INTRODUCTION

JAVA is a programming language created by James Gosling from Sun Microsystems(sun) in 1991.The target of Java is to write the program once and run on multiple operating systems.in this project we are using the jswing ,thread,class ,jbdc and etc features which are inbuilt in java in order to implement the linear regression method,here we already have a set of value stored in the data base wch contains details of the previous sales we use this data to predict the price of the house

The user can select the given situations such as “1BHK” “2BHK” “HOUSE” “SWIMMING” “BALCONY” depends on the conditions selected and the entered value of the square-feet we will predict the price of the house

SPECIFIC REQUIREMENTS

HARDWARE REQUIREMENTS:

- Minimum Windows 95 software
- IBM-compatible 486 system
- Hard Drive and Minimum of 8 MB memory
- A CD-ROM drive
- Mouse, keyboard.

SOFTWARE REQUIREMENTS:

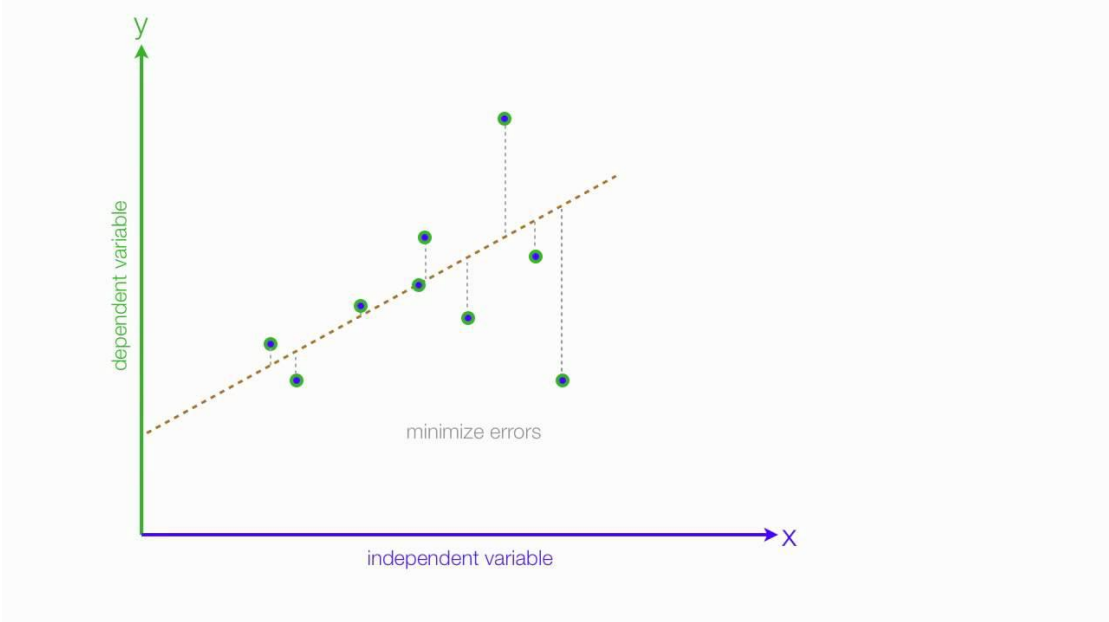
- Operating System
- Java SDK or JRE 1.6 or higher
- Java Servlet Container (Free Servlet Container available)
- Supported Database and library that supports the database connection with Java.

IMPLEMENTATION

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable *causes* the other (for example, higher SAT scores do not *cause* higher college grades), but that there is some significant association between the two variables. A scatterplot can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

A linear regression line has an equation of the form $\boldsymbol{Y = a + bX}$, where $\boldsymbol{X}$ is the explanatory variable and $\boldsymbol{Y}$ is the dependent variable. The slope of the line is $\boldsymbol{b}$, and $\boldsymbol{a}$ is the intercept (the value of $\boldsymbol{y}$ when $\boldsymbol{x} = 0$).



