JAVA SWING BASED – FITNESS GUARDIAN-SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of
BACHELOR OF TECHNOLOGY
IN

INFORMATION TECHNOLOGY

By

KHIZRA NAUSHAD <1602-19-737-076>

Under the Guidance of

B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2020-2021

BONAFIDE CERTIFICATE

This to Certify that the project report titled **“FITNESS GUARDIAN”** project work of Ms KHIZRA NAUSHAD bearing Roll.no:1602-19-737-076 who carried out this project under my supervision in the IV semester for the academic year 2020-2021.

Signature external examiner *Signature internal examiner*

ABSTRACT:

The objective of this project is developing the application dietry suggestion to the users based on their health conditions for adults between 20 – 60 years. This application utilizes a computer analysis of a, body mass index to provide a menu of a specific number of calories to maintain a reasonable weight. The system further provides instructions to maintain a healthy nutrition balance.

Introduction:

REQUIREMENT ANALYSIS

List of tables:

- USER
- USER DETAILS
- CALCULATES
- BMI
- GIVE PLAN
- DIET PLAN

List of attributes with their domain types:

USERS

- USER ID: user_id number (20)
- USER NAME: user_name varchar2(40)
- USER PASSWORD: user_password varchar2(10)

USER DETAILS

- DETAILS ID: details_id number(20)
- USER GENDER: gender varchar(6)
- USER AGE: age number(5)
- USER HEIGHT: height number(5)
- USER WEIGHT: weight number(5)
- EXERCISE TIME : exercise_time number(20)
- HEALTH CONDITION : health_condition varchar(40)

CALCULATES

- DETAIL ID : details_id number(20)
- BMI NUMBER : bmi_number number(20)

BMI

- BMI ID : bmi_id number(20)

- BMI CATEGORY : bmi_category varchar(40)

GIVE PLAN

- DETAIL ID : details_id number(20)
- BMI ID : bmi_id number(20)
- TYPE OF DIET : type_of_diet varchar(100)

DIET PLAN

- DIET ID : diet_id number(20)
- NAME : name varchar(40)
- MORNING DIET : morning_diet varchar(200)
- LUNCH DIET : lunch_diet varchar(200)
- DINNER DIET : dinner_diet varchar(200)
- INSTRUCTIONS : general_intructions varchar(200)
- WORKOUT : workout varchar(200)

PROJECT:

This project helps to store data in a efficient way and it can be achieved through various sql commands and we can also store this for any future use and also we can save our data in a many different areas so we cannot lost all the data at once. The quality and product details are must in now a days because quality matters every where, these project stores details and feedback and testing details in database so that whenever it is necessary to know how and when a product can be used.

ARCHITECTURE AND TECHNOLOGY USED:

SOFTWARE USED:

Java Eclipse, Oracle 11g Database, Java SE version 8, SQL Plus.

Java SWING:

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

SQL:

Structure Query Language (SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

Java-SQL Connectivity using JDBC:

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

The connection to the database can be performed using Java programming (JDBC API) as:

```
try{
                    //step1 load the driver class
                    Class.forName("oracle.jdbc.driver.OracleDriver");   // or
Class.forName("oracle.jdbc.OracleDriver");

                    //step2 create  the connection object
                    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","it19737076"
,"nida");

                    //step3 create the statement object
                    Statement stmt=con.createStatement();

                    //step4 execute query
                    ResultSet rs=stmt.executeQuery("select * from sailors");
                    while(rs.next())
                                 System.out.println(rs.getInt(1)+"
"+rs.getString(2)+"  "+rs.getString(3));

                    //step5 close the connection object
                    con.close();
```

```
                }catch(Exception e){ System.out.println(e);}
        }
```

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

Table Created in SQL for above mentioned purpose is as:

CREATE TABLE users (user_id number(20), user_name varchar2(40), user_password varchar2(20), primary key(user_id));

CREATE TABLE user_details(details_id number(20), gender varchar(6), age number(5), height number(5), weight number(5), exercise_time number(20), health_condition varchar(40),primary key(details_id));

alter table user_details modify height number(5) not null;

alter table user_details modify weight number(5) not null;

CREATE TABLE calculates(details_id number(20), bmi_number number(20),foreign key(details_id) references user_details);

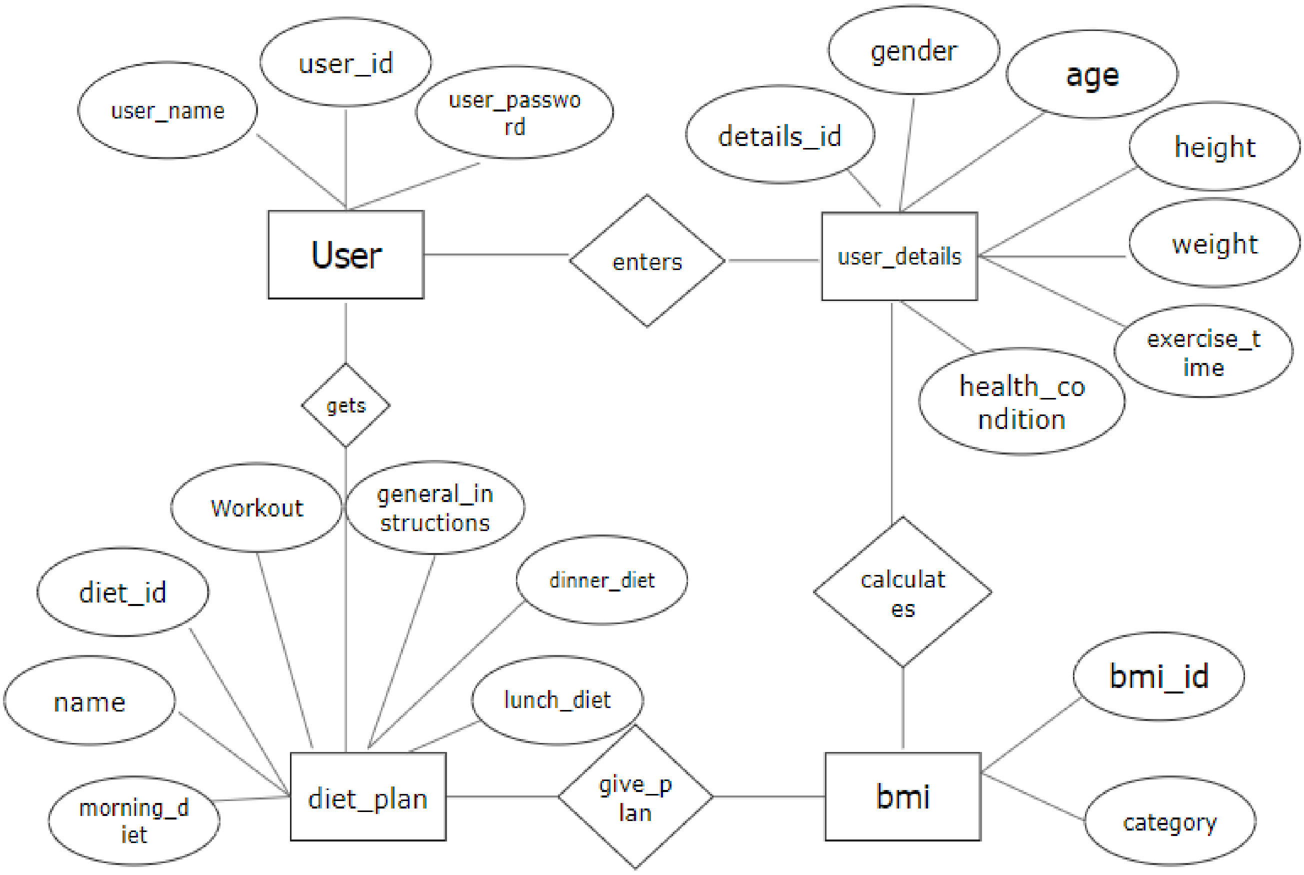
CREATE TABLE bmi(bmi_id number(20), bmi_category varchar(40),primary key(bmi_id));

