirstJDBC();

```

```
Connection con =obc.connectDB();
```

```
PreparedStatement p = null;
```

```
String q ="UPDATE customer_id SET password=?  
where email_id=?";
```

```
//ResultSet rs = null;
```

```
try
```

```
{
```

```
p = con.prepareStatement(q);
```

```
p.setString(1,pass);
```

```
p.setString(2,ForgotPassword.reset_email);
```

```
p.executeUpdate();
```

```
}
```

```
/*catch (SQLException
```

o)

```
{
```

```
Sign_Up l = new Sign_Up() ;
```

```
l.errorofrepeat();
```

```
        // TODO: handle exception
    }*/
    catch(Exception sql)
    {
        sql.printStackTrace();
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        new ResetPassword();

    }

}
```

### **Sign Up:**

```
package SwingMpr;
```

```
import java.util.regex.Pattern;
```

```
import javax.swing.*;
```

```
import studyJDBC.FirstJDBC;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLIntegrityConstraintViolationException;
import java.sql.Statement;
```

```
public class Sign_Up extends JFrame implements
ActionListener
```

```
{
```

```
    JLabel label_username,label_email,label_mobilenumber,
label_password,label_confirmpassword,label_heading,label_e
rror,label_error2, label_error3,label_error4,
label_pin,street,city,state,zipcode;
```

```
    JTextField
text_username,text_email,text_mobilenumber,text_street,text_
city,text_state,text_zipcode;
```

```
    JPasswordField text_password, text_confirmpassword,
text_pin;
```

```
    JButton b,b2,b3,b4,b5,b6;
```

```
    int a;
```

```
    String text;
```

```
    public String y;
```

```
    public static String registration_mail;
```

```
    Emailsend u = new Emailsend();
```

```
    Sign_Up()
```

```
{
```

```
this.setLayout(null);  
this.getContentPane().setBackground(Color.black);  
this.setSize(700, 725);  
this.setTitle("Sign Up");
```

```
    label_heading =new JLabel("Newark Store Sign  
Up",SwingConstants.CENTER);
```

```
        label_heading.setFont(new  
Font("Tahoma",Font.BOLD,25));
```

```
        label_heading.setForeground(Color.BLACK);
```

```
        label_heading.setBackground(Color.WHITE);
```

```
        label_heading.setBounds(0,15,800,50);
```

```
        label_heading.setOpaque(true);
```

```
        label_heading.setVisible(true);
```

```
    this.add(label_heading);
```

```
    label_username= new JLabel("Username");
```

```
        label_username.setFont(new  
Font("Tahoma",Font.BOLD,22));
```

```
        label_username.setForeground(Color.white);
```

```
        label_username.setBounds(100,100,150,25);
```

```
        label_username.setVisible(true);
```

```
this.add(label_username);
```

```
text_username = new JTextField(20);
```

```
text_username.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_username.setBounds(325,100,150,25);
```

```
this.add(text_username);
```

```
label_email= new JLabel("Email");
```

```
label_email.setFont(new Font("Times New  
Roman",Font.BOLD,22));
```

```
label_email.setForeground(Color.white);
```

```
label_email.setBounds(100,150,150,25);
```

```
label_email.setVisible(true);
```

```
this.add(label_email);
```

```
text_email = new JTextField(20);
```

```
text_email.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_email.setBounds(325,150,150,25);
```

```
this.add(text_email);
```



```
label_pin = new JLabel("Set 4-digit PIN");  
label_pin.setFont(new  
Font("Serif",Font.BOLD,22));  
label_pin.setForeground(Color.white);  
label_pin.setBounds(100,200,200,25);  
label_pin.setVisible(true);  
//label_pin.setBounds(100,350,200,25);  
//label_pin.setVisible(true);
```

```
this.add(label_pin);
```

```
text_pin = new JPasswordField(20);  
text_pin.setFont(new  
Font("Tahoma",Font.BOLD,18));  
text_pin.setBounds(325,200,150,25);
```

```
label_mobilenumber = new JLabel("Mobile  
Number");  
label_mobilenumber.setFont(new Font("Times New  
Roman",Font.BOLD,22));  
label_mobilenumber.setForeground(Color.white);  
label_mobilenumber.setBounds(100,250,175,25);
```

```
label_mobilenumber.setVisible(true);
```

```
this.add(label_mobilenumber);
```

```
text_mobilenumber = new JTextField(20);
```

```
text_mobilenumber.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_mobilenumber.setBounds(325,250,150,25);
```