Given a data set of n statistical units, a linear regression model assumes that the relationship between the dependent variable y_i and the p-vectorof regressors $\boldsymbol{x}_i$ is linear. This relationship is modeled through a *disturbance term* or *error variable* ε_i — an unobserved random variable that adds noise to the linear relationship between the dependent variable and regressors. Thus the model takes the form

$$y_i = \beta_0 \mathbf{1} + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\top} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where ^{T} denotes the transpose, so that $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$ is the inner product between vectors $\boldsymbol{x}_i$ and $\boldsymbol{\beta}$.

Often these n equations are stacked together and written in vector form as

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

$$X = \begin{pmatrix} \mathbf{x}_1^{\top} \\ \mathbf{x}_2^{\top} \\ \vdots \\ \mathbf{x}_n^{\top} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

SOURCE CODE

JDBC.java:

```
//STEP 1. Import required packages
import java.sql.*;

public class JDBC    {
    // JDBC driver name and database URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/Java";
  static double x[]=new double[1000];
  static double y[]=new double[1000];
  static int z[]= new int[1000];
  static int a[]=new int[1000];
  static int s[]=new int[1000];
    //    Database credentials
    static final String USER = "staff";
    static final String PASS = "qwerty";
      int n=0;
    int nvalue(){
      return n;
}
    double[] yvalue(){
          return(y);

    }
    double[] xvalue(){
          return(x);

    }
      int[] zvalue(){
          return(z);
      }
```

```
    int[] bvalue(){
      return(a);

}
    int[] svalue(){
          return(s);
    }
public    void dbms() {
Connection conn = null;
Statement stmt = null;
try{
     //STEP 2: Register JDBC driver
     Class.forName("com.mysql.jdbc.Driver");//jar

     //STEP 3: Open a connection
     System.out.println("Connecting to a selected database...");
     conn = DriverManager.getConnection(DB_URL, USER, PASS);
     System.out.println("Connected database successfully...");

     //STEP 4: Execute a query
     System.out.println("Creating statement...");
     stmt = conn.createStatement();

     String sql = "SELECT * FROM house";
     ResultSet rs = stmt.executeQuery(sql);

     //STEP 5: Extract data from result set
     while(rs.next()){
          //Retrieve by column name
          String type    = rs.getString("type");
          if("1BHK".equals(type))
          {
               z[n]=1;
          }
```

```
            if("2BHK".equals(type))
            {
                z[n]=2;
            }
            if("H".equals(type))
            {
                z[n]=3;
            }


            a[n] = rs.getInt("balcony");

            x[n] = rs.getFloat("sq_feet");
            y[n] = rs.getFloat("price");
            s[n] = rs.getInt("swimming");
            System.out.print("type: " + type);

            System.out.print(", sq: " + x[n]);
           System.out.println(", price: " + y[n]);
           ++n;
        }
            // x[n]=sq;
                // y[n]=price;
            //Display values



        rs.close();
    }catch(SQLException se){
        //Handle errors for JDBC
        se.printStackTrace();
    }catch(Exception e){
        //Handle errors for Class.forName
        e.printStackTrace();
```

```
    }finally{
        //finally block used to close resources
        try{
            if(stmt!=null)
                conn.close();
        }catch(SQLException se){
        }// do nothing
        try{
            if(conn!=null)
                conn.close();
        }catch(SQLException se){
            se.printStackTrace();
        }//end finally try
    }//end try
    System.out.println("Goodbye!n="+n);
}//end main
    public static void main(String args[])
    {
         JDBC j=new JDBC();
         j.dbms();
    }
}//end JDBCExample
```

LinearRegression.java:

```
public class LinearRegression {

      public static void main(String[] args) {
         NewJFrame f=new NewJFrame();
                    f.setVisible(true);
            }
        }
```