CREATE TABLE give_plan(bmi_id number(20), details_id number(20), type_of_diet varchar(100), foreign key(details_id) references user_details, foreign key(bmi_id) references bmi));

CREATE TABLE diet_plan(diet_id number(20), name varchar(40), morning_diet varchar(200), lunch_diet varchar(200), dinner_diet varchar(200), general_instructions varchar(200), workout varchar(200),primary key(diet_id));

DESIGN:

ER DIAGRAM:



Database Design:

```
SQL> select * from tab;

TNAME                    TABTYPE  CLUSTERID

------------------------------ ------- ----------

BMI                      TABLE

CALCULATES               TABLE

DIET_PLAN                TABLE

GIVE_PLAN                TABLE

USERS                    TABLE

USER_DETAILS             TABLE
```

```
6 rows selected.

SQL> desc users;

 Name                                      Null?    Type

 ----------------------------------------- -------- ----------------------------

 USER_ID                                   NOT NULL NUMBER(20)

 USER_NAME                                          VARCHAR2(40)

 USER_PASSWORD                                      VARCHAR2(20)

SQL> desc user_details;

 Name                                      Null?    Type

 ----------------------------------------- -------- ----------------------------

 DETAILS_ID                                NOT NULL NUMBER(20)

 GENDER                                             VARCHAR2(6)

 AGE                                                NUMBER(5)

 HEIGHT                                    NOT NULL NUMBER(5)

 WEIGHT                                    NOT NULL NUMBER(5)

 EXERCISE_TIME                                      NUMBER(20)

 HEALTH_CONDITION                                   VARCHAR2(40)

SQL> desc calculates;

 Name                                      Null?    Type

 ----------------------------------------- -------- ----------------------------

 DETAILS_ID                                         NUMBER(20)

 BMI_NUMBER                                         NUMBER(20)

SQL> desc bmi;

 Name                                      Null?    Type

 ----------------------------------------- -------- ----------------------------

 BMI_ID                                    NOT NULL NUMBER(20)

 BMI_CATEGORY                                       VARCHAR2(40)
```

```
SQL> desc give_plan;

 Name                                      Null?    Type

 ----------------------------------------- -------- ----------------------------

 BMI_ID                                             NUMBER(20)

 DETAILS_ID                                         NUMBER(20)

 TYPE_OF_DIET                                       VARCHAR2(100)

SQL> desc diet_plan;

 Name                                      Null?    Type

 ----------------------------------------- -------- ----------------------------

 DIET_ID                                   NOT NULL NUMBER(20)

 NAME                                               VARCHAR2(40)

 MORNING_DIET                                       VARCHAR2(200)

 LUNCH_DIET                                         VARCHAR2(200)

 DINNER_DIET                                        VARCHAR2(200)

 GENERAL_INSTRUCTIONS                               VARCHAR2(200)

 WORKOUT                                            VARCHAR2(200)
```

Implementation:

Program:

User Interface:

```
import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.awt.Color;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class ProjectUI extends JFrame {
```

```java
/**
 * 
 */
private static final long serialVersionUID = 1L;
private JMenu mnuuser,mnuuserdet,mnucalcu,mnubmi,mnugiveplan,mnudietplan;
private JMenuBar mnuBar;
private JMenuItem miInsert1,miUpdate1,midelete1,miView1;
private JMenuItem miInsert2,miUpdate2,midelete2,miView2;
private JMenuItem miInsert3,miUpdate3,midelete3,miView3;
private JMenuItem miInsert4,miUpdate4,midelete4,miView4;
private JMenuItem miInsert5,miUpdate5,midelete5,miView5;
private JMenuItem miInsert6,miUpdate6,midelete6,miView6;

private JTextField txtField;

static JPanel p1;

void initialize()
{
	p1=new JPanel();
	//p1.setBackground(Color.PINK);
	mnuuser=new JMenu("User Login");
	mnuuserdet=new JMenu("User details");
	mnucalcu=new JMenu("Calculation");
	mnubmi=new JMenu("BMI");
	mnugiveplan=new JMenu("Diet Type");
	mnudietplan=new JMenu("Diet Plan");

	mnuBar=new JMenuBar();

	miInsert1=new JMenuItem("Insert");
	miUpdate1=new JMenuItem("Update");
	midelete1=new JMenuItem("Delete");
	miView1=new JMenuItem("View");

	miInsert2=new JMenuItem("Insert");
	miUpdate2=new JMenuItem("Update");
	midelete2=new JMenuItem("Delete");
	miView2=new JMenuItem("View");

	miInsert3=new JMenuItem("Insert");
	miUpdate3=new JMenuItem("Update");
	midelete3=new JMenuItem("Delete");
	miView3=new JMenuItem("View");

	miInsert4=new JMenuItem("Insert");
	miUpdate4=new JMenuItem("Update");
	midelete4=new JMenuItem("Delete");
	miView4=new JMenuItem("View");

	miInsert5=new JMenuItem("Insert");
	miUpdate5=new JMenuItem("Update");
	midelete5=new JMenuItem("Delete");
	miView5=new JMenuItem("View");

	miInsert6=new JMenuItem("Insert");
	miUpdate6=new JMenuItem("Update");
	midelete6=new JMenuItem("Delete");
	miView6=new JMenuItem("View");
```

```
        txtField = new JTextField("         FITNESS  GUARDIAN");
        txtField.setBackground(Color.PINK);
        txtField.setFont(new Font("Serif", Font.PLAIN, 25));
        txtField.setEditable(false);

    }
    void addComponentsToFrame()
    {
        mnuuser.add(miInsert1);
        mnuuser.add(miUpdate1);
        mnuuser.add(midelete1);
        mnuuser.add(miView1);


        mnuuserdet.add(miInsert2);
        mnuuserdet.add(miUpdate2);
        mnuuserdet.add(midelete2);
        mnuuserdet.add(miView2);

        mnucalcu.add(miInsert3);
        mnucalcu.add(miUpdate3);
        mnucalcu.add(midelete3);
        mnucalcu.add(miView3);

        mnubmi.add(miInsert4);
        mnubmi.add(miUpdate4);
        mnubmi.add(midelete4);
        mnubmi.add(miView4);

        mnugiveplan.add(miInsert5);
        mnugiveplan.add(miUpdate5);
        mnugiveplan.add(midelete5);
        mnugiveplan.add(miView5);

        mnudietplan.add(miInsert6);
        mnudietplan.add(miUpdate6);
        mnudietplan.add(midelete6);
        mnudietplan.add(miView6);



        mnuBar.add(mnuuser);
        mnuBar.add(mnuuserdet);
        mnuBar.add(mnucalcu);
        mnuBar.add(mnubmi);
        mnuBar.add(mnugiveplan);
        mnuBar.add(mnudietplan);

        setJMenuBar(mnuBar);


        p1.setLayout(new BorderLayout());
        p1.add(txtField,BorderLayout.CENTER);

        this.setLayout(new BorderLayout());
        add(p1,BorderLayout.CENTER);

    }
    void register()
    {
        User t1=new
User(p1,ProjectUI.this,miInsert1,miUpdate1,midelete1,miView1);
        t1.regis_fitness();
        Userdetail t2=new
Userdetail(p1,ProjectUI.this,miInsert2,miUpdate2,midelete2,miView2);
        t2.regis_fitness();
        Calculate t3=new
Calculate(p1,ProjectUI.this,miInsert3,miUpdate3,midelete3,miView3);
        t3.regis_fitness();
        Bmi t4=new Bmi(p1, ProjectUI.this, miInsert4, miUpdate4, midelete4,
miView4);
        t4.regis_fitness();
        GivePlan t5=new
GivePlan(p1,ProjectUI.this,miInsert5,miUpdate5,midelete5,miView5);
        t5.regis_fitness();
        DietPlan t6=new DietPlan(p1, ProjectUI.this, miInsert6,miUpdate6,
midelete6, miView6);
        t6.regis_fitness();

         addWindowListener(new WindowAdapter(){
                public void windowClosing(WindowEvent evt)
                {

                        ImageIcon icon = new
ImageIcon("src/t2s1.jpg");
                     int
a=JOptionPane.showConfirmDialog(ProjectUI.this,"Are you sure?", "This will close",
JOptionPane.OK_CANCEL_OPTION, JOptionPane.INFORMATION_MESSAGE, icon);
                     if(a==JOptionPane.YES_OPTION)

ProjectUI.this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                }
             });


    }


    public ProjectUI()
    {
        initialize();
        addComponentsToFrame();
        register();
        pack();
        setTitle("FITNESS GUARDIAN");
        setSize(600,500);
        setVisible(true);
        //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}
```