```
this.add(text_mobilenumber);
```

```
label_password= new JLabel("Password");
```

```
label_password.setFont(new Font("Times New  
Roman",Font.BOLD,22));
```

```
label_password.setForeground(Color.white);
```

```
label_password.setBounds(100,300,150,25);
```

```
label_password.setVisible(true);
```

```
this.add(label_password);
```

```
text_password = new JPasswordField(20);
```

```
text_password.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_password.setBounds(325,300,150,25);  
text_password.setEchoChar('*');
```

```
this.add(text_password);
```

```
label_confirmpassword = new JLabel("Confirm  
Password");
```

```
label_confirmpassword.setFont(new  
Font("Serif",Font.BOLD,22));
```

```
label_confirmpassword.setForeground(Color.white);
```

```
label_confirmpassword.setBounds(100,350,200,25);
```

```
label_confirmpassword.setVisible(true);
```

```
this.add(label_confirmpassword);
```

```
text_confirmpassword = new JPasswordField(20);
```

```
text_confirmpassword.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_confirmpassword.setBounds(325,350,150,25);
```

```
text_confirmpassword.setEchoChar('*');
```

```
this.add(text_confirmpassword);
```

```
street= new JLabel("Flat no/street");  
street.setFont(new Font("Serif",Font.BOLD,22));  
street.setForeground(Color.white);  
street.setBounds(100,400,200,25);  
street.setVisible(true);
```

```
this.add(street);
```

```
text_street = new JTextField(20);  
text_street.setFont(new  
Font("Tahoma",Font.BOLD,18));  
text_street.setBounds(325,400,150,25);
```

```
this.add(text_street);
```

```
city= new JLabel("City");  
city.setFont(new Font("Serif",Font.BOLD,22));  
city.setForeground(Color.white);  
city.setBounds(100,450,200,25);  
city.setVisible(true);
```

```
this.add(city);
```

```
text_city = new JTextField(20);
```

```
text_city.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_city.setBounds(325,450,150,25);
```

```
this.add(text_city);
```

```
state = new JLabel("State");
```

```
state.setFont(new Font("Serif",Font.BOLD,22));
```

```
state.setForeground(Color.white);
```

```
state.setBounds(100,500,200,25);
```

```
state.setVisible(true);
```

```
this.add(state);
```

```
text_state = new JTextField(20);
```

```
text_state.setFont(new  
Font("Tahoma",Font.BOLD,18));
```

```
text_state.setBounds(325,500,150,25);
```

```
this.add(text_state);
```

```
zipcode= new JLabel("Zipcode");  
zipcode.setFont(new Font("Serif",Font.BOLD,22));  
zipcode.setForeground(Color.white);  
zipcode.setBounds(100,550,200,25);  
zipcode.setVisible(true);
```

```
this.add(zipcode);
```

```
text_zipcode= new JTextField(20);  
text_zipcode.setFont(new  
Font("Tahoma",Font.BOLD,18));  
text_zipcode.setBounds(325,550,150,25);  
  
this.add(text_zipcode);
```

```
b2=new JButton("Verify Email");  
b2.setForeground(Color.white);  
b2.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));
```

```
b2.setBounds(500,150,150,25);  
b2.addActionListener(this);
```

```
this.add(b2);
```

```
this.add(text_pin);
```

```
b=new JButton("Register");  
b.setForeground(Color.white);  
b.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
b.setBounds(200,600,100,25);  
b.addActionListener(this);  
this.add(b);
```

```
b4=new JButton("NEXT");  
b4.setBackground(Color.white);  
b4.setForeground(Color.black);  
b4.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));  
b4.setBounds(500,650,300,25);
```

```
b4.addActionListener(this);
```

```
b4.setVisible(false);
```

```
this.add(b4);
```

```
b5=new JButton("Go to LOGIN");
```

```
b5.setForeground(Color.white);
```

```
b5.setBackground(Color.black);
```

```
b5.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));
```

```
b5.setBounds(500,600,200,25);
```

```
b5.addActionListener(this);
```

```
b5.setVisible(false);
```

```
this.add(b5);
```

```
b6=new JButton("Go to hell");
```

```
b6.setForeground(Color.white);
```

```
b6.setBackground(Color.black);
```

```
b6.setFont(new  
Font("Serif",Font.BOLD|Font.ITALIC,20));
```

```
b6.setBounds(500,600,200,25);
```

```
b6.addActionListener(this);
```

```
b6.setVisible(false);
```

```
this.add(b6);
```



```
label_error= new JLabel();  
label_error.setFont(new Font("Times New  
Roman",Font.ITALIC,22));  
label_error.setForeground(Color.white);  
label_error.setBounds(100,650,450,25);  
label_error.setVisible(true);  
  