Lm.java:

```
public class Lm extends Thread {
    double qw;
    Lm(double x)
    {
        this.qw=x;
    }

    double calc(int action1,int action2,int action3) throws InterruptedException
    {
        JDBC j=new JDBC();
        j.dbms();
      int n= j.nvalue();
      int ni=0;
        int MAXN = 1000;
        int z[]=new int[1000];
        int b[]=new int[1000];
        int s[]=new int[1000];
        int i=0;
        double y[]=new double[1000];
        double x[] = new double[1000];
          x = j.xvalue();
        double x1[] = new double[1000];
          y = j.yvalue();
        double y1[]=new double[1000];
      z=j.zvalue();
        b=j.bvalue();
        s=j.svalue();
      for(i=0;i<n;++i)
      {
            System.out.println(x[i]+","+y[i]+","+z[i]+","+b[i]);
      }
      if(action1==0)
```

```
{
        for(i=0;i<n;++i)
        {
                if(z[i]==1)
                {
                        x1[ni]=x[i];
                        y1[ni]=y[i];
                        ++ni;
                }
        }
         x=x1;
        y=y1;
        n=ni;
}
else if(action1==1)
{
        for(i=0;i<n;++i)
        {
                if(z[i]==2)
                {
                        x1[ni]=x[i];
                        y1[ni]=y[i];
                        ++ni;
                }
        }
        x=x1;
        y=y1;
        n=ni;
}
else if(action1==2)
{
        for(i=0;i<n;++i)
        {
                if(z[i]==3)
```

```
                {
                        x1[ni]=x[i];
                        y1[ni]=y[i];
                        ++ni;
                }
        }
        x=x1;
        y=y1;
        n=ni;
}
ni=0;
if(action2==1)
{
        for(i=0;i<n;++i)
        {
                if(b[i]==1)
                {
                        x1[ni]=x[i];
                        y1[ni]=y[i];
                        ++ni;
                }
        }
        x=x1;
        y=y1;
        n=ni;
}
ni=0;
if(action3==1)
{
        for(i=0;i<n;++i)
        {
                if(s[i]==1)
                {
                        x1[ni]=x[i];
```

```
                y1[ni]=y[i];
                ++ni;
            }
        }
        x=x1;
        y=y1;
        n=ni;

    }
            for(i=0;i<n;++i)
    {
        System.out.println("#"+x[i]+","+y[i]+","+z[i]+","+b[i]);
    }
    // first pass: read in data, compute xbar and ybar
    double sumx = 0.0, sumy = 0.0, sumx2 = 0.0;
    while(i!=n) {
      //    x[n] = s[i];
          //y[n] = r[i];
          sumx    += x[n];
          sumx2 += x[n] * x[n];
          sumy    += y[n];
          i++;
    }
    double xbar = sumx / n;
    double ybar = sumy / n;

    // second pass: compute summary statistics
    double xxbar = 0.0, yybar = 0.0, xybar = 0.0;
    for ( i = 0; i < n; i++) {
          xxbar += (x[i] - xbar) * (x[i] - xbar);
          yybar += (y[i] - ybar) * (y[i] - ybar);
          xybar += (x[i] - xbar) * (y[i] - ybar);
    }
    double beta1 = xybar / xxbar;
```

```
            double beta0 = ybar - beta1 * xbar;

            // print results
            System.out.println("y      = " + beta1 + " * x + " + beta0);

System.out.println("x="+qw+","+"y?"+(beta1*qw+beta0));
            // analyze results-
            int df = n - 2;
            double rss = 0.0;           // residual sum of squares
            double ssr = 0.0;           // regression sum of squares
            for ( i = 0; i < n; i++) {
                double fit = beta1*x[i] + beta0;
                rss += (fit - y[i]) * (fit - y[i]);
                ssr += (fit - ybar) * (fit - ybar);
            }
            double R2       = ssr / yybar;
            double svar    = rss / df;
            double svar1 = svar / xxbar;
            double svar0 = svar/n + xbar*xbar*svar1;
              System.out.println("R^2                          = " + R2);
              System.out.println("std error of beta_1 = " + Math.sqrt(svar1));
              System.out.println("std error of beta_0 = " + Math.sqrt(svar0));
            svar0 = svar * sumx2 / (n * xxbar);
              System.out.println("std error of beta_0 = " + Math.sqrt(svar0));

              System.out.println("SSTO = " + yybar);
              System.out.println("SSE    = " + rss);
            System.out.println("SSR    = " + ssr);
            double fina1=(double)(beta1*qw+beta0);
            return(fina1);
        }
}
```