Users Table:

```java
import java.awt.BorderLayout;
import java.awt.Color;

import  javax.swing.JTable;
import  javax.swing.JScrollPane;
import  javax.swing.table.DefaultTableModel;
import java.awt.FlowLayout;
import java.awt.GridLayout;
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 java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import  javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;


public class User {

        private JPanel p1;
        private JFrame frame;

        private JMenuItem miInsert1,miUpdate1,midelete1,miView1;
        private JLabel lbluserid,lblusername,lblpassword;
        private JTextField txtuserid,txtusername,txtpassword;
        private JButton btn;
        private JTextArea txtmsg;
        //private static Statement stmt;
        private Connection connection;
        private Statement statement;
        public User(JPanel p1,JFrame frame,JMenuItem miInsert1,JMenuItem
miUpdate1,JMenuItem midelete1,JMenuItem miView1) {
                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                }
                catch (Exception e)
                {
                        System.err.println("Unable to find and load driver");
                        System.exit(1);
                }
                connectToDB();
                this.frame=frame;
                this.p1=p1;
                this.miInsert1=miInsert1;
```

```
       this.midelete1=midelete1;
       this.miUpdate1=miUpdate1;
       this.miView1=miView1;
       txtuserid=new JTextField(20);
       txtusername=new JTextField(20);
       txtpassword=new JTextField(20);
       txtmsg=new JTextArea(8,50);
       lbluserid=new JLabel("User Id:");
       lblusername =new JLabel("User Name:");
       lblpassword=new JLabel("Password:");
       //queryHandler();
   }
   public void connectToDB()
  {
       try
       {
        connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","it19737076","ni
da");
        statement = connection.createStatement();

       }
       catch (SQLException connectException)
       {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
       }
  }
   public void regis_fitness() {
       miInsert1.addActionListener(new ActionListener() {
          public void actionPerformed(ActionEvent ae) {
             p1.removeAll();
             frame.invalidate();
             frame.validate();
             frame.repaint();
          //   p1.setBackground(Color.CYAN);
             JPanel p=new JPanel();
             txtuserid=new JTextField(20);
             txtusername=new JTextField(20);
             txtpassword=new JTextField(20);
             txtmsg=new JTextArea(8,50);
             btn=new JButton();
             //a grid of lbl and txtfield
             p.add(lbluserid);
             p.add(txtuserid);
             p.add(lblusername);
             p.add(txtusername);
             p.add(lblpassword);
             p.add(txtpassword);
             p.setLayout(new GridLayout(4,2));

             p1.add(p);
             p1.add(btn);
             p1.add(txtmsg);//msg text area added to panel
             btn.setText("SUBMIT");
             p1.setLayout(new FlowLayout());
             frame.add(p1,BorderLayout.CENTER);
```

```
                          frame.validate();

                          //register listener
                          btn.addActionListener(new ActionListener() {

                                 @Override
                                 public void actionPerformed(ActionEvent e) {
                                       // TODO Auto-generated method stub
                                       try {
                                              Statement statement =
connection.createStatement();
                                              String query= "INSERT INTO users
VALUES(" + txtuserid.getText() + ", " + "'" + txtusername.getText() + "'," + "'" +
txtpassword.getText() + "')";
                                              int i =
statement.executeUpdate(query);
                                                    txtmsg.append("\nInserted " +
i + " rows successfully");
                                       } catch (SQLException e1) {
                                              // TODO Auto-generated catch block
                                       //     e1.printStackTrace();
                                              txtmsg.append(e1.getMessage());
                                       }
                                 }
                          });

                    }
             });

             miUpdate1.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent ae) {
                    //     p1.setBackground(Color.LIGHT_GRAY);
                          p1.removeAll();
                          frame.invalidate();
                          frame.validate();
                          frame.repaint();
                          JPanel p=new JPanel();
                          txtuserid=new JTextField(20);
                          txtusername=new JTextField(20);
                          txtpassword=new JTextField(20);
                          txtmsg=new JTextArea(8,50);
                          txtmsg.setEditable(false);
                          btn=new JButton();
                          List idlist =new List(10);
                          try
                          {
                           ResultSet rs = statement.executeQuery("SELECT user_id
FROM users");

                            while (rs.next())
                            {
                                   idlist.add(rs.getString("user_id"));
                            }
                          }
                          catch (SQLException e)
                          {
                            txtmsg.append(e.getMessage());
                          }
                          p1.add(idlist);
                          p.add(lbluserid);
```

```
                                p.add(txtuserid);
                                p.add(lblusername);
                                p.add(txtusername);
                                p.add(lblpassword);
                                p.add(txtpassword);
                                p.setLayout(new GridLayout(4,2));
                                p1.add(p);
                                p1.add(btn);
                                p1.add(txtmsg);
                                btn.setText("MODIFY");
                                p1.setLayout(new FlowLayout());

                                frame.add(p1,BorderLayout.CENTER);
                                frame.validate();

                                idlist.addItemListener(new ItemListener()
                                {
                                      @Override
                                      public void itemStateChanged(ItemEvent arg0) {
                                            // TODO Auto-generated method stub

                                            try
                                            {
                                               ResultSet rs =
statement.executeQuery("SELECT * FROM users where user_id
="+idlist.getSelectedItem());
                                                  rs.next();

      txtuserid.setText(rs.getString("user_id"));

      txtusername.setText(rs.getString("user_name"));

      txtpassword.setText(rs.getString("user_password"));
                                            }
                                            catch (SQLException selectException)
                                            {

      txtmsg.append(selectException.getMessage());
                                            }
                                      }
                                });


                                btn.addActionListener(new ActionListener() {

                                      @Override
                                      public void actionPerformed(ActionEvent e) {
                                            // TODO Auto-generated method stub
                                            try
                                            {
                                                  Statement statement =
connection.createStatement();
                                                  int i =
statement.executeUpdate("UPDATE users "
                                                  + "SET user_name='" +
txtusername.getText() + "', "
                                                  + "user_password ='" +
txtpassword.getText()  + "'" +" WHERE user_id = "
                                                  + idlist.getSelectedItem());
```

```
                                              txtmsg.append("\nUpdated " + i + "
rows successfully");
                                              //idlist.removeAll();
                                              //loadSailors();
                                       }
                                       catch (SQLException insertException)
                                       {
      txtmsg.append(insertException.getMessage());
                                       }

                                 }
                          });
                    }
             });


             midelete1.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent ae) {
                    //     p1.setBackground(Color.BLUE);
                           p1.removeAll();
                           frame.invalidate();
                           frame.validate();
                           frame.repaint();
                           JPanel p=new JPanel();
                           txtuserid=new JTextField(20);
                           txtusername=new JTextField(20);
                           txtpassword=new JTextField(20);
                           txtmsg=new JTextArea(8,50);
                           btn=new JButton();
                           List idlist =new List(10);
                           try
                           {
                            ResultSet rs = statement.executeQuery("SELECT user_id
FROM users");
                             while (rs.next())
                             {
                                    idlist.add(rs.getString("user_id"));
                             }
                           }
                           catch (SQLException e)
                           {
                             txtmsg.append(e.getMessage());
                           }
                           p1.add(idlist);
                           p.add(lbluserid);
                           p.add(txtuserid);
                           p.add(lblusername);
                           p.add(txtusername);
                           p.add(lblpassword);
                           p.add(txtpassword);
                           p.setLayout(new GridLayout(4,2));
                           p1.add(p);
                           p1.add(btn);
                           p1.add(txtmsg);
                           btn.setText("DELETE");
                           p1.setLayout(new FlowLayout());

                           frame.add(p1,BorderLayout.CENTER);
```

```
                frame.validate();

                idlist.addItemListener(new ItemListener()
                {
                    @Override
                    public void itemStateChanged(ItemEvent arg0) {
                        // TODO Auto-generated method stub

                        try
                        {
                           ResultSet rs =
statement.executeQuery("SELECT * FROM users where user_id
="+idlist.getSelectedItem());
                            rs.next();

    txtuserid.setText(rs.getString("user_id"));

    txtusername.setText(rs.getString("user_name"));

    txtpassword.setText(rs.getString("user_password"));
                        }
                        catch (SQLException selectException)
                        {

    txtmsg.append(selectException.getMessage());
                        }
                    }
                });


                btn.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {
                        // TODO Auto-generated method stub
                        try
                        {
                            Statement statement =
connection.createStatement();
                            int i =
statement.executeUpdate("delete from users where user_id=" +
idlist.getSelectedItem());
                            txtmsg.append("\nDeleted " + i + "
rows successfully");
                            //idlist.removeAll();
                            //loadSailors();
                            txtuserid.setText(null);
                            txtusername.setText(null);
                            txtpassword.setText(null);
                            idlist.removeAll();
                        }
                        catch (SQLException insertException)
                        {

    txtmsg.append(insertException.getMessage());
                        }

                    }
                });
```