this.add(label_error);  
  
label_error2 = new JLabel("uppercase letter, one  
lowercase letter, one number and one ");  
label_error2.setFont(new Font("Times New  
Roman",Font.ITALIC,15));  
label_error2.setForeground(Color.white);  
label_error2.setBounds(100,700,450,25);  
  
label_error3 = new JLabel("special character");  
label_error3.setFont(new Font("Times New  
Roman",Font.ITALIC,15));  
label_error3.setForeground(Color.white);  
label_error3.setBounds(100,750,450,25);
```

```
        label_error4 = new JLabel("Please enter your  
email");
```

```
        label_error4.setFont(new Font("Times New  
Roman",Font.ITALIC,15));
```

```
        label_error4.setForeground(Color.white);
```

```
        label_error4.setBounds(500,250,450,25);
```

```
        label_error4.setVisible(false);
```

```
        this.add(label_error4);
```

```
        this.revalidate();
```

```
        this.repaint();
```

```
        this.setVisible(true);
```

```
    }
```

```
@Override
```

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

```
    if(e.getSource()==b)
```

```
    {
```

```
        boolean abcd=false;
```

```

        System.out.println("hey1");
        label_error2.setVisible(false);
        label_error3.setVisible(false);

        if (text_username.getText().equals("") ||
text_email.getText().equals("") ||
String.valueOf(text_password.getPassword()).equals("") ||
String.valueOf(text_confirmpassword.getPassword()).equals("
") || String.valueOf(text_pin.getPassword()).equals("") ||
text_mobilenumber.getText().equals(""))
        {
            label_error.setText("Please fill all fields");
        }
        else
        if(String.valueOf(text_password.getPassword()).equals((String.
valueOf(text_confirmpassword.getPassword())))) &&
!text_password.getPassword().toString().equals(""))
        {

            System.out.println(String.valueOf(text_password.getPass
word()));

            System.out.println(text_password.getPassword());

            System.out.println(text_password.getPassword().toString
());

```

```

        boolean strong_password =
Pattern.matches("(^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-
9])(?=.*?[#?!@$%^&*~]).{8,}$",
String.valueOf(text_password.getPassword()));

        if (strong_password)
        {

            text =
String.valueOf(text_pin.getPassword());

            boolean Mob_No =
Pattern.matches("[0-9]{10}",text_mobilenumber.getText());

            if(Mob_No)
            {
                try {

                    a = Integer.parseInt(text);

                    // OK, valid number.

                    if(1000<=a && a<10000)
                    {

                        boolean isemail =
Pattern.matches("^[A-Z0-9._%+-]+@[A-Z0-9.-]+{2,6}$",
(String.valueOf(text_email.getText())).toUpperCase());

                        if(isemail){

```

```
        if(u.verified(String.valueOf(text_pin.getPassword()))==0
    )

        {

            System.out.print(String.valueOf(u.verified(String.valueOf(
text_pin.getPassword()))));

            create_new(text_username.getText(),text_mobilenumber.
getText(),text_email.getText(),text_street.getText(),text_city.g
etText(),text_state.getText(),text_zipcode.getText(),String.val
ueOf(text_confirmpassword.getPassword()));

            Sign_Up.registration_mail=text_email.getText();

            label_error.setText("Account Created Successfully");

            text_username.setText("");

text_email.setText("");

            text_mobilenumber.setText("");

            text_state.setText("");

            text_street.setText("");
```

```
text_city.setText(" ");
```

```
text_zipcode.setText(" ");
```

```
b.setVisible(false);
```

```
b4.setVisible(true);
```

```
abcd=true;
```

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

```
new
```

```
Pay_Sign_Up();
```

```
// Add in
```

```
Database
```

```
}
```

```
}
```

```
else
```

```
{
```

```
label_error.setText("Invalid Email");
```

```
}
```

```
}
```

```
else
```

```

        {

            label_error.setText("Invalid Pin");

        }

    } catch
(NumberFormatException nfe) {

        label_error.setText("Invalid PIN");

        // Not a number.