Newframe.java

```
*/
public class NewJFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jTextField1 = new javax.swing.JTextField();
        jPasswordField1 = new javax.swing.JPasswordField();
        jLabel3 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setBackground(new java.awt.Color(153, 51, 0));

        jLabel1.setText("Username:");
```

```
        jLabel2.setText("Password:");

        jButton1.setBackground(new java.awt.Color(51, 255, 0));
        jButton1.setForeground(new java.awt.Color(153, 153, 153));
        jButton1.setText("LOGIN");
        jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jButton1MouseClicked(evt);
            }
        });

        jPasswordField1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jPasswordField1ActionPerformed(evt);
            }
        });

        jLabel3.setFont(new java.awt.Font("Tempus Sans ITC", 1, 18)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(204, 0, 204));
        jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        jLabel3.setText("HOUSE PRICE PREDICTOR");

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(108, 108, 108)
```

```
                                .addGroup(layout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.TRAILING)
                                        .addComponent(jLabel2)
                                        .addComponent(jLabel1))
                                .addGap(78, 78, 78)
                                .addGroup(layout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.LEADING, false)
                                        .addComponent(jTextField1,
javax.swing.GroupLayout.DEFAULT_SIZE, 113, Short.MAX_VALUE)
                                        .addComponent(jPasswordField1)))
                        .addGroup(layout.createSequentialGroup()
                                .addGap(154, 154, 154)
                                .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(0, 87, Short.MAX_VALUE)
                .addComponent(jLabel3)
                .addGap(83, 83, 83))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(37, 37, 37)
                .addComponent(jLabel3)
                .addGap(45, 45, 45)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
                        .addComponent(jLabel1)
```

```
                                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGap(27, 27, 27)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
                                .addComponent(jLabel2)
                                .addComponent(jPasswordField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGap(24, 24, 24)
                        .addComponent(jButton1)
                        .addContainerGap(73, Short.MAX_VALUE))
            );

            pack();
        }// </editor-fold>

        private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
            // TODO add your handling code here:
            try{
            String x = jTextField1.getText();

              String y =jPasswordField1.getText();
              if(x.equals("123")&&y.equals("123"))
              {
                      GUI g=new GUI();
                      g.setVisible(true);
              }
            }
            catch(Exception e)
            {
```

```
                System.out.println("REad errro");

            }
        }

        private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent evt)
{
            // TODO add your handling code here:
        }

        /**
         * @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(NewJFrame.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
            } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
            } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
            } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
            }
            //</editor-fold>

            /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
                    new NewJFrame().setVisible(true);
                }
            });
        }
  // Variables declaration - do not modify
        private javax.swing.JButton jButton1;
        private javax.swing.JLabel jLabel1;
        private javax.swing.JLabel jLabel2;
        private javax.swing.JLabel jLabel3;
        private javax.swing.JPasswordField jPasswordField1;
        private javax.swing.JTextField jTextField1;
        // End of variables declaration
}
```