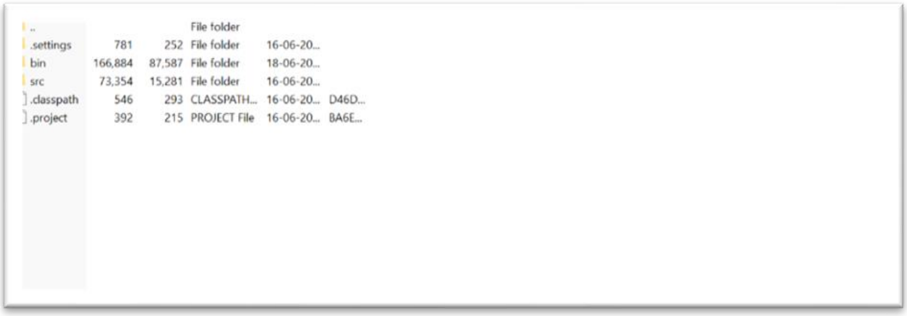
```
                    }
             });

        miView1.addActionListener(new ActionListener() {

              @Override
                    public void actionPerformed(ActionEvent e) {
                          // TODO Auto-generated method stub
                          p1.removeAll();
                          frame.invalidate();
                          frame.validate();
                          frame.repaint();
                          //p1.add(txtmsg);
                          frame.add(p1,BorderLayout.CENTER);
                          frame.validate();
                          JTable  j;
                          DefaultTableModel  S  =  new  DefaultTableModel();
                          j  =  new  JTable(S);
                     S.addColumn("User Id");
                          S.addColumn("User Name");
                          S.addColumn("Password");

                          try {
                                ResultSet rs=statement.executeQuery("select *
from users");
                                while(rs.next())
                                      S.addRow(new
Object[]{rs.getString("user_id"),rs.getString("user_name"),rs.getString("user_pass
word")});
                          } catch (SQLException e1) {
                                JOptionPane.showMessageDialog(frame,"Something
Went Wrong");
                          }
                          JScrollPane  sp  =  new  JScrollPane(j);

                          p1.add(sp,BorderLayout.CENTER);
                          p1.setBackground(Color.PINK);
                          frame.invalidate();
                          frame.validate();
                          frame.repaint();

                     }
              });
```

Main:

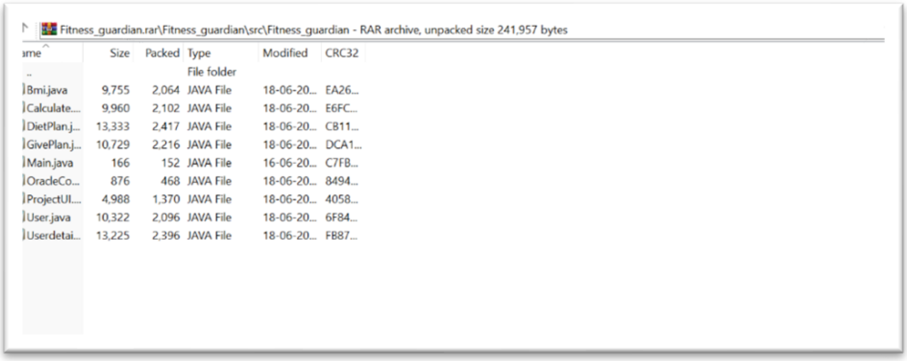
```
public class Main {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            new ProjectUI();

      }

}
```