    }

    }

    else{

        label_error.setText("Please
Enter Valid Mobile Number");

    }

    }

    else{

        label_error.setText("Please enter a
strong 8-digit password containing at least one ");

        label_error.setFont(new Font("Times
New Roman",Font.ITALIC,15));

        label_error2.setVisible(true);
        label_error3.setVisible(true);

```

```

        this.add(label_error2);
        this.add(label_error3);

    }

}

else
{
    label_error.setText("Password
Mismatch");
}

text_password.setText("");
text_confirmpassword.setText("");
text_pin.setText("");
if(abcd==false) {
    this.revalidate();
    this.repaint();
    this.setVisible(true);
}

}

// handle null pointer exception here
if(e.getSource()==b2)
{

```



```

        if(text_email.getText().equals(""))==true)
    {
        System.out.println("hey");
        label_error4.setVisible(true);
        label_error4.setText("Enter your email");
    }
else
{
    label_error4.setVisible(false);
    int l=u.generate();
    System.out.println(l);
    y=text_email.getText();
    try {
        sendmail(y,l);
    } catch (Exception e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}

```

```

this.revalidate();
this.repaint();
this.setVisible(true);

```

```
}  
if(e.getSource()==b4)  
{  
    this.dispose();  
    new Pay_Sign_Up();  
}  
if(e.getSource()==b5)  
{  
    this.dispose();  
    new Login_Page();  
}  
if(e.getSource()==b6)  
{  
    this.dispose();  
    new Pay_Sign_Up();  
}
```

```
}  
  
public static void sendmail(String y,int l) throws  
Exception{
```

```

        Emailsend.sendMail(y,l);
        System.out.println("Hw");

    }

    public static void create_new(String name,String
mobile_no, String email, String flat,String city,String
stt,String pincode, String pasword)
    {
        FirstJDBC obc = new FirstJDBC();
        Connection con =obc.connectDB();

        PreparedStatement p = null;

        String q ="INSERT INTO
customer_id(username,phone,email_id,street,city
,state,zip_code,password) "
            + "VALUES(?,?,?,?,?,?,?,?)";

        //ResultSet rs = null;

        try
        {

            p = con.prepareStatement(q);

            p.setString(1,name);

```

```

        p.setString(2, mobile_no);
        p.setString(3, email);
        p.setString(4, flat);
        p.setString(5, city);
        p.setString(6, stt);
        p.setString(7, pincode);
        p.setString(8, password);
    p.executeUpdate();
}
catch (SQLException o)
{
    Sign_Up l = new Sign_Up() ;
    l.errorofrepeat();

    // TODO: handle exception
}
catch(SQLException sql)
{
    sql.printStackTrace();
}

public void errorofrepeat()
{

```

```
        label_error.setText("ACCOUNT ALREADY  
PRESENT");
```

```
        b5.setVisible(true);
```

```
    }
```

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

```
{
```

```
    new Sign_Up();
```

```
}
```

```
}
```

### **Store:**

```
package SwingMpr;
```

```
import java.awt.*;
```

```
import java.util.*;
```

```
import java.awt.event.*;
```

```
import javax.swing.*;
```

```
public class Store extends JFrame implements ActionListener
```

```
{
```

```
    JLabel heading,items;
```

```
 JButton b1,b2,b3,b4,b5,b6,b7,b8;
```

```
 public static boolean cart_visible=false;
```

```
 Store()
```

```
 {
```

```
     this.setSize(450,700);
```

```
     this.setLayout(null);
```

```
     this.setTitle("Welcome Page");
```

```
     this.getContentPane().setBackground(Color.gray);
```

```
     this.setVisible(true);
```

```
 this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
 heading = new JLabel("Welcome ");
```

```
 heading.setBounds(150,50,250,50);
```

```
 heading.setFont(new Font("Arial",Font.BOLD,16));
```

```
 this.add(heading);
```

```
 items = new JLabel("What are you looking for");
```

```
items.setBounds(25,150,250,50);  
items.setFont(new Font("Arial",Font.BOLD,16));  
this.add(items);
```

```
b1= new JButton("Iphone");  
b1.setBounds(150,200,150,45);  
b1.setFont(new Font("Arial",Font.BOLD,16));  
b1.setBackground(Color.black);  
b1.setForeground(Color.white);  
b1.addActionListener(this);  
this.add(b1);
```

```
b2= new JButton("Ipad");  
b2.setBounds(150,260,150,45);  
b2.setFont(new Font("Arial",Font.BOLD,16));  
b2.setBackground(Color.black);  
b2.setForeground(Color.white);  
b2.addActionListener(this);  
this.add(b2);
```

