JAVA CAT-3

|  |
| --- |
| NAME:**S.HARIHARAN**  ROLLNO:**1831017**  DEPARTMENT:**COMPUTING**  **(SOFTWARE SYSTEMS)**  YEAR:**II YEAR** |
| PROJECT TITLE:**GENERATION OF VEHICLE MOTOR INSURANCE** |

**PROJECT AIM:**

To generate a vehicle motor insurance policy to the customer’s and Manager can view the Policy purchased details.

**PROJECT DESCRIPTION:**

As the Extensive growth of vehicle in all over the world they are not bothering about the vehicle accidents.If the Vehicle accident occurs then the owner need not pay the amount where he could claim for that vehicle using the motor insurance policy.This project is a Stand alone one which can be used by the employee of Motor insurance Company.For future enhancement I would like to add the renew the policy which has been generated and make a message to the policy holders who are all nearing the expiry date of Policy.

**USAGE OF DIFFERENT TYPES OF CONTROL:**

1) jTextBox :



2)jButton:



3)jMenuBar:



1. jLabel:



1. jComboBox:



1. jForamtTextField:

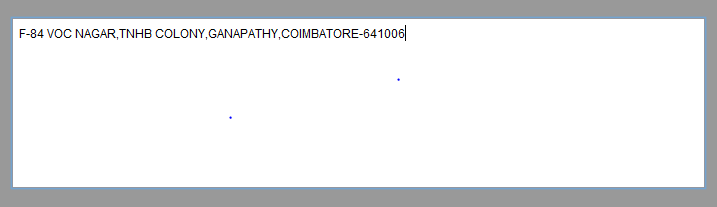




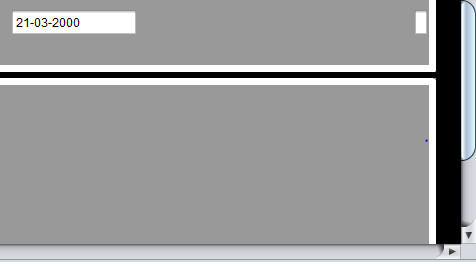
1. jCheckBox:



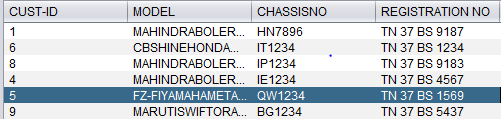
1. jTextArea:



1. jScrollBar:



10)jTable:



11)jPasswordField:



DATABASE CONNECTIVITY CODE:

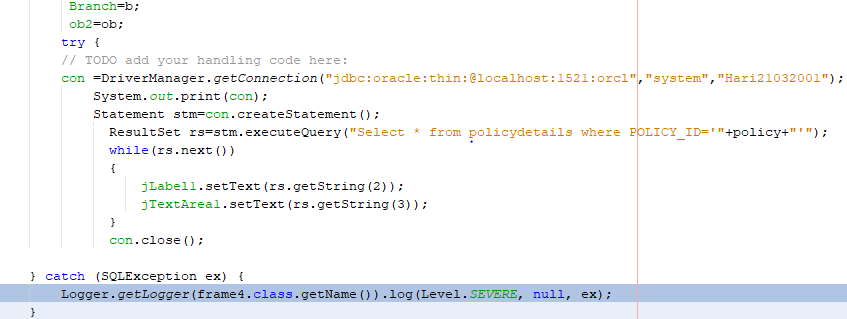
CONNECTIVITY CODE FOR EMPLOYEE LOGIN PAGE:



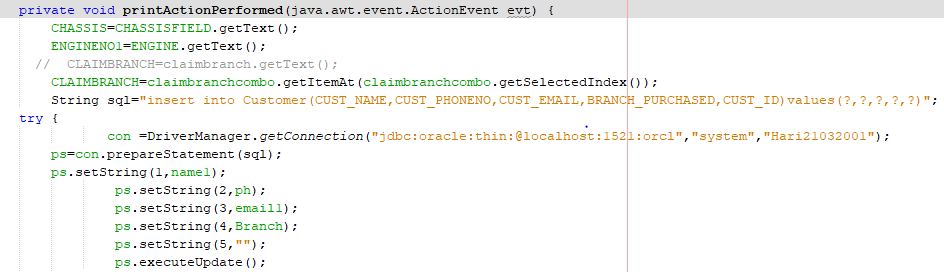
CONNECTIVITY CODE FOR MANAGER LOGIN PAGE:



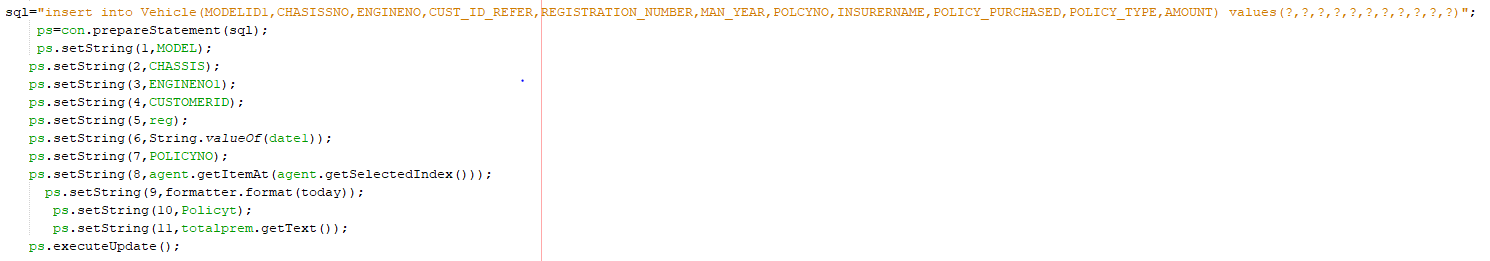
CONNECTIVITY CODE FOR SHOWING THE POLICY DETAILS:



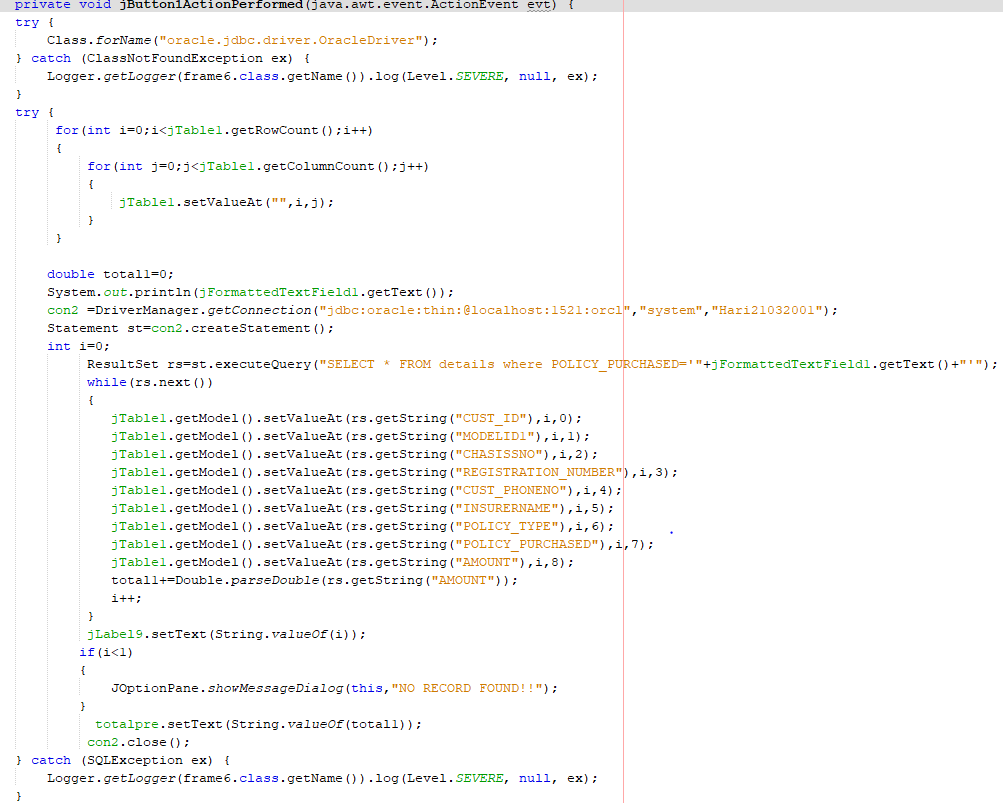
CONNECTIVITY CODE FOR INSERTING THE CUSTOMER DETIALS:



CONNECTIVITY CODE FOR INSERTING THE VEHICLE DETIALS:



CONNECTIVITY CODE FOR SEARCHING THE POLICY DETAILS ACCORDING TO THE DATE:



SOURCE CODE:

1. LOGIN PAGE:

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.Timer;

public class frame1 extends javax.swing.JFrame {

String username;

String pwd;

String emp;

String region;

public frame1() {

initComponents();

}

Timer timer1=new Timer(30,new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

if(messagep.getWidth()!=0)

{

messagep.setBounds(0,0,0,52);

if(messagep.getWidth()==0)

{

messagep.setVisible(false);

timer1.stop();

}

}

}

});

Timer timer2=new Timer(30,new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

if(messagep.getWidth()!=267)

{

messagep.setBounds(0,0,getWidth()+1,52);

if(messagep.getWidth()==267)

{ messagep.setVisible(true);

timer2.stop();

}

}

}

});

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

//new FRAME2("SALEM","101","HARIHARAN").setVisible(true);

try {

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

Connection con2;

con2 =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

System.out.print(con2);

username=String.valueOf(user.getText());

pwd=String.valueOf(password.getPassword());

emp=String.valueOf(empid.getText());

System.out.println(username+""+pwd);

int empid=Integer.parseInt(emp);

String Branch;

Statement stm=con2.createStatement();

ResultSet rs=stm.executeQuery("select \* from Employee where USERNAME='"+username+"' and PASSWORD='"+pwd+"' and EMP\_ID='"+empid+"'and DESIGNATION!='"+"MANAGER"+"'");

if(rs.next())

{ Branch=rs.getString(10);

FRAME2 f2=new FRAME2(Branch,rs.getString(2),rs.getString(1));

f2.setVisible(true);

con2.close();

this.setVisible(false);

this.dispose();

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

else{

messagep.setVisible(true);

jLabel6.setText("wrong password or username or empid");

timer2.start();

}

}

catch(NumberFormatException e1){

messagep.setVisible(true);

jLabel6.setText(" empid should be a number");

timer2.start();

}

catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(frame1.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void empidActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

timer1.start();

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

try {

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

Connection con2;

con2 =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

System.out.print(con2);

username=String.valueOf(user.getText());

pwd=String.valueOf(password.getPassword());

emp=String.valueOf(empid.getText());

System.out.println(username+""+pwd);

int empid=Integer.parseInt(emp);

String Branch;

Statement stm=con2.createStatement();

ResultSet rs=stm.executeQuery("select \* from Employee where USERNAME='"+username+"' and PASSWORD='"+pwd+"' and EMP\_ID='"+empid+"' and DESIGNATION='"+"MANAGER"+"'");

if(rs.next())

{ Branch=rs.getString(10);

frame6 f6=new frame6(Branch,this);

f6.setVisible(true);

con2.close();

this.setVisible(false);

}

else{

messagep.setVisible(true);

jLabel6.setText("wrong password or username or empid");

timer2.start();

}

}

catch(NumberFormatException e1){

messagep.setVisible(true);

jLabel6.setText(" empid should be a number");

timer2.start();

}

catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(frame1.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(frame1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(frame1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frame1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(frame1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new frame1().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JTextField empid;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel messagep;

private javax.swing.JPasswordField password;

private javax.swing.JTextField user;

private void setvisible(boolean b) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

}

HOMEPAGE OF APPLICATION:

import java.awt.Color;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author Home

\*/

public class FRAME2 extends javax.swing.JFrame {

/\*\*

\* Creates new form FRAME2

\*/

String branch;

String NAME;

String EMPID;

String region1;

public FRAME2() {

initComponents();

}

public FRAME2(String b,String n,String i) {

branch=b;

NAME=n;

EMPID=i;

initComponents();

jLabel6.setText(b);

jLabel7.setText(i);

jLabel10.setText(n);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

new frame4(branch,102,this).setVisible(true);

this.setEnabled(false);

}

private void jMenu2MouseClicked(java.awt.event.MouseEvent evt) {

frame3 frame3;

try {

frame3 = new frame3(branch);

frame3.setVisible(true);

} catch (ClassNotFoundException ex) {

Logger.getLogger(FRAME2.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void logoutMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

this.dispose();

new frame1().setVisible(true);

}

private void owndamageMouseClicked(java.awt.event.MouseEvent evt) {

new frame4(branch,101,this).setVisible(true);

this.setEnabled(false);

}

private void thirdpartymenuActionPerformed(java.awt.event.ActionEvent evt) {

new frame4(branch,102,this).setVisible(true);

this.setEnabled(false);

}

private void owndamagemenuActionPerformed(java.awt.event.ActionEvent evt) {

new frame4(branch,101,this).setVisible(true);

this.setEnabled(false);

}

private void logoutActionPerformed(java.awt.event.ActionEvent evt) {

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(FRAME2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(FRAME2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(FRAME2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(FRAME2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(() -> {

FRAME2 f2=new FRAME2();

f2.setVisible(true);

});

}

// Variables declaration - do not modify

private java.awt.Canvas canvas1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JMenu jMenu2;

private javax.swing.JMenu jMenu3;

private javax.swing.JMenuBar jMenuBar1;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JMenu logout;

private javax.swing.JButton owndamage;

private javax.swing.JMenuItem owndamagemenu;

private javax.swing.JMenuItem thirdpartymenu;

// End of variables declaration

}

POLICY PAGE:

import java.awt.Graphics;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.awt.print.PageFormat;

import java.awt.print.Printable;

import java.awt.print.PrinterException;

import java.awt.print.PrinterJob;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.util.Calendar;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import java.util.regex.Pattern;

import javax.swing.JOptionPane;

public class frame3 extends javax.swing.JFrame {

Connection con;

PreparedStatement ps=null;

/\*\*

\* Creates new form frame3

\*/

SimpleDateFormat format=new SimpleDateFormat("dd-MM-yyyy");

String Branch;

String region3;

String selected,selected2,selectedyear,Policygen;

Date date1;

int total,drive,own;

String selectedvehicle,CC,agentname,reg,name1,email1,ph,MODEL,CUSTOMERID,POLICYID,POLICYNO,CHASSIS,ENGINENO1;

double totalpre=0,pa=0,od1=0,tp=0;

String vt,g,vy,CLAIMBRANCH;

String od;

String Policyt;

int vehicle;

String ge,geographical;

String colored,seating,fuel;

public frame3() {

initComponents();

}

public frame3(String B) throws ClassNotFoundException

{initComponents();

Branch=B;

discountncbpercentage.setVisible(false);

discountncb.setVisible(false);

discountncbfield.setVisible(false);

driver.setVisible(false);

owner.setVisible(false);

gst.setEditable(false);

drive=0;

own=0;

total=0;

print.setEnabled(false);

idvfield3.setEditable(false);

try {

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

// TODO add your handling code here:

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

System.out.print(con);

Statement stm=con.createStatement();

Statement stm1=con.createStatement();

ResultSet rs=stm.executeQuery("Select DISTINCT VEHICLETYPE\_REFER from Vehicle\_REFERENCE order by VEHICLETYPE\_REFER ASC");

while(rs.next())

{

vehicletype.addItem(rs.getString("VEHICLETYPE\_REFER"));

}

rs=stm.executeQuery("Select \* from Employee where BRANCH\_NAME='"+B+"' and DESIGNATION!='"+"MANAGER"+"'");

while(rs.next()){

agent.addItem(rs.getString(2)+"("+rs.getString(1)+")");

}

rs=stm1.executeQuery("Select \* from Agent where AGENT\_BRANCH\_REF='"+B+"'");

while(rs.next()){

agent.addItem(rs.getString(1)+"(AGENT)"+"("+rs.getString(3)+")");

}

rs=stm1.executeQuery("Select DISTINCT POLICY\_TYPE from policydetails order by POLICY\_TYPE desc");

while(rs.next()){

if(!"".equals(rs.getString("POLICY\_TYPE"))){

jComboBox6.addItem(rs.getString("POLICY\_TYPE"));

}

}

rs=stm1.executeQuery("Select DISTINCT GEOGRAPHICAL1 from IRDA\_DETAILS order by GEOGRAPHICAL1 ASC");

while(rs.next()){

jComboBox3.addItem(rs.getString("GEOGRAPHICAL1"));

}

rs=stm1.executeQuery("Select \* from Branch");

while(rs.next()){

claimbranchcombo.addItem(rs.getString("BRANCH\_NAME1")+""+rs.getString("TYPE\_BRANCH"));

}

con.close();

} catch (SQLException ex) {

Logger.getLogger(frame4.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jComboBox8ItemStateChanged(java.awt.event.ItemEvent evt) {