GitHub links and folder structure

https://github.com/khizranaushad/DBMS-Project.git

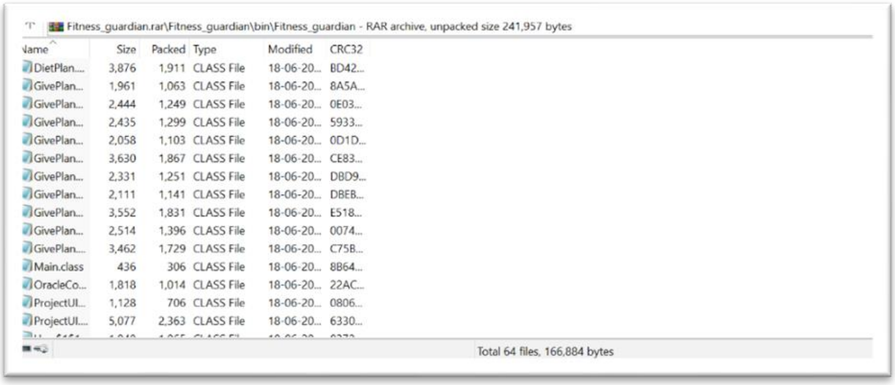


..			File folder		
.settings	781	252	File folder	16-06-20...	
bin	166,884	87,587	File folder	18-06-20...	
src	73,354	15,281	File folder	16-06-20...	
.classpath	546	293	CLASSPATH...	16-06-20...	D46D...
.project	392	215	PROJECT File	16-06-20...	BA6E...



Fitness_guardian.rar\Fitness_guardian\src\Fitness_guardian - RAR archive, unpacked size 241,957 bytes

ame	Size	Packed	Type	Modified	CRC32
..			File folder		
Bmi.java	9,755	2,064	JAVA File	18-06-20...	EA26...
Calculate....	9,960	2,102	JAVA File	18-06-20...	E6FC...
DietPlan.j...	13,333	2,417	JAVA File	18-06-20...	CB11...
GivePlan.j...	10,729	2,216	JAVA File	18-06-20...	DCA1...
Main.java	166	152	JAVA File	16-06-20...	C7FB...
OracleCo...	876	468	JAVA File	18-06-20...	8494...
ProjectUI....	4,988	1,370	JAVA File	18-06-20...	4058...
User.java	10,322	2,096	JAVA File	18-06-20...	6F84...
Userdetai...	13,225	2,396	JAVA File	18-06-20...	FB87...



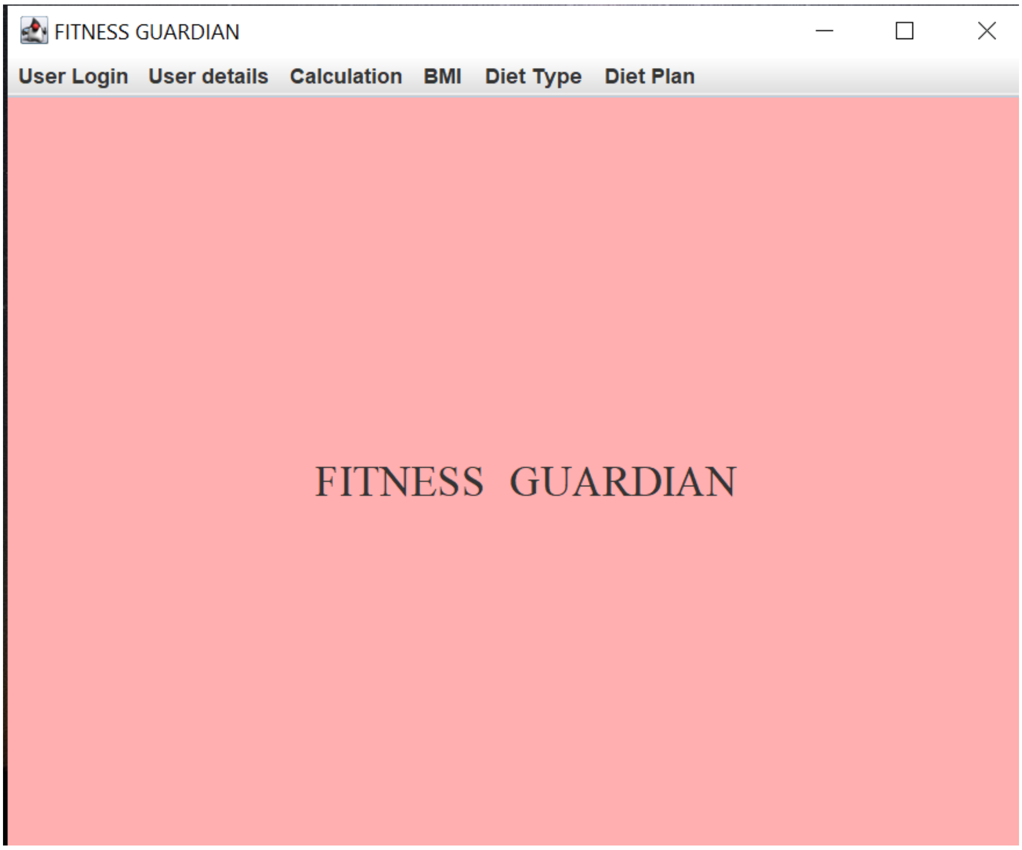
Fitness_guardian.rar\Fitness_guardian\bin\Fitness_guardian - RAR archive, unpacked size 241,957 bytes