SNAPSHOTS

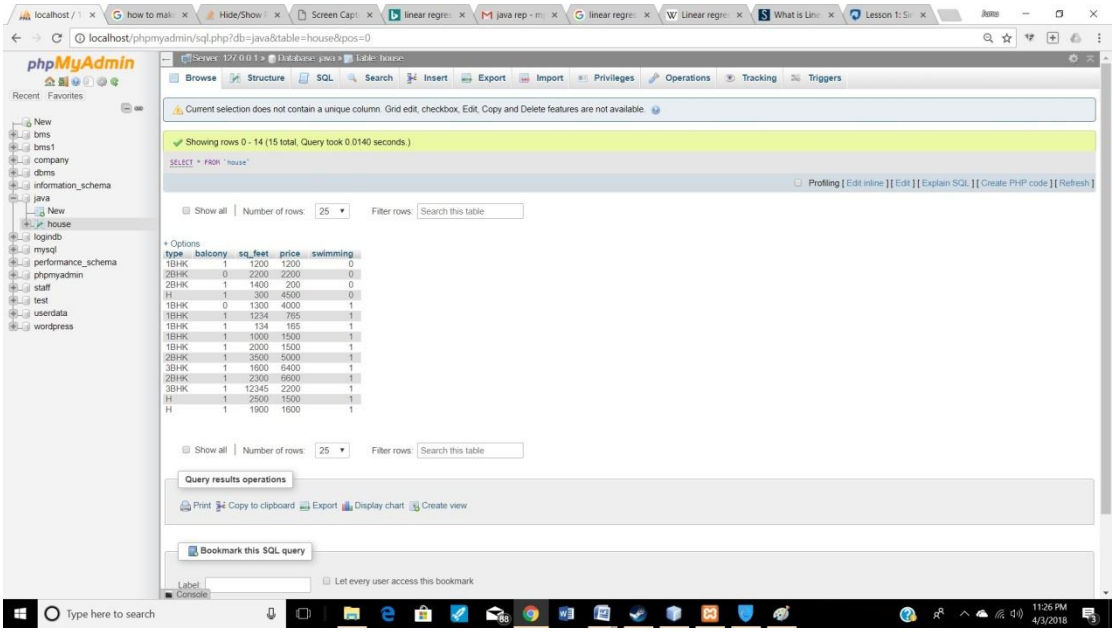


Fig1.The values presenting database which is used for prediction.