// TODO add your handling code here:

selected=jComboBox8.getItemAt(jComboBox8.getSelectedIndex());

System.out.print(selected);

jComboBox9.removeAllItems();

try {

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

// TODO add your handling code here

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select DISTINCT MODEL from Vehicle\_REFERENCE where MAKER='"+selected+"' order by MODEL");

while(rs.next())

{

jComboBox9.addItem(rs.getString("MODEL"));

}

con.close();

} catch (SQLException ex) {

Logger.getLogger(frame4.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void colorActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void agentActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void vehicletypeFocusLost(java.awt.event.FocusEvent evt) {

vehicleyear.removeAllItems();

cc.removeAllItems();

jComboBox8.removeAllItems();

selectedvehicle=vehicletype.getItemAt(vehicletype.getSelectedIndex());

try {

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

// TODO add your handling code here

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select DISTINCT MAKER from Vehicle\_REFERENCE where VEHICLETYPE\_REFER='"+selectedvehicle+"'order by MAKER ASC");

while(rs.next())

{

jComboBox8.addItem(rs.getString("MAKER"));

}

rs=stm.executeQuery("Select DISTINCT CC from IRDA\_VEHICLE\_REFERENCE where VEHICLETYPE='"+selectedvehicle+"' order by CC");

while(rs.next())

{

cc.addItem(rs.getString("CC"));

}

rs=stm.executeQuery("Select DISTINCT YEAR from IRDA\_VEHICLE\_REFERENCE where VEHICLETYPE='"+selectedvehicle+"' order by YEAR");

while(rs.next())

{

if(!"NULL".equals(rs.getString("YEAR"))){

vehicleyear.addItem(rs.getString("YEAR"));

}

}

con.close();

} catch (SQLException ex) {

Logger.getLogger(frame4.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jComboBox9FocusLost(java.awt.event.FocusEvent evt) {

}

private void jComboBox9ItemStateChanged(java.awt.event.ItemEvent evt) {

// TODO add your handling code here:

color.removeAllItems();

selected2=jComboBox9.getItemAt(jComboBox9.getSelectedIndex());

System.out.print(selected2);

try {

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

// TODO add your handling code here

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select DISTINCT COLOR from Vehicle\_REFERENCE where MODEL='"+selected2+"' order by COLOR");

while(rs.next())

{

color.addItem(rs.getString("COLOR"));

}

con.close();

} catch (SQLException ex) {

Logger.getLogger(frame4.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void printActionPerformed(java.awt.event.ActionEvent evt) {

CHASSIS=CHASSISFIELD.getText();

ENGINENO1=ENGINE.getText();

// CLAIMBRANCH=claimbranch.getText();

CLAIMBRANCH=claimbranchcombo.getItemAt(claimbranchcombo.getSelectedIndex());

String sql="insert into Customer(CUST\_NAME,CUST\_PHONENO,CUST\_EMAIL,BRANCH\_PURCHASED,CUST\_ID)values(?,?,?,?,?)";

try {

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

ps=con.prepareStatement(sql);

ps.setString(1,name1);

ps.setString(2,ph);

ps.setString(3,email1);

ps.setString(4,Branch);

ps.setString(5,"");

ps.executeUpdate();

Statement stm=con.createStatement();

ResultSet rs1=stm.executeQuery("Select \* from Customer");

while(rs1.next()){

CUSTOMERID=rs1.getString("CUST\_ID");

}

ResultSet rs=stm.executeQuery("Select MODELID from Vehicle\_REFERENCE where MAKER='"+selected+"' AND MODEL='"+selected2+"' AND COLOR='"+colored+"'");

while(rs.next())

{

MODEL=rs.getString("MODELID");

}

SimpleDateFormat formatter=new SimpleDateFormat("dd-MM-yyyy");

Date today=new Date();

Calendar now=Calendar.getInstance();

int year=Calendar.getInstance().get(Calendar.YEAR);

String year2=String.valueOf(year);

POLICYNO=Branch+year2;

rs=stm.executeQuery("Select policynogenerator from policygenerater");

while(rs.next())

{

Policygen=rs.getString("policynogenerator");

}

POLICYNO+=Policygen;

if("THIRD-PARTY INSURANCE".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex()))){

Policyt="THIRD-PARTY INSURANCE";

}

else if("THIRD-PARTY INSURANCE".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex())) && driver.isSelected() && owner.isSelected()){

Policyt="THIRD-PARTY INSURANCE"+","+"PERSONAL ACCIDENT FOR DRIVER"+","+"OWNER";

}

else if("THIRD-PARTY INSURANCE".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex())) && owner.isSelected()){

Policyt="THIRD-PARTY INSURANCE"+","+"PERSONAL ACCIDENT FOR OWNER";

}

else if("THIRD-PARTY INSURANCE".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex())) && driver.isSelected()){

Policyt="THIRD-PARTY INSURANCE"+","+"PERSONAL ACCIDENT FOR DRIVER";

}

else if("OWN DAMAGE COVER".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex()))){

Policyt="OWN DAMAGE COVER";

}

else if("OWN DAMAGE COVER".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex())) && driver.isSelected() && owner.isSelected()){

Policyt="OWN DAMAGE COVER"+","+"PERSONAL ACCIDENT FOR DRIVER"+","+"OWNER";

}

else if("OWN DAMAGE COVER".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex())) && owner.isSelected()){

Policyt="OWN DAMAGE COVER"+","+"PERSONAL ACCIDENT FOR OWNER";

}

else if("OWN DAMAGE COVER".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex())) && driver.isSelected()){

Policyt="OWN DAMAGE COVER"+","+"PERSONAL ACCIDENT FOR DRIVER";

}

sql="insert into Vehicle(MODELID1,CHASISSNO,ENGINENO,CUST\_ID\_REFER,REGISTRATION\_NUMBER,MAN\_YEAR,POLCYNO,INSURERNAME,POLICY\_PURCHASED,POLICY\_TYPE,AMOUNT) values(?,?,?,?,?,?,?,?,?,?,?)";

ps=con.prepareStatement(sql);

ps.setString(1,MODEL);

ps.setString(2,CHASSIS);

ps.setString(3,ENGINENO1);

ps.setString(4,CUSTOMERID);

ps.setString(5,reg);

ps.setString(6,String.valueOf(date1));

ps.setString(7,POLICYNO);

ps.setString(8,agent.getItemAt(agent.getSelectedIndex()));

ps.setString(9,formatter.format(today));

ps.setString(10,Policyt);

ps.setString(11,totalprem.getText());

ps.executeUpdate();

con.close();

frame5 f5=new frame5(this);

f5.setVisible(true);

this.setEnabled(false);

this.setVisible(false);

}

catch (SQLException ex) {

Logger.getLogger(frame3.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void idvfield1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void ncbcheckActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(ncbcheck.isSelected()){

discountncbpercentage.setVisible(true);

discountncb.setVisible(true);

discountncbfield.setVisible(true);

}

else{

discountncbpercentage.setVisible(false);

discountncb.setVisible(false);

discountncbfield.setVisible(false);

}

}

private void jComboBox6ItemStateChanged(java.awt.event.ItemEvent evt) {

// TODO add your handling code here:

try {

if("THIRD-PARTY INSURANCE".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex()))){

jLabel2.setVisible(false);

vehicleyear.setVisible(false);

idv1.setVisible(false);

idvfield1.setVisible(false);

CC=cc.getItemAt(cc.getSelectedIndex());

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select VEHICLEID from IRDA\_VEHICLE\_REFERENCE where CC='"+CC+"' AND YEAR='"+"NULL"+"'");

while(rs.next())

{

vehicle=rs.getInt("VEHICLEID");

}

rs=stm.executeQuery("Select \* from IRDA\_DETAILS where VEHICLEID1='"+vehicle+"' ");

while(rs.next())

{ POLICYNO=rs.getString(1);

idvfield3.setText(rs.getString("AMOUNT"));

} con.close();}

else {

jLabel2.setVisible(true);

vehicleyear.setVisible(true);

idv1.setVisible(true);

idvfield1.setVisible(true);

CC=cc.getItemAt(cc.getSelectedIndex());

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select VEHICLEID from IRDA\_VEHICLE\_REFERENCE where CC='"+CC+"' AND YEAR='"+"NULL"+"'");

while(rs.next())

{

vehicle=rs.getInt("VEHICLEID");

}

rs=stm.executeQuery("Select AMOUNT from IRDA\_DETAILS where VEHICLEID1='"+vehicle+"' ");

while(rs.next())