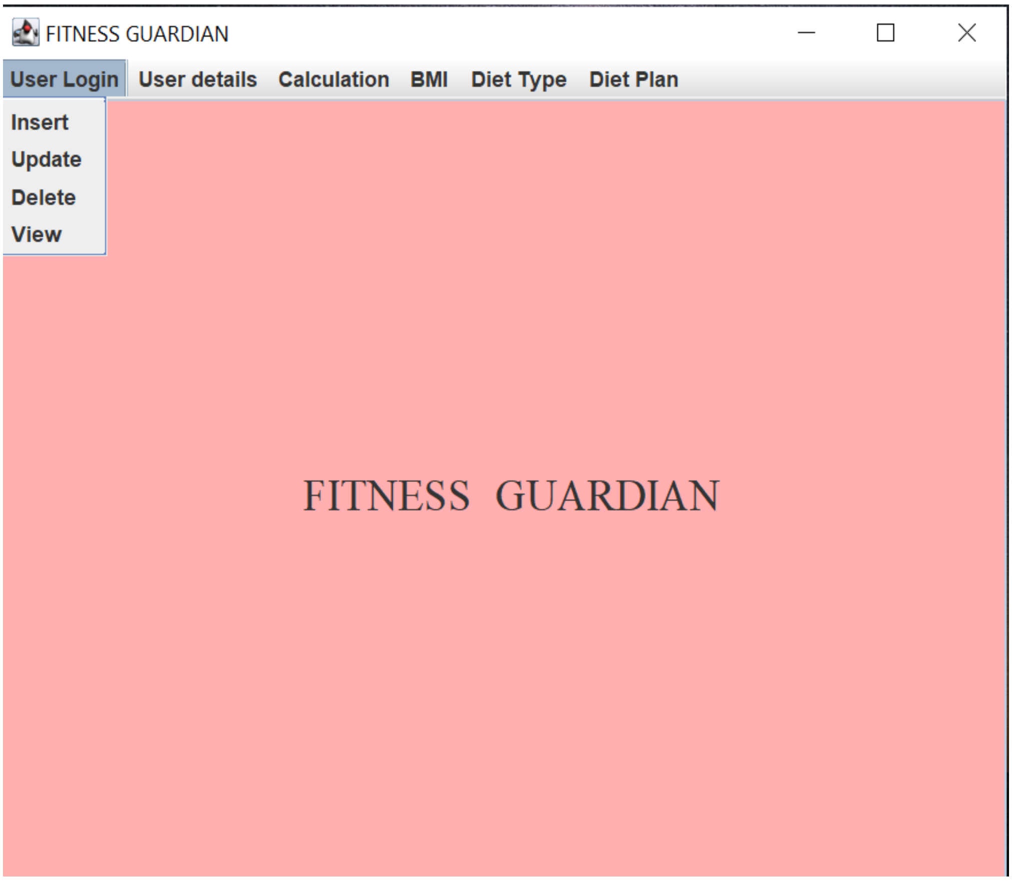
Name	Size	Packed	Type	Modified	CRC32
DietPlan....	3,876	1,911	CLASS File	18-06-20...	BD42...
GivePlan...	1,961	1,063	CLASS File	18-06-20...	8A5A...
GivePlan...	2,444	1,249	CLASS File	18-06-20...	0E03...
GivePlan...	2,435	1,299	CLASS File	18-06-20...	5933...
GivePlan...	2,058	1,103	CLASS File	18-06-20...	0D1D...
GivePlan...	3,630	1,867	CLASS File	18-06-20...	CE83...
GivePlan...	2,331	1,251	CLASS File	18-06-20...	DBD9...
GivePlan...	2,111	1,141	CLASS File	18-06-20...	DBEB...
GivePlan...	3,552	1,831	CLASS File	18-06-20...	E518...
GivePlan...	2,514	1,396	CLASS File	18-06-20...	0074...
GivePlan....	3,462	1,729	CLASS File	18-06-20...	C75B...
Main.class	436	306	CLASS File	18-06-20...	8B64...
OracleCo...	1,818	1,014	CLASS File	18-06-20...	22AC...
ProjectUI...	1,128	706	CLASS File	18-06-20...	0806...
ProjectUI....	5,077	2,363	CLASS File	18-06-20...	6330...

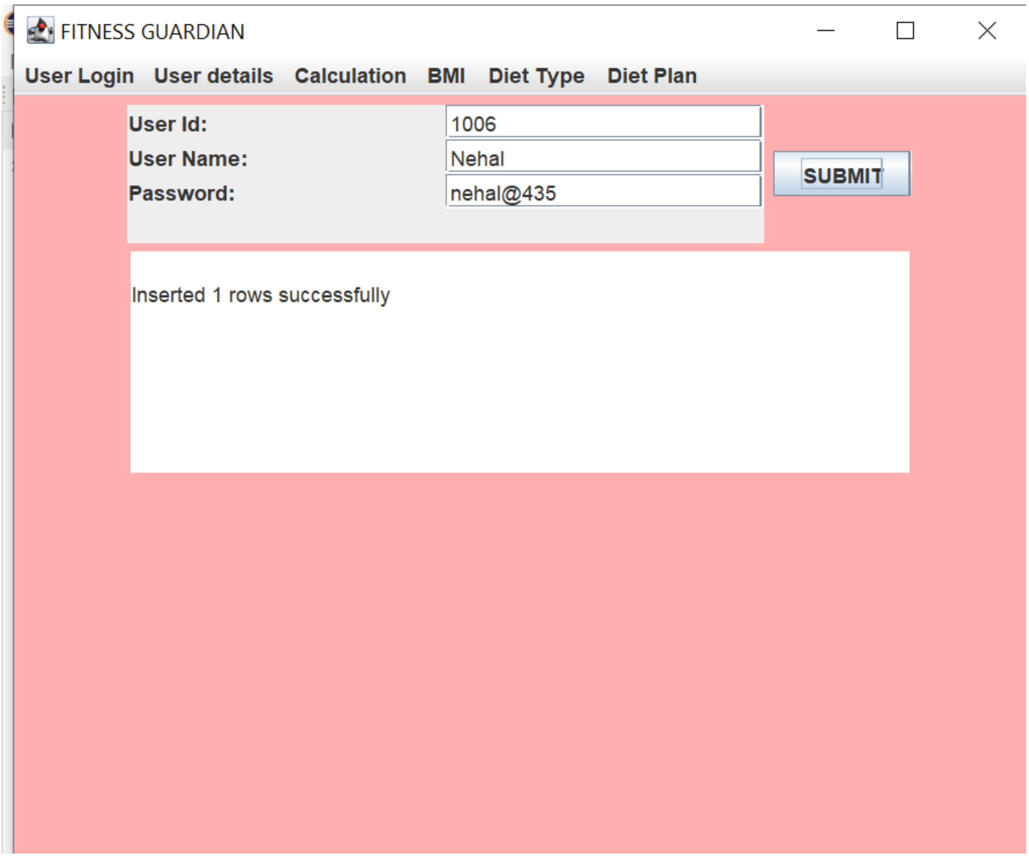
Total 64 files, 166,884 bytes

Testing:

Java GUI Testing:


FITNESS GUARDIAN
User Login
User details
Calculation
BMI
Diet Type
Diet Plan
FITNESS GUARDIAN


FITNESS GUARDIAN
User Login
User details
Calculation
BMI
Diet Type
Diet Plan
Insert
Update
Delete
View
FITNESS GUARDIAN