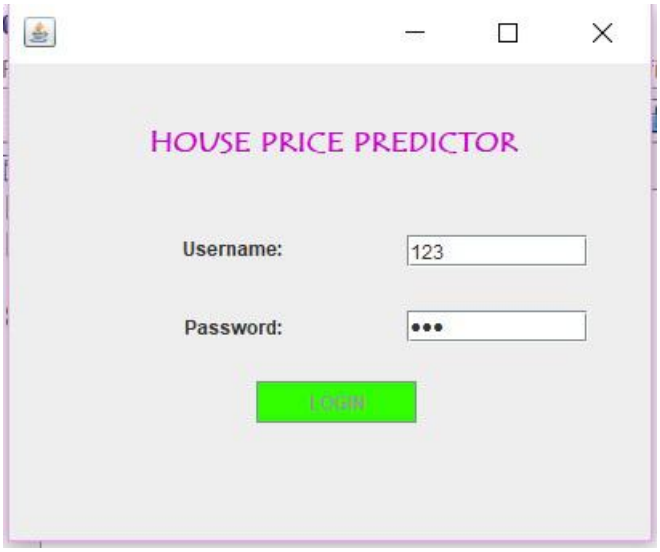
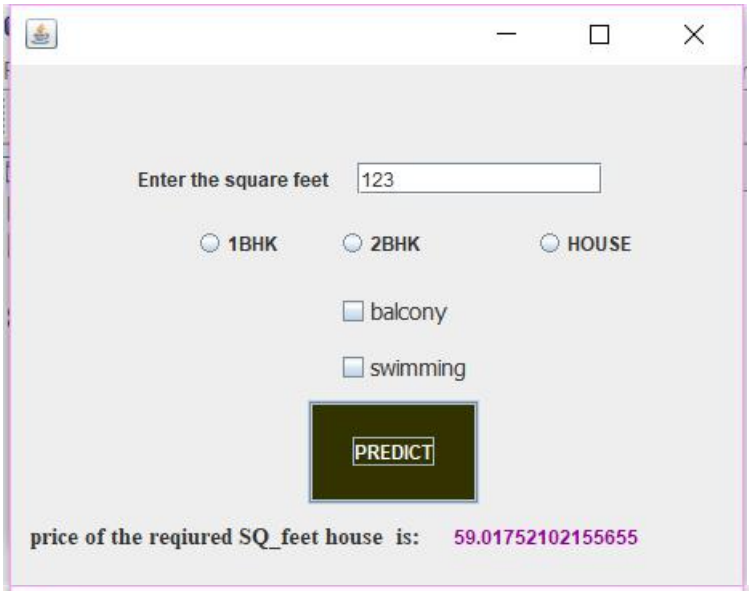
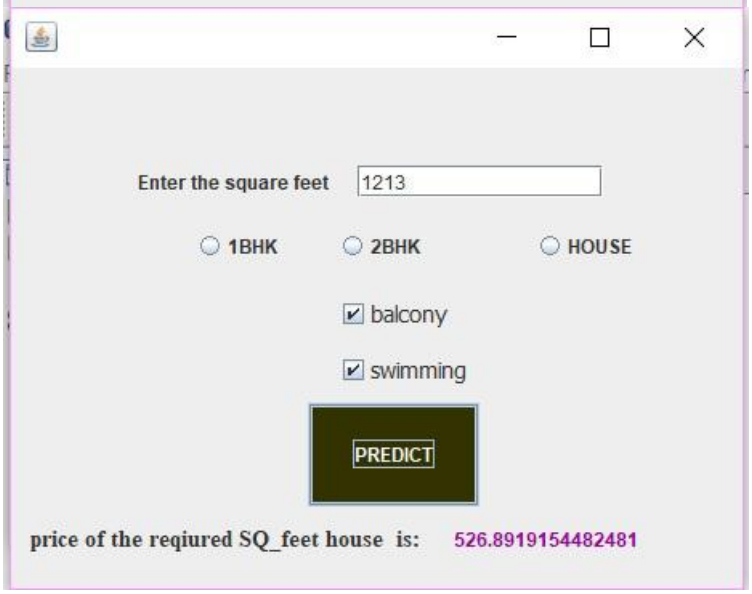
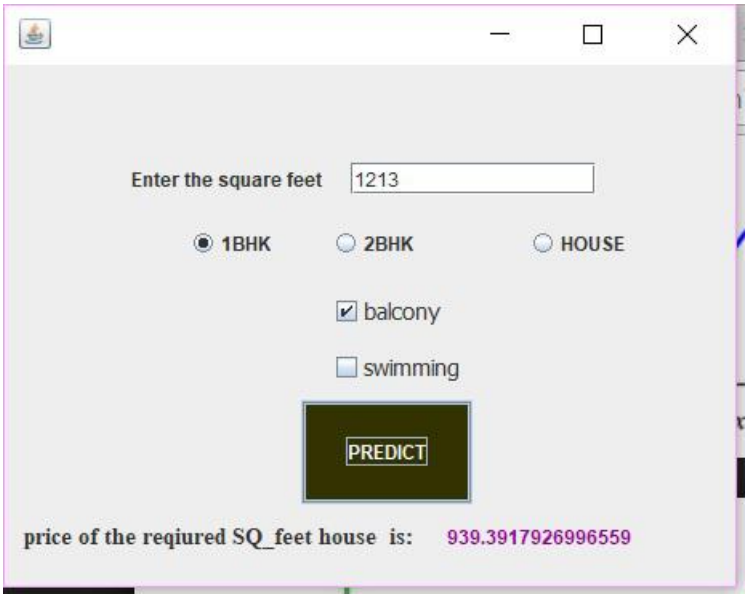


Fig2.login


Enter the square feet
123
1BHK
2BHK
HOUSE
balcony
swimming
PREDICT
price of the reqiured SQ_feet house is:
59.01752102155655