{

idvfield3.setText(rs.getString("AMOUNT"));

}

con.close();

}

} catch (SQLException ex) {

Logger.getLogger(frame4.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jComboBox6FocusLost(java.awt.event.FocusEvent evt) {

try {

// TODO add your handling code here

if("THIRD-PARTY INSURANCE".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex()))){

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

CC=cc.getItemAt(cc.getSelectedIndex());

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select VEHICLEID from IRDA\_VEHICLE\_REFERENCE where CC='"+CC+"' AND YEAR='"+"NULL"+"'");

while(rs.next())

{

vehicle=rs.getInt("VEHICLEID");

System.out.print(vehicle);

}

rs=stm.executeQuery("Select \* from IRDA\_DETAILS where VEHICLEID1='"+vehicle+"' ");

while(rs.next())

{

idvfield3.setText(rs.getString(6));

System.out.println(rs.getString(6));

}

} con.close();

} catch (SQLException ex) {

Logger.getLogger(frame4.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void ncbcheck1ActionPerformed(java.awt.event.ActionEvent evt) {

if(ncbcheck1.isSelected()){

driver.setVisible(true);

owner.setVisible(true);

totalprem.setText(String.valueOf(totalpre));

}

else{

driver.setVisible(false);

owner.setVisible(false);

driver.setSelected(false);

owner.setSelected(false);

drive=0;

own=0;

}

}

private void driverActionPerformed(java.awt.event.ActionEvent evt) {

}

private void ownerActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

if("".equals(idvfield3.getText())){

JOptionPane.showMessageDialog(null,"PLEASE SELECT POLICYTYPE");

}

else{

print.setEnabled(true);

pa=0;

if(ncbcheck1.isSelected()){

pa=0;

if(driver.isSelected())

{

pa+=50;

}

else{

JOptionPane.showMessageDialog(null,"PLEASE SELECT PA OR UNCHECK THE PACHECKBOX");

ncbcheck1.setSelected(false);

}

if(owner.isSelected())

{

pa+=275;

}

}

else{

owner.setVisible(false);

driver.setVisible(false);

}

try{

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

if("THIRD-PARTY INSURANCE".equals(jComboBox6.getItemAt(jComboBox6.getSelectedIndex()))){

tp=Integer.parseInt(idvfield3.getText());

int g=Integer.parseInt(gst.getText());

if(ncbcheck.isSelected()){

if(!"".equals(discountncbfield.getText())){

int ncb=Integer.parseInt(discountncbfield.getText());

totalpre=tp\*(0.18);

System.out.println(totalpre);

totalpre-=tp\*ncb/100;

System.out.println(totalpre);

totalpre+=tp;

tp=totalpre;

totalpre+=pa;

System.out.print(totalpre);

totalprem.setText(String.valueOf(totalpre));

}

else{

JOptionPane.showMessageDialog(null,"PLEASE ENETER THE VALUE FOR NCB FIELD");

}

}

else{

System.out.println(g);

totalpre=(tp\*g/100)+tp;

totalpre+=pa;

totalprem.setText(String.valueOf(totalpre));

}

vt=vehicletype.getItemAt(vehicletype.getSelectedIndex());

geographical=jComboBox3.getItemAt(jComboBox3.getSelectedIndex());

vy=vehicleyear.getItemAt(vehicleyear.getSelectedIndex());

fuel=jComboBox2.getItemAt(jComboBox2.getSelectedIndex());

colored=color.getItemAt(color.getSelectedIndex());

seating=jComboBox4.getItemAt(jComboBox4.getSelectedIndex());

agentname=agent.getItemAt(agent.getSelectedIndex());

reg=register.getText();

name1=name.getText();

email1=email.getText();

ph=phone.getText();

}

else{

tp=Integer.parseInt(idvfield3.getText());

int gee=Integer.parseInt(gst.getText());

if(ncbcheck.isSelected()){

if(!"".equals(discountncbfield.getText())){

int ncb=Integer.parseInt(discountncbfield.getText());

totalpre=tp\*gee/100;

System.out.println(totalpre);

totalpre-=tp\*ncb/100;

System.out.println(totalpre);

totalpre+=tp;

tp=totalpre;

totalpre+=pa;

System.out.print(totalpre);

}

else{

JOptionPane.showMessageDialog(null,"PLEASE ENETER THE VALUE FOR NCB FIELD");

}

}

else{

System.out.println(gee);

totalpre=(tp\*gee/100)+tp;

totalpre+=pa;

}

CC=cc.getItemAt(cc.getSelectedIndex());

vt=vehicletype.getItemAt(vehicletype.getSelectedIndex());

geographical=jComboBox3.getItemAt(jComboBox3.getSelectedIndex());

vy=vehicleyear.getItemAt(vehicleyear.getSelectedIndex());

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select VEHICLEID from IRDA\_VEHICLE\_REFERENCE where CC='"+CC+"' AND YEAR='"+vy+"' AND VEHICLETYPE='"+vt+"'");

while(rs.next())

{System.out.println("VEHICLEID"+rs.getString("VEHICLEID"));

ge=rs.getString("VEHICLEID");

}

rs=stm.executeQuery("Select \* from IRDA\_DETAILS where VEHICLEID1='"+ge+"' AND GEOGRAPHICAL1='"+geographical+"'");

while(rs.next())

{

od=rs.getString(6);

POLICYNO=rs.getString(1);

System.out.println(rs.getString(1));

}

if(!"".equals(idvfield1.getText())){

int idv=Integer.parseInt(idvfield1.getText());

od1=(idv\*Double.parseDouble(od))/100;

double tot=od1+totalpre;

System.out.print(tot);

totalprem.setText(String.valueOf(tot));

}

else{

JOptionPane.showMessageDialog(null,"PLEASE ENTER VALUE IN IDVFIELD");

}

fuel=jComboBox2.getItemAt(jComboBox2.getSelectedIndex());

colored=color.getItemAt(color.getSelectedIndex());

seating=jComboBox4.getItemAt(jComboBox4.getSelectedIndex());

agentname=agent.getItemAt(agent.getSelectedIndex());

reg=register.getText();

name1=name.getText();

email1=email.getText();

ph=phone.getText();

}

con.close();

}

catch(NumberFormatException e)

{

JOptionPane.showMessageDialog(null,"PLEASE ENTER INTEGER VALUE");

} catch (SQLException ex) {

Logger.getLogger(frame3.class.getName()).log(Level.SEVERE, null, ex);

}}

}

private void jTextField10ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextField10FocusLost(java.awt.event.FocusEvent evt) {

String date=jTextField10.getText();

try {

Date today=new Date();

date1=format.parse(date);

SimpleDateFormat format1=new SimpleDateFormat("dd-MM-yyyy");

if(date1.after(today)){

JOptionPane.showMessageDialog(null,"THE GIVEN DATE IS GRETAER THAN THE CURRENT DATE");

jTextField10.setText("");

}

} catch (ParseException ex) {

JOptionPane.showMessageDialog(null,"DATE FORMAT IS WRONG PLEASE ENTER THE CORRECT FORMAT");

}

}

private void CHASSISFIELDActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void nameFocusLost(java.awt.event.FocusEvent evt) {

if(name.getText().isEmpty())

{

//JOptionPane.showMessageDialog(null,"NAME FIELD SHOULD NOT BE EMPTY");

}

}

private void emailFocusLost(java.awt.event.FocusEvent evt) {

try {

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

} catch (SQLException ex) {

Logger.getLogger(frame3.class.getName()).log(Level.SEVERE, null, ex);

}

Statement stm;

//try {

/\*stm = con.createStatement();

ResultSet rs=stm.executeQuery("Select CUST\_EMAIL from Customer where CUST\_EMAIL ='"+email.getText()+"'");

while(rs.next())

{

if(rs.getString("CUST\_EMAIL ").equals(email.getText()))

{

JOptionPane.showMessageDialog(null,"EMAIL ALREADY EXIST");

}

}

con.close();

}

catch (SQLException ex) {

Logger.getLogger(frame3.class.getName()).log(Level.SEVERE, null, ex);

}

\*/

}

private void phoneFocusLost(java.awt.event.FocusEvent evt) {

//try {

if("".equals(phone.getText()))

{

JOptionPane.showMessageDialog(null,"PHONENOFIELD SHOULD NOT BE EMPTY");

}

else if(!"".equals(phone.getText())){

/\*con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select CUST\_PHONENO from Customer where CUST\_PHONENO='"+phone.getText()+"'");

while(rs.next())

{

if(rs.getString("CUST\_PHONENO").equals(phone.getText()))