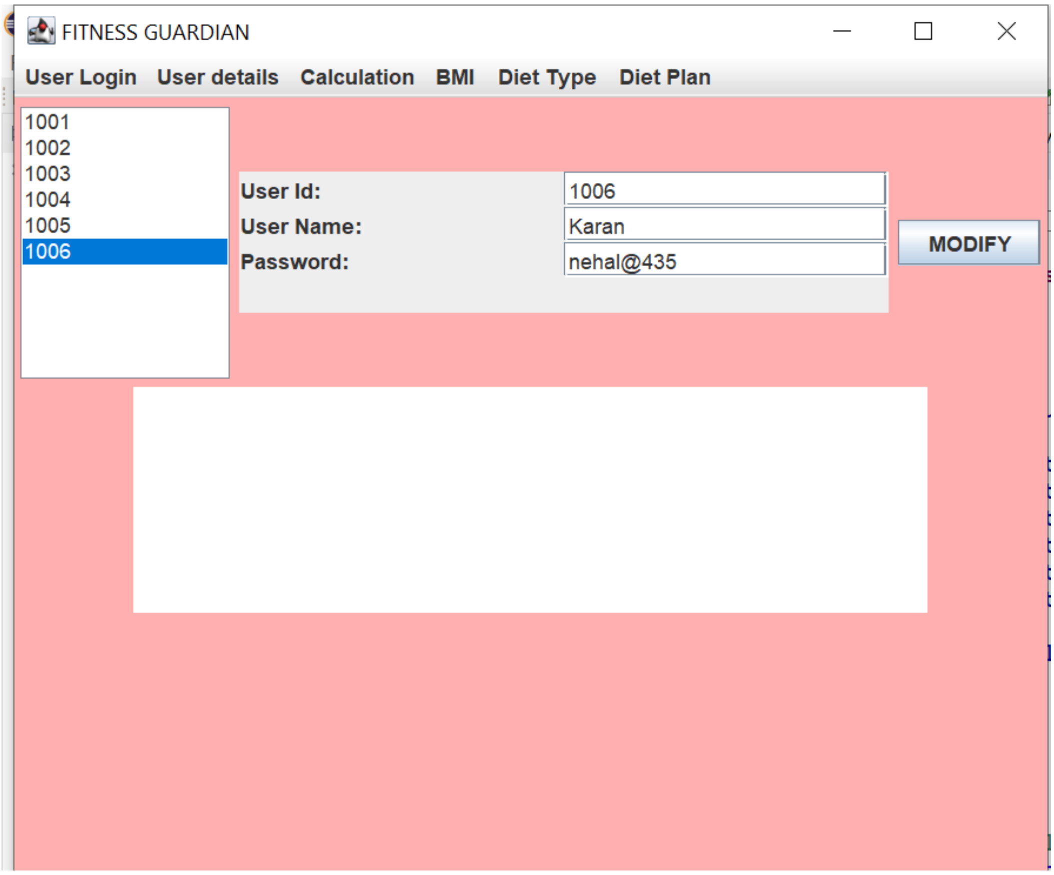


User Id: 1006

User Name: Nehal

Password: nehal@435

Inserted 1 rows successfully

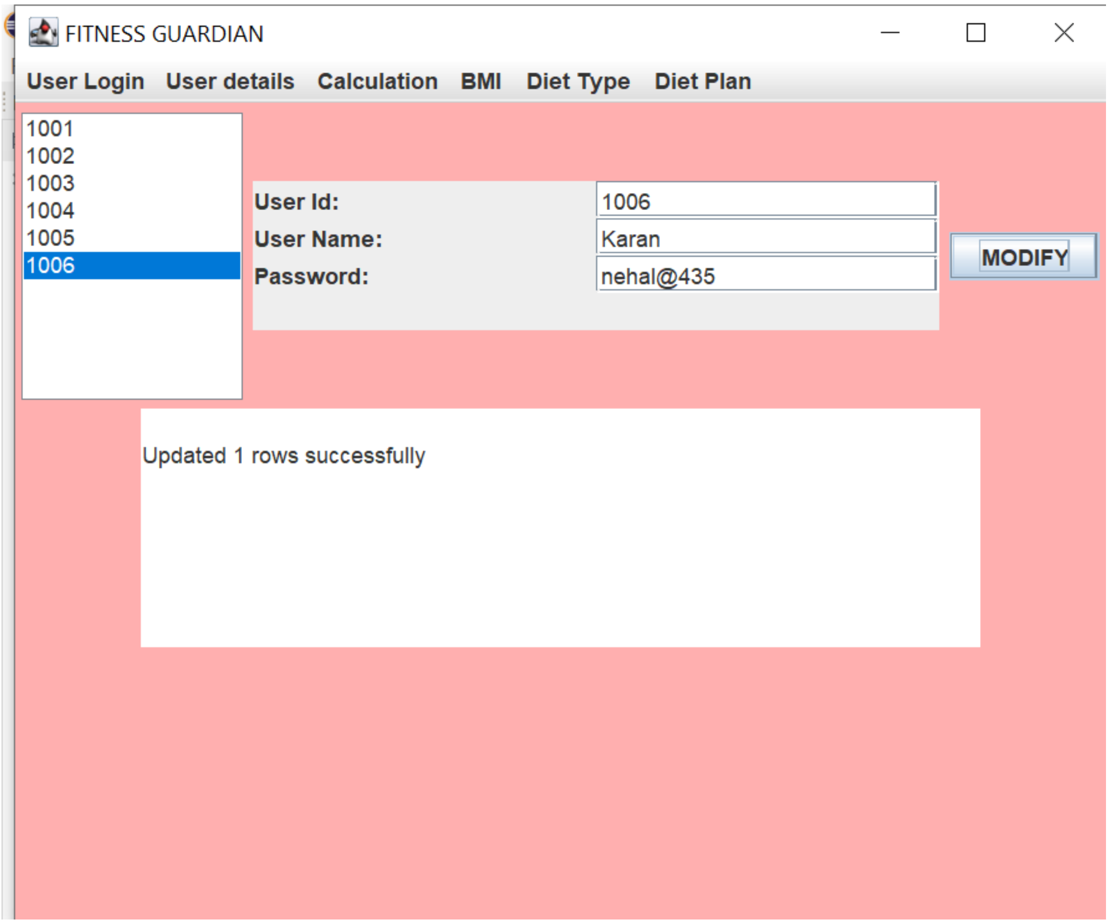


1001
1002
1003
1004
1005
1006

User Id: 1006

User Name: Karan

Password: nehal@435

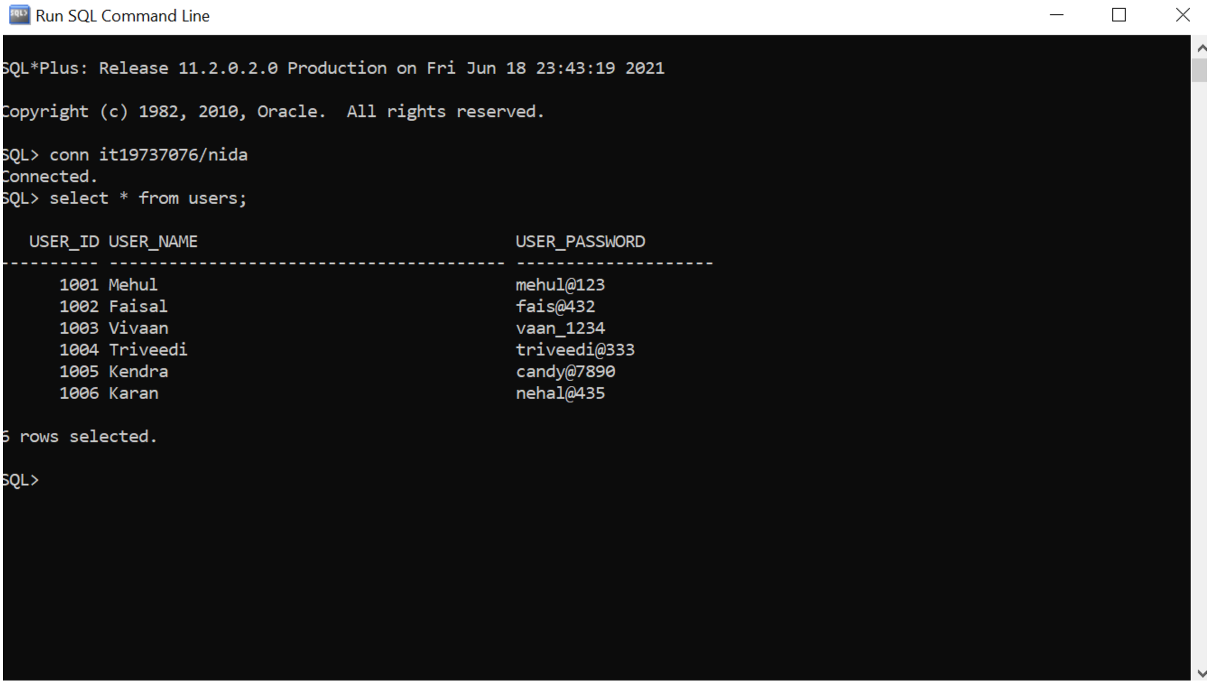


User Id: 1006

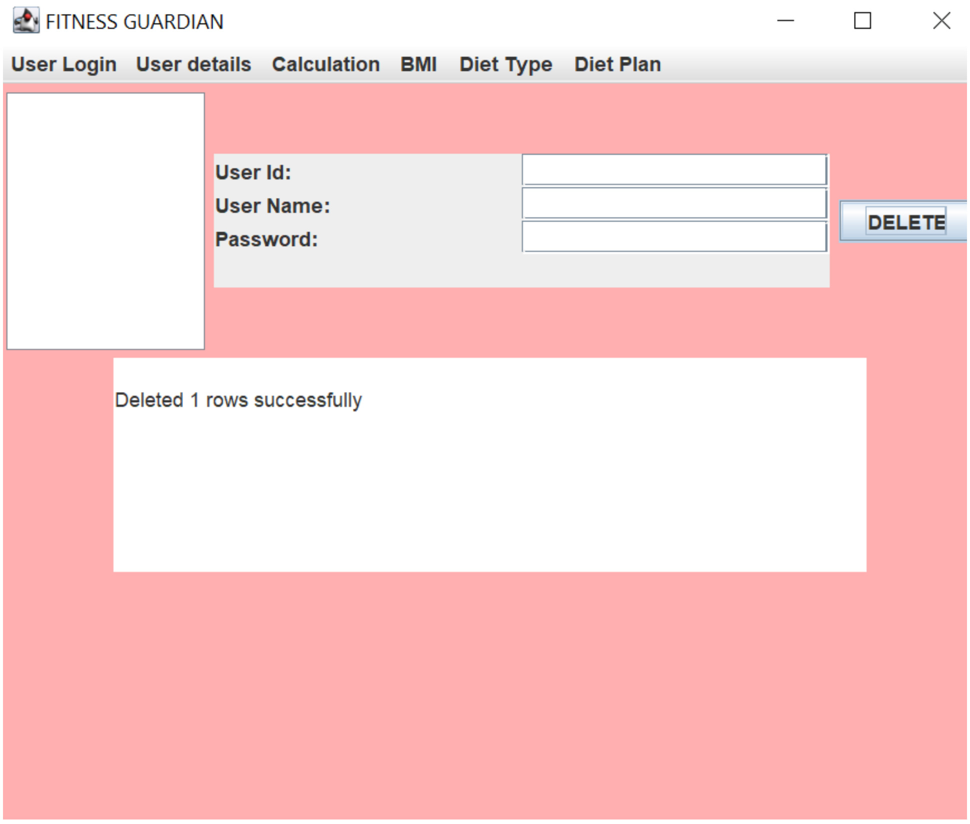
User Name: Karan

Password: nehal@435

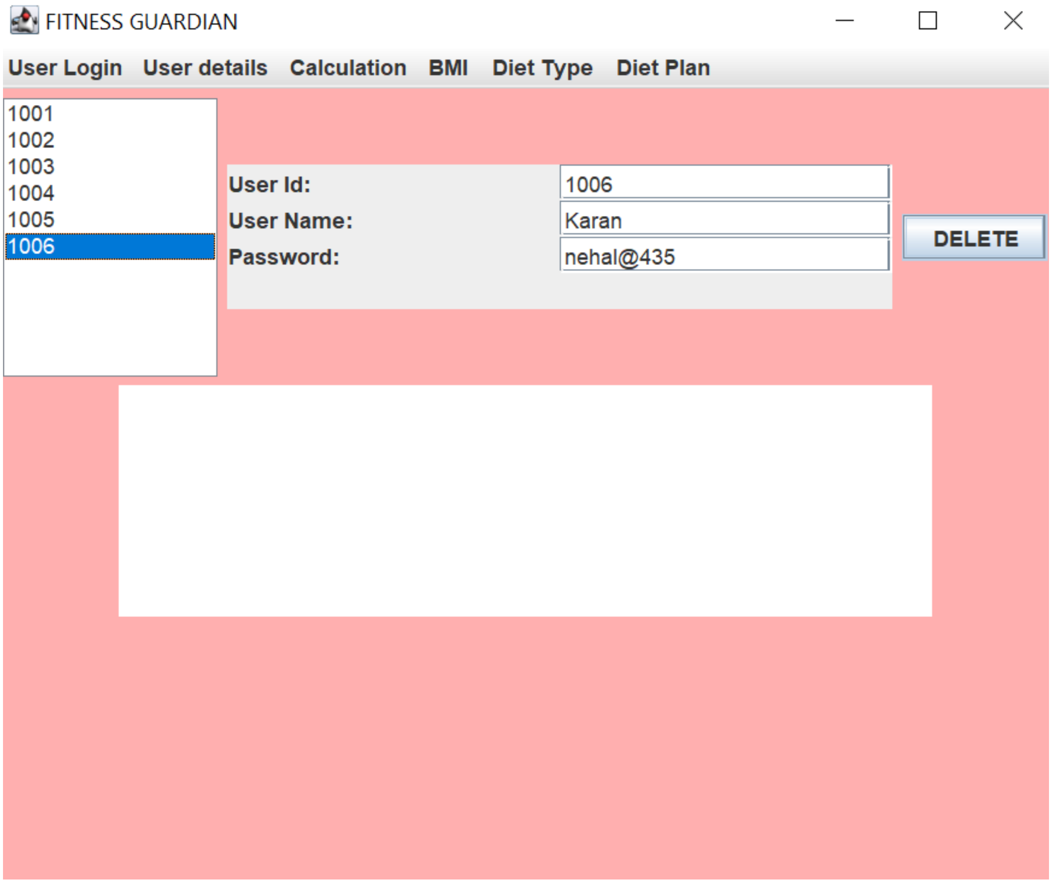
Updated 1 rows successfully



```
SQL*Plus: Release 11.2.0.2.0 Production on Fri Jun 18 23:43:19 2021

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> conn it19737076/nida
Connected.
SQL> select * from users;

   USER_ID USER_NAME                                USER_PASSWORD
---------- ---------------------------------------- --------------------
      1001 Mehul                                    mehul@123
      1002 Faisal                                   fais@432
      1003 Vivaan                                   vaan_1234
      1004 Triveedi                                 triveedi@333
      1005 Kendra                                   candy@7890
      1006 Karan                                    nehal@435

6 rows selected.

SQL>
```



Deleted 1 rows successfully