Enter the square feet
1213
1BHK
2BHK
HOUSE
balcony
swimming
PREDICT
price of the reqiured SQ_feet house is:
526.8919154482481


Enter the square feet
1213
1BHK
2BHK
HOUSE
balcony
swimming
PREDICT
price of the reqiured SQ_feet house is:
939.3917926996559

Java console output:

```
Connecting to a selected database...
Connected database successfully...
Creating statement...
type: 1BHK, sq: 1200.0, price: 1200.0
type: 2BHK, sq: 2200.0, price: 2200.0
type: 2BHK, sq: 1400.0, price: 200.0
type: H, sq: 300.0, price: 4500.0
type: 1BHK, sq: 1300.0, price: 4000.0
type: 1BHK, sq: 1234.0, price: 765.0
type: 1BHK, sq: 134.0, price: 165.0
type: 1BHK, sq: 1000.0, price: 1500.0
type: 1BHK, sq: 2000.0, price: 1500.0
type: 2BHK, sq: 3500.0, price: 5000.0
type: 3BHK, sq: 1600.0, price: 6400.0
type: 2BHK, sq: 2300.0, price: 6600.0
type: 3BHK, sq: 12345.0, price: 2200.0
type: H, sq: 2500.0, price: 1500.0
type: H, sq: 1900.0, price: 1600.0
Goodbye!n=15
1200.0,1200.0,1,1
2200.0,2200.0,2,0
1400.0,200.0,2,1
300.0,4500.0,3,1
1300.0,4000.0,1,0
1234.0,765.0,1,1
134.0,165.0,1,1
1000.0,1500.0,1,1
2000.0,1500.0,1,1
3500.0,5000.0,2,1
1600.0,6400.0,0,1
2300.0,6600.0,2,1
12345.0,2200.0,0,1
```

2500.0,1500.0,3,1

1900.0,1600.0,3,1

#1200.0,1200.0,1,1

#2200.0,2200.0,2,0

#1400.0,200.0,2,1

#300.0,4500.0,3,1

#1300.0,4000.0,1,0

#1234.0,765.0,1,1

#134.0,165.0,1,1

#1000.0,1500.0,1,1

#2000.0,1500.0,1,1

#3500.0,5000.0,2,1

#1600.0,6400.0,0,1

#2300.0,6600.0,2,1

#12345.0,2200.0,0,1

#2500.0,1500.0,3,1

#1900.0,1600.0,3,1

y = 0.4798172440776955 * x + 0.0

x=123.0,y?59.01752102155655

R^2 = 0.2744713334390051

std error of beta_1 = 0.21636298677553228

std error of beta_0 = 787.9093907327132

std error of beta_0 = 0.0

SSTO = 1.6685245E8

SSE = 1.2105623556093508E8

SSR = 4.579621443906493E7

YES BALCONY

Yes swimming

Connecting to a selected database...

Connected database successfully...

Creating statement...