{

JOptionPane.showMessageDialog(null,"PHONE NUMBER ALREADY EXIST");

}

}

con.close();

}} catch (SQLException ex) {

Logger.getLogger(frame3.class.getName()).log(Level.SEVERE, null, ex);

} \*/}

}

private void emailFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

jLabel29.setText("");

}

private void registerFocusLost(java.awt.event.FocusEvent evt) {

try {

if("".equals(register.getText()))

{

JOptionPane.showMessageDialog(null,"REGISTRATION NUMBER FIELD SHOULD NOT BE EMPTY");

register.requestFocus();

}

else if(!"".equals(register.getText())){

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select REGISTRATION\_NUMBER from Vehicle where REGISTRATION\_NUMBER='"+register.getText()+"'");

while(rs.next())

{

if(rs.getString("REGISTRATION\_NUMBER").equals(register.getText()))

{

JOptionPane.showMessageDialog(null,"REGISTRATION NUMBER ALREADY EXIST");

}

}

con.close();

}} catch (SQLException ex) {

Logger.getLogger(frame3.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void CHASSISFIELDFocusLost(java.awt.event.FocusEvent evt) {

try {

if("".equals(CHASSISFIELD.getText()))

{

JOptionPane.showMessageDialog(null,"CHASISSNO NUMBER FIELD SHOUL NOT BE EMPTY");

CHASSISFIELD.requestFocus();

}

else if(!"".equals(CHASSISFIELD.getText())){

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select CHASISSNO from Vehicle where CHASISSNO='"+CHASSISFIELD.getText()+"'");

while(rs.next())

{

if(rs.getString("CHASISSNO").equals(CHASSISFIELD.getText()))

{

JOptionPane.showMessageDialog(null,"CHASISSNO NUMBER ALREADY EXIST");

CHASSISFIELD.setText("");

}

}

con.close();

}} catch (SQLException ex) {

Logger.getLogger(frame3.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void ENGINEFocusLost(java.awt.event.FocusEvent evt) {

try {

if("".equals(ENGINE.getText()))

{

JOptionPane.showMessageDialog(null,"ENGINE NUMBER FIELD SHOUL NOT BE EMPTY");

ENGINE.requestFocus();

}

else if(!"".equals(ENGINE.getText())){

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select ENGINENO from Vehicle where ENGINENO='"+ENGINE.getText()+"'");

while(rs.next())

{

if(rs.getString("ENGINENO").equals(ENGINE.getText()))

{

JOptionPane.showMessageDialog(null,"ENGINE NUMBER ALREADY EXIST");

ENGINE.setText("");

}

}

con.close();

}} catch (SQLException ex) {

Logger.getLogger(frame3.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void nameKeyPressed(java.awt.event.KeyEvent evt) {

}

private void nameKeyTyped(java.awt.event.KeyEvent evt) {

if(!"".equals(name.getText())){

if(name.getText().matches("^[a-zA-Z\\s]{1,}[\\.]{0,1}[A-Za-z\\s]{0,}$")){

jLabel28.setText("");

}

else{

jLabel28.setText("INVALID NAME!!");

}

}

}

private void emailKeyTyped(java.awt.event.KeyEvent evt) {

String emailregex="^[a-zA-Z0-9\_+&\*-]+(?:\\."+

"[a-zA-Z0-9\_+&\*-]+)\*@"+

"(?:[a-zA-Z0-9-]+\\.)+[a-z"+"A-Z]{2,7}$";

Pattern pat=Pattern.compile(emailregex);

if(!pat.matcher(email.getText()).matches())

{

jLabel29.setText("EMAIL IS NOT A VALD ONE");

email.setFocusable(true);

}

else{

jLabel29.setText("");

}

}

private void phoneKeyTyped(java.awt.event.KeyEvent evt) {

String phoneno=phone.getText();

if(phoneno.matches("[0-9]+")&& phoneno.length()==9)

{jLabel30.setText("");

address.requestFocus();

}

else{

jLabel30.setText("PHONENO IS NOT A VALD ONE");

}

}

private void registerKeyTyped(java.awt.event.KeyEvent evt) {

String emailregex="^[A-Z]{2}\\s[0-9]{2}\\s[A-Z]{2}\\s[0-9]{3}$";

Pattern pat=Pattern.compile(emailregex);

if(!pat.matcher(register.getText()).matches())

{

jLabel31.setText("REGISTRATIONNO IS NOT A VALID!");

register.setFocusable(true);

}

else{

jLabel31.setText("");

}

}

private void nameFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

}

private void phoneFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

}

private void jFormattedTextField1FocusLost(java.awt.event.FocusEvent evt) {

String date=jFormattedTextField1.getText();

try {

Date today=new Date();

date1=format.parse(date);

SimpleDateFormat format1=new SimpleDateFormat("dd-MM-yyyy");

if(date1.after(today)){

JOptionPane.showMessageDialog(null,"THE GIVEN DATE IS GRETAER THAN THE CURRENT DATE");

jTextField10.setText("");

}

} catch (ParseException ex) {

JOptionPane.showMessageDialog(null,"DATE FORMAT IS WRONG PLEASE ENTER THE CORRECT FORMAT");

}

}

private void jButton2FocusLost(java.awt.event.FocusEvent evt) {

// print.setEnabled(false);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(frame3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(frame3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frame3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(frame3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new frame3().setVisible(true);

}

});

}

VIEW OF POLICY PURCHASED DETAILS BY MANAGER:

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class frame6 extends javax.swing.JFrame {

Connection con2;

frame1 ob2;

double Total=0;

public frame6() throws ClassNotFoundException, SQLException {

initComponents();

}

public frame6(String Branch,frame1 ob) throws ClassNotFoundException, SQLException {

initComponents();

ob2=ob;

Date date=new Date();

DateFormat dateformat=new SimpleDateFormat("dd-MM-yyyy");

String date1=dateformat.format(date);

System.out.print(date1);

try{

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

con2 =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement st=con2.createStatement();

jLabel2.setText(Branch);

ResultSet rs=st.executeQuery("SELECT \* FROM Branch where BRANCH\_NAME1='"+Branch+"'");

while(rs.next())

{

jLabel5.setText(rs.getString("MANAGER\_NAME"));

}

rs=st.executeQuery("SELECT \* FROM Employee where BRANCH\_NAME='"+Branch+"'");

while(rs.next())

{

insurer.addItem(rs.getString("EMP\_NAME")+"("+rs.getString("EMP\_ID")+")");

}

rs=st.executeQuery("SELECT \* FROM details where BRANCH\_PURCHASED='"+Branch+"' and POLICY\_PURCHASED='"+date1+"'");

int i=0;

while(rs.next())

{

jTable1.getModel().setValueAt(rs.getString("CUST\_ID"),i,0);

jTable1.getModel().setValueAt(rs.getString("MODELID1"),i,1);

jTable1.getModel().setValueAt(rs.getString("CHASISSNO"),i,2);

jTable1.getModel().setValueAt(rs.getString("REGISTRATION\_NUMBER"),i,3);

jTable1.getModel().setValueAt(rs.getString("CUST\_PHONENO"),i,4);

jTable1.getModel().setValueAt(rs.getString("INSURERNAME"),i,5);

jTable1.getModel().setValueAt(rs.getString("POLICY\_TYPE"),i,6);

jTable1.getModel().setValueAt(rs.getString("POLICY\_PURCHASED"),i,7);

jTable1.getModel().setValueAt(rs.getString("AMOUNT"),i,8);

Total+=Double.parseDouble(rs.getString("AMOUNT"));

i++;

}

jLabel9.setText(String.valueOf(i));

if(i<1)

{

JOptionPane.showMessageDialog(this,"NO RECORD FOUND!!");

}

totalpre.setText(String.valueOf(Total));

con2.close();

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

}

ob2.setVisible(true);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

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

} catch (ClassNotFoundException ex) {

Logger.getLogger(frame6.class.getName()).log(Level.SEVERE, null, ex);

}

try {

for(int i=0;i<jTable1.getRowCount();i++)

{

for(int j=0;j<jTable1.getColumnCount();j++)

{

jTable1.setValueAt("",i,j);

}

}

double total1=0;

System.out.println(jFormattedTextField1.getText());

con2 =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement st=con2.createStatement();

int i=0;

ResultSet rs=st.executeQuery("SELECT \* FROM details where POLICY\_PURCHASED='"+jFormattedTextField1.getText()+"'");

while(rs.next())