```
b3= new JButton("MacBook");  
b3.setBounds(150,320,150,45);
```

```
b3.setFont(new Font("Arial",Font.BOLD,16));  
b3.setBackground(Color.black);  
b3.setForeground(Color.white);  
b3.addActionListener(this);  
this.add(b3);
```

```
b4= new JButton("AirPods");  
b4.setBounds(150,380,150,45);  
b4.setFont(new Font("Arial",Font.BOLD,16));  
b4.setBackground(Color.black);  
b4.setForeground(Color.white);  
b4.addActionListener(this);  
this.add(b4);
```

```
b5= new JButton("View Cart");  
b5.setBounds(150,440,150,45);  
b5.setFont(new Font("Arial",Font.BOLD,16));  
b5.setBackground(Color.black);  
b5.setForeground(Color.white);  
b5.addActionListener(this);  
this.add(b5);
```

```
b6= new JButton("CHAT");
```



```
b6.setBounds(150,500,150,45);  
b6.setFont(new Font("Arial",Font.BOLD,16));  
b6.setBackground(Color.black);  
b6.setForeground(Color.white);  
b6.addActionListener(this);
```

```
b7= new JButton("Pay");  
b7.setBounds(150,560,150,45);  
b7.setFont(new Font("Arial",Font.BOLD,16));  
b7.setBackground(Color.black);  
b7.setForeground(Color.white);  
b7.addActionListener(this);
```

```
this.add(b6);  
this.add(b7);
```

```
this.revalidate();  
    this.repaint();  
    this.setVisible(true);  
/*b8= new Button("Pay");  
b8.setBounds(150,560,150,45);
```

```
b8.setFont(new Font("Arial",Font.BOLD,16));
b8.setBackground(Color.black);
b8.setForeground(Color.white);
b8.addActionListener(this);*/
}
```

```
public static void main(String[] args) {
    new Store();
}
```

@Override

```
public void actionPerformed(ActionEvent e) {
    if(e.getSource()==b1)
    {
        this.dispose();
        new Iphone();

    }

    if(e.getSource()==b2)
    {    this.dispose();
        new Ipad();
    }
}
```

```
if(e.getSource()==b3)
{
    this.dispose();
        new MacBook();
}
if(e.getSource()==b4)
{
    this.dispose();
        new Airpod();
}
if(e.getSource()==b5)
{
    this.dispose();
        getcart oc= new getcart();
        oc.create_Frame(getcart.id, getcart.ap,
getcart.mc, getcart.ip, getcart.idcost, getcart.ipcost,
getcart.mccost, getcart.apcost);
}
if(e.getSource()==b6)
{
    this.dispose();
        new Chatbox();
}
if(e.getSource()==b7)
{
    this.dispose();
        new Apple_Pay(Cartdummy.gtotal);
}
```

```
}
```

```
}
```

### **Thank you:**

```
package SwingMpr;
```

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import java.awt.event.*;
```

```
import java.awt.event.ActionListener;
```

```
public class Thank_you extends JFrame implements  
ActionListener {
```

```
    JLabel l1;
```

```
    JButton b1,b2;
```

```
Thank_you()
```

```
{
```

```
    this.setSize(550,450);
```

```
    this.setLayout(null);
```

```
this.getContentPane().setBackground(Color.orange);
```

```
    l1 =new JLabel("THANK YOU FOR  
SHOPPING WITH US");
```

```
    l1.setFont(new  
Font("Tahoma",Font.BOLD,25));
```

```
    l1.setForeground(Color.BLACK);
```

```
    l1.setBackground(Color.white);
```

```
    l1.setBounds(30,125,550,50);
```

```
    l1.setVisible(true);
```

```
    this.add(l1);
```

```
        b2 = new JButton("EXIT");
        b2.setFont(new
Font("Tahoma",Font.BOLD,15));
        b2.setForeground(Color.RED);
        b2.setBackground(Color.black);
        b2.setBounds(225,250,100,20);
        b2.addActionListener(this);;
        this.add(b2);

        this.revalidate();
        this.repaint();
        this.setVisible(true);
```

```
}
```

@Override

```
public void actionPerformed(ActionEvent e) {  
    if(e.getSource()==b2)  
    {  
        System.exit(0);    }  
}
```

```
public static void main(String[] args) {  
    new Thank_you();  
}  
  
}
```