type: 1BHK, sq: 1200.0, price: 1200.0

type: 2BHK, sq: 2200.0, price: 2200.0

type: 2BHK, sq: 1400.0, price: 200.0

type: H, sq: 300.0, price: 4500.0

type: 1BHK, sq: 1300.0, price: 4000.0

type: 1BHK, sq: 1234.0, price: 765.0

type: 1BHK, sq: 134.0, price: 165.0

type: 1BHK, sq: 1000.0, price: 1500.0

type: 1BHK, sq: 2000.0, price: 1500.0

type: 2BHK, sq: 3500.0, price: 5000.0

type: 3BHK, sq: 1600.0, price: 6400.0

type: 2BHK, sq: 2300.0, price: 6600.0

type: 3BHK, sq: 12345.0, price: 2200.0

type: H, sq: 2500.0, price: 1500.0

type: H, sq: 1900.0, price: 1600.0

Goodbye!n=15

1200.0,1200.0,1,1

2200.0,2200.0,2,0

1400.0,200.0,2,1

300.0,4500.0,3,1

1300.0,4000.0,1,0

1234.0,765.0,1,1

134.0,165.0,1,1

1000.0,1500.0,1,1

2000.0,1500.0,1,1

3500.0,5000.0,2,1

1600.0,6400.0,0,1

2300.0,6600.0,2,1

12345.0,2200.0,0,1

2500.0,1500.0,3,1

1900.0,1600.0,3,1

#134.0,165.0,1,1

#1000.0,1500.0,2,0

#2000.0,1500.0,2,1

#3500.0,5000.0,3,1

#1600.0,6400.0,1,0

#2300.0,6600.0,1,1

#12345.0,2200.0,1,1

#2500.0,1500.0,1,1

#1900.0,1600.0,1,1

y = 0.4343709113340875 * x + 0.0

x=1213.0,y?526.8919154482481

R^2 = 0.2858102162372112

std error of beta_1 = 0.25952527295161465

std error of beta_0 = 1184.1767240939278

std error of beta_0 = 0.0

SSTO = 1.23697225E8

SSE = 8.834329437480704E7

SSR = 3.535393062519297E7

!BHK ONLY

YES BALCONY

Connecting to a selected database...

Connected database successfully...

Creating statement...

type: 1BHK, sq: 1200.0, price: 1200.0

type: 2BHK, sq: 2200.0, price: 2200.0

type: 2BHK, sq: 1400.0, price: 200.0

type: H, sq: 300.0, price: 4500.0

type: 1BHK, sq: 1300.0, price: 4000.0

type: 1BHK, sq: 1234.0, price: 765.0

type: 1BHK, sq: 134.0, price: 165.0

type: 1BHK, sq: 1000.0, price: 1500.0

type: 1BHK, sq: 2000.0, price: 1500.0

type: 2BHK, sq: 3500.0, price: 5000.0

type: 3BHK, sq: 1600.0, price: 6400.0

type: 2BHK, sq: 2300.0, price: 6600.0

type: 3BHK, sq: 12345.0, price: 2200.0

type: H, sq: 2500.0, price: 1500.0

type: H, sq: 1900.0, price: 1600.0

Goodbye!n=15

1200.0,1200.0,1,1

2200.0,2200.0,2,0

1400.0,200.0,2,1

300.0,4500.0,3,1

1300.0,4000.0,1,0

1234.0,765.0,1,1

134.0,165.0,1,1

1000.0,1500.0,1,1

2000.0,1500.0,1,1

3500.0,5000.0,2,1

1600.0,6400.0,0,1

2300.0,6600.0,2,1

12345.0,2200.0,0,1

2500.0,1500.0,3,1

1900.0,1600.0,3,1

#1200.0,1200.0,1,1

#1234.0,765.0,2,0

#134.0,165.0,2,1

#2000.0,1500.0,3,1

y = 0.7744367623245308 * x + 0.0

x=1213.0,y?939.3917926996559

R^2 = 0.9730962752705767

std error of beta_1 = 0.09105404905885048

std error of beta_0 = 120.28712029042616

std error of beta_0 = 0.0

SSTO = 4302450.0

SSE = 115751.93046210763

SSR = 4186698.069537893

CONCLUSION

- Our project was house price predection using linear regression.It has been matter of immense pleasure,honour and challenge to complete it successfully.
- While developing this project we have learnt about how to implement the jswing,multithreading ,exception handling and jdbc connection.
- Also how to use the database through phpmyadmin and to access the SQL database thorough it.
- In this project we also learned how the linear regression works and what are its advantages and disadvantages and which are the situation we have to use it.
- This project helped us to have fun with java and also thought us the value of information i.e it can be used to predict inferences.

BIBILOGRAPHY

- Wikipedia.org.
- www.tutorialspoint.com.
- Java-The Complete Reference by Herbert Schildt.
- Class Notes.