{

jTable1.getModel().setValueAt(rs.getString("CUST\_ID"),i,0);

jTable1.getModel().setValueAt(rs.getString("MODELID1"),i,1);

jTable1.getModel().setValueAt(rs.getString("CHASISSNO"),i,2);

jTable1.getModel().setValueAt(rs.getString("REGISTRATION\_NUMBER"),i,3);

jTable1.getModel().setValueAt(rs.getString("CUST\_PHONENO"),i,4);

jTable1.getModel().setValueAt(rs.getString("INSURERNAME"),i,5);

jTable1.getModel().setValueAt(rs.getString("POLICY\_TYPE"),i,6);

jTable1.getModel().setValueAt(rs.getString("POLICY\_PURCHASED"),i,7);

jTable1.getModel().setValueAt(rs.getString("AMOUNT"),i,8);

total1+=Double.parseDouble(rs.getString("AMOUNT"));

i++;

}

jLabel9.setText(String.valueOf(i));

if(i<1)

{

JOptionPane.showMessageDialog(this,"NO RECORD FOUND!!");

}

totalpre.setText(String.valueOf(total1));

con2.close();

} catch (SQLException ex) {

Logger.getLogger(frame6.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

try {

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

} catch (ClassNotFoundException ex) {

Logger.getLogger(frame6.class.getName()).log(Level.SEVERE, null, ex);

}

try {

for(int i=0;i<jTable1.getRowCount();i++)

{

for(int j=0;j<jTable1.getColumnCount();j++)

{

jTable1.setValueAt("",i,j);

}

}

double total1=0;

System.out.println(jFormattedTextField1.getText());

con2 =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

Statement st=con2.createStatement();

int i=0;

ResultSet rs=st.executeQuery("SELECT \* FROM details where INSURERNAME='"+insurer.getItemAt(insurer.getSelectedIndex())+"'");

while(rs.next())

{

jTable1.getModel().setValueAt(rs.getString("CUST\_ID"),i,0);

jTable1.getModel().setValueAt(rs.getString("MODELID1"),i,1);

jTable1.getModel().setValueAt(rs.getString("CHASISSNO"),i,2);

jTable1.getModel().setValueAt(rs.getString("REGISTRATION\_NUMBER"),i,3);

jTable1.getModel().setValueAt(rs.getString("CUST\_PHONENO"),i,4);

jTable1.getModel().setValueAt(rs.getString("INSURERNAME"),i,5);

jTable1.getModel().setValueAt(rs.getString("POLICY\_TYPE"),i,6);

jTable1.getModel().setValueAt(rs.getString("POLICY\_PURCHASED"),i,7);

jTable1.getModel().setValueAt(rs.getString("AMOUNT"),i,8);

total1+=Double.parseDouble(rs.getString("AMOUNT"));

i++;

}

if(i<1)

{

JOptionPane.showMessageDialog(this,"NO RECORD FOUND!!");

}

jLabel9.setText(String.valueOf(i));

totalpre.setText(String.valueOf(total1));

con2.close();

} catch (SQLException ex) {

Logger.getLogger(frame6.class.getName()).log(Level.SEVERE, null, ex);

}

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

try {

new frame6().setVisible(true);

} catch (ClassNotFoundException ex) {

Logger.getLogger(frame6.class.getName()).log(Level.SEVERE, null, ex);

} catch (SQLException ex) {

Logger.getLogger(frame6.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

}

GENERATION OF POLICY:

import java.awt.Desktop;

import java.awt.Graphics;

import static java.awt.PageAttributes.MediaType.C;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.awt.print.PageFormat;

import java.awt.print.Printable;

import java.awt.print.PrinterException;

import java.awt.print.PrinterJob;

import java.io.File;

import java.io.FileFilter;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Calendar;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

public class frame5 extends javax.swing.JFrame {

frame3 ob2;

Connection con;

/\*\*

\* Creates new form frame5

\*/

public frame5() {

initComponents();

}

public frame5(frame3 ob) throws SQLException {

initComponents();

try {

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

ob2=ob;

maker.setText(ob2.selected+"/"+ob2.selected2);

branch.setText(ob2.Branch);

claimbranch.setText(ob2.CLAIMBRANCH);

geographical.setText(ob2.geographical);

sc.setText(ob2.seating);

fueltype.setText(ob2.fuel);

color.setText(ob2.colored);

agent.setText(ob2.agentname);

reg.setText(ob2.reg);

email.setText(ob2.email1);

policyno.setText(ob2.POLICYNO);

name.setText(ob2.name1);

Calendar cal=Calendar.getInstance();

Date today=cal.getTime();

cal.add(Calendar.YEAR,1);

Date nextyear=cal.getTime();

coverdate.setText(String.valueOf(nextyear));

phone.setText(ob2.ph);

pa.setText(String.valueOf(ob2.pa));

tp.setText(String.valueOf(ob2.tp));

od.setText(String.valueOf(ob2.od));

AGE.setText(ob2.vy);

cc.setText(ob2.CC);

totalprem.setText(String.valueOf(ob2.totalpre));

try{

con =DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","system","Hari21032001");

System.out.print(con);

Statement stm=con.createStatement();

ResultSet rs=stm.executeQuery("Select \* from Vehicle");

while(rs.next())

{

custid.setText(rs.getString("CUST\_ID\_REFER"));

policyno.setText(rs.getString("POLCYNO"));

engine.setText(rs.getString("ENGINENO"));

CHASSIS.setText(rs.getString("CHASISSNO"));

}

rs=stm.executeQuery("Select \* from Branch where BRANCH\_NAME1='"+ob2.Branch+"'");

while(rs.next())

{

phoneno.setText(rs.getString("BRANCH\_PHONENO"));

branch3.setText(rs.getString("ADDRESS")+""+rs.getString("BRANCH\_PHONENO"));

}

con.close();

}

catch (SQLException ex) {

Logger.getLogger(frame5.class.getName()).log(Level.SEVERE, null, ex);

}

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

ob2.setEnabled(true);

ob2.setVisible(true);

}});

} catch (ClassNotFoundException ex) {

Logger.getLogger(frame5.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

PrinterJob job=PrinterJob.getPrinterJob();

job.setJobName("print data");

job.setPrintable(new Printable(){

public int print(Graphics pg, PageFormat pf, int pageNum) {

pf.setOrientation(PageFormat.PORTRAIT);

if(pageNum>0){

return Printable.NO\_SUCH\_PAGE;

}

java.awt.Graphics2D g2=(java.awt.Graphics2D)pg;

g2.translate(pf.getImageableX(),pf.getImageableY());

g2.scale(1,1);

jPanel1.paint(g2);

return Printable.PAGE\_EXISTS;}

});

boolean complete=job.printDialog();

if(complete){

try{

job.print();

}

catch(PrinterException e){

}

}

File theNewestFile=null;

File dir=new File("C:/Users/Home/Documents/harifees/PROJECT/policy\_paper");

File[] files=dir.listFiles();

if(files==null || files.length==0)

{

}

File lastModifiedFile=files[0];

for(int i=0;i<files.length;i++)

{

if(lastModifiedFile.lastModified()<files[i].lastModified())

{

lastModifiedFile=files[i];

}

}

Desktop desktop=Desktop.getDesktop();

if(lastModifiedFile.exists())

{

try {

desktop.open(lastModifiedFile);

} catch (IOException ex) {

Logger.getLogger(frame5.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

this.dispose();

//ob2.setVisible(true);

//ob2.setEnabled(true);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(frame5.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(frame5.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frame5.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(frame5.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new frame5().setVisible(true);

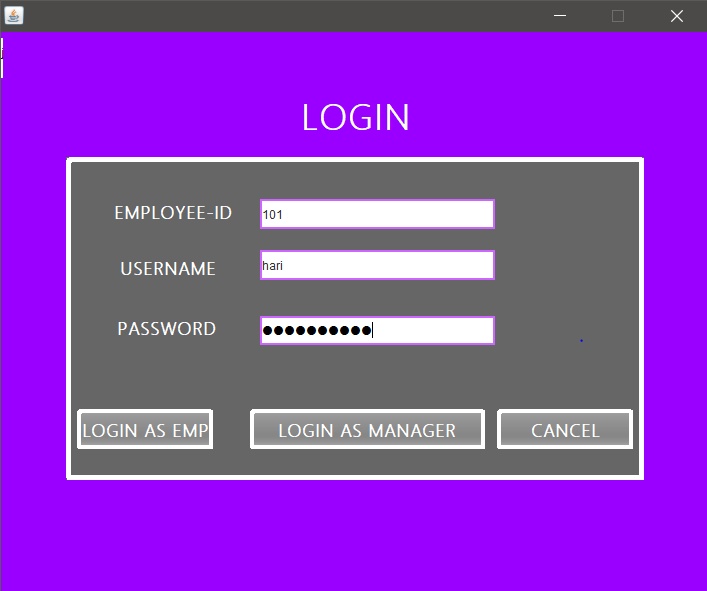
}

});

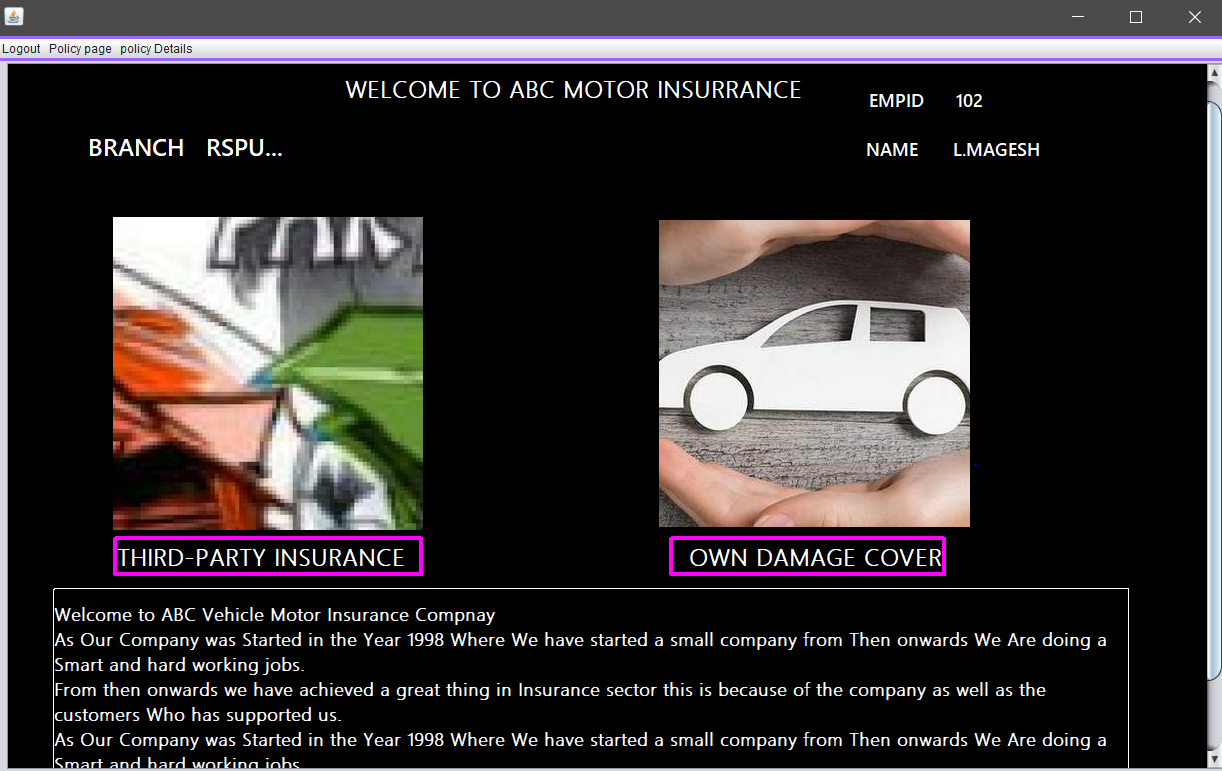
}

OUTPUT:

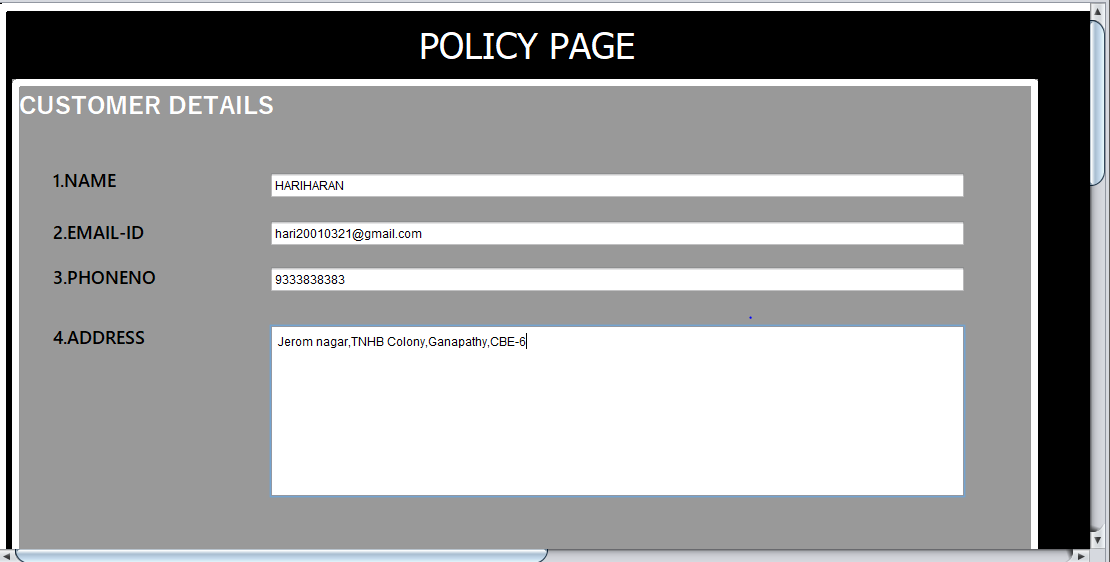
Login page:

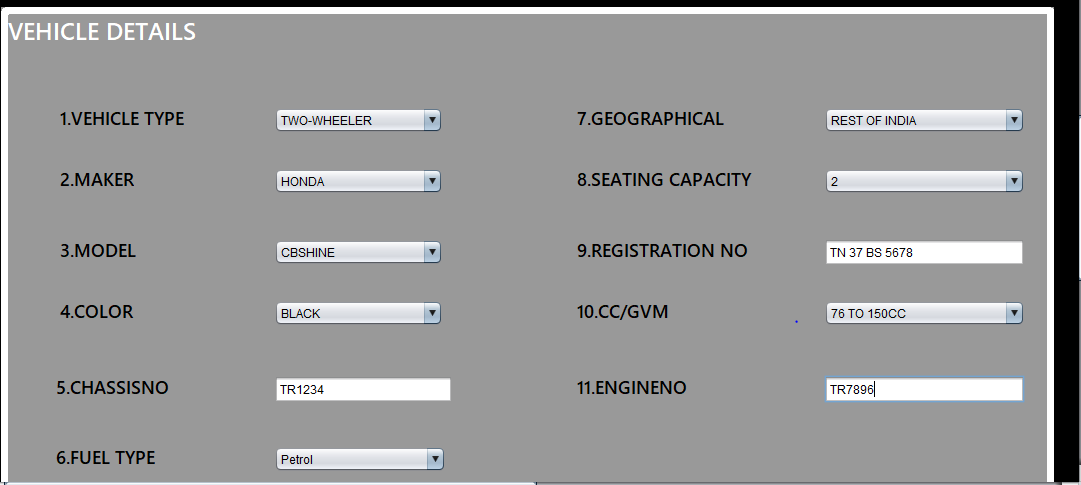


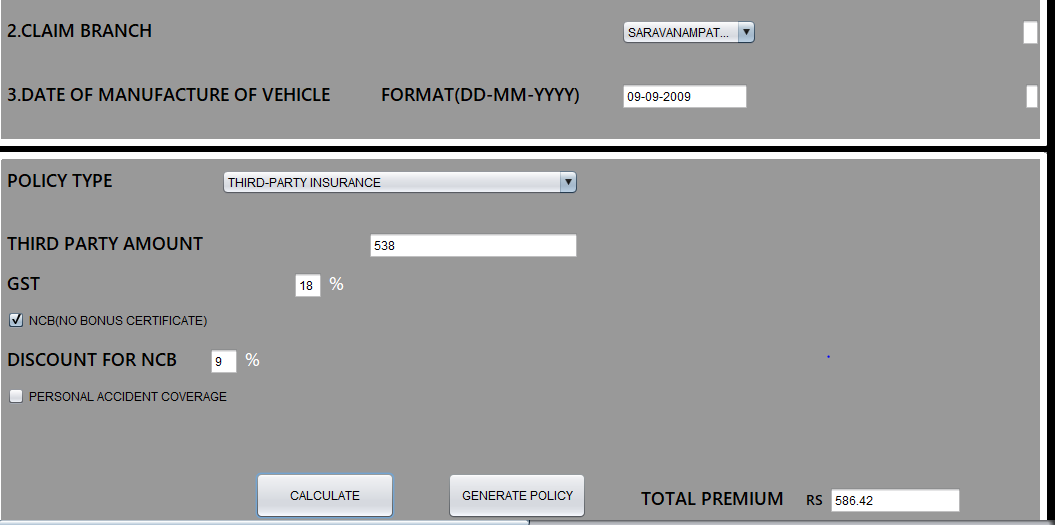
HOME PAGE:



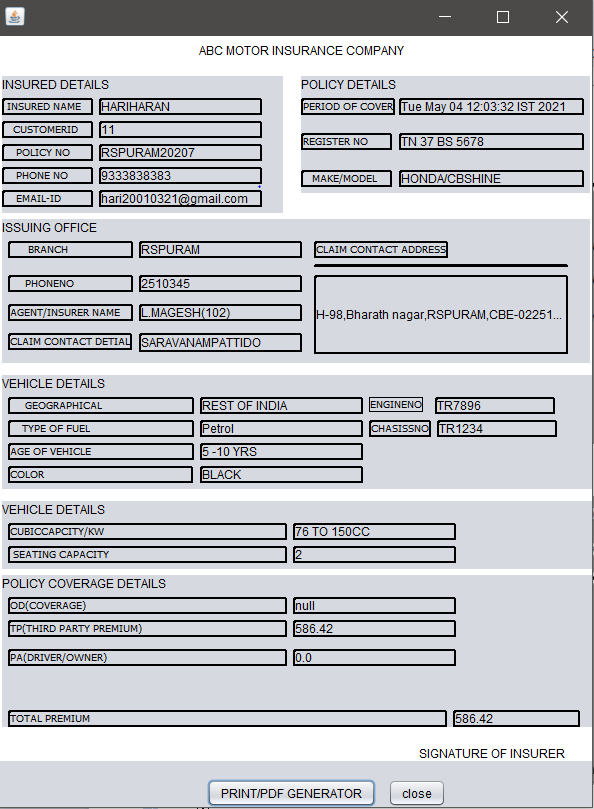
HOME PAGE:





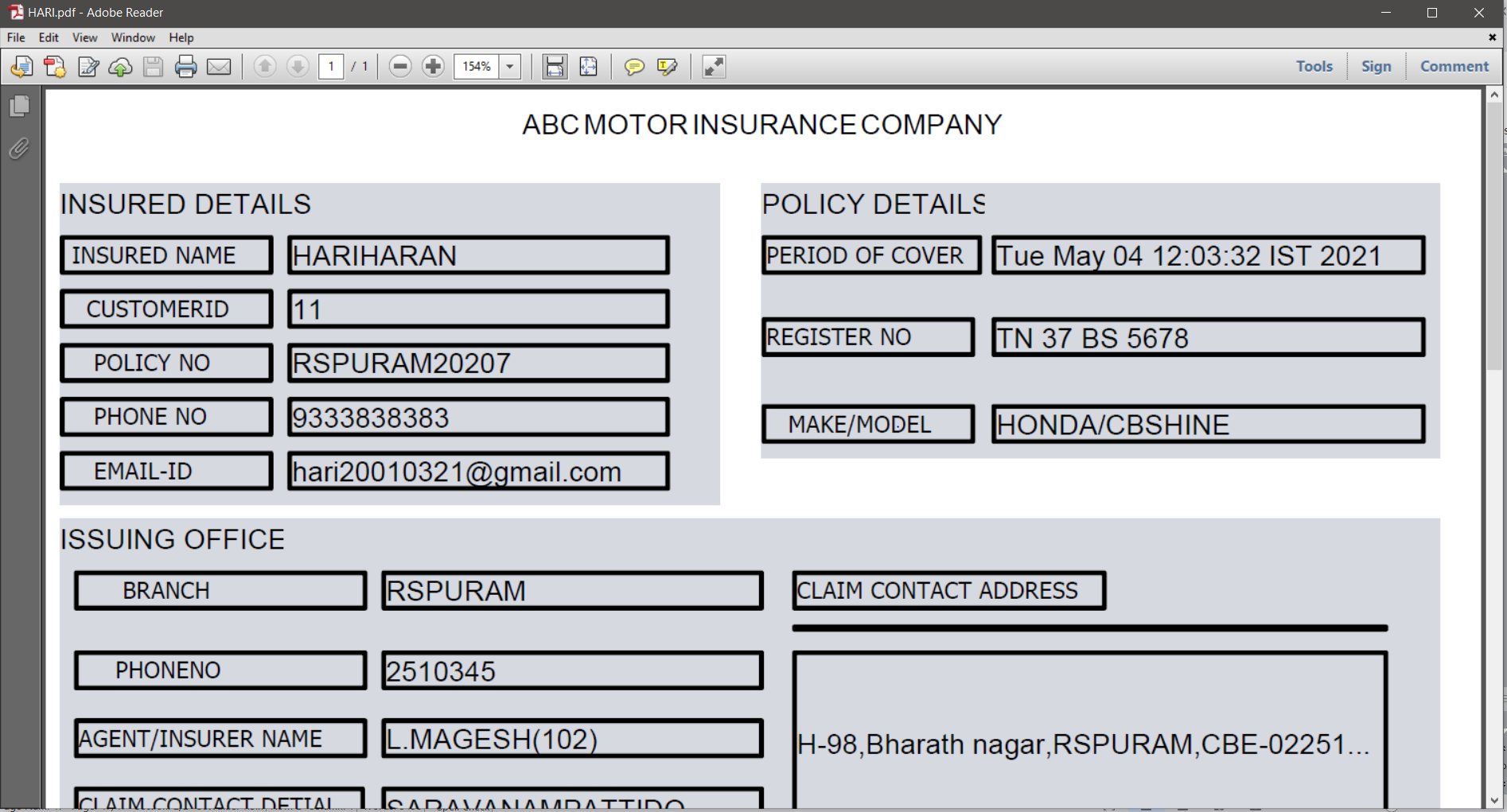


GENERATION OF POLICY-PAGE:

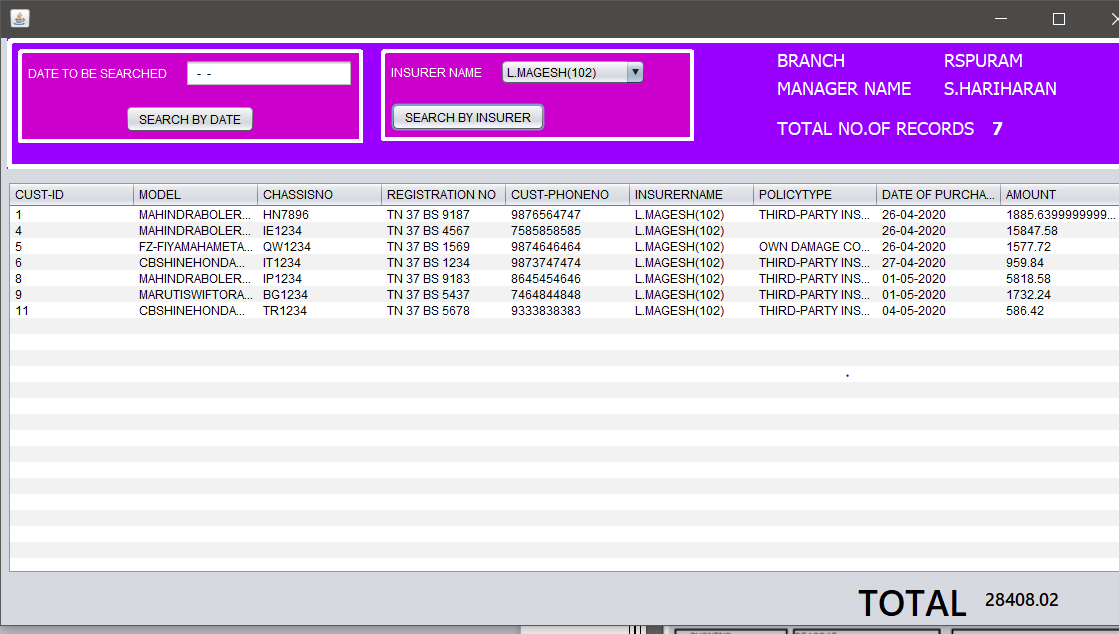


**REPORT GENERATED:**

POLICY GENERATION:



POLICY PURCHASED DETAILS REPORT FOR A PARTICULAR INSURERNAME:



POLICY PURCHASED DETAILS REPORT FOR A PARTICULAR DATE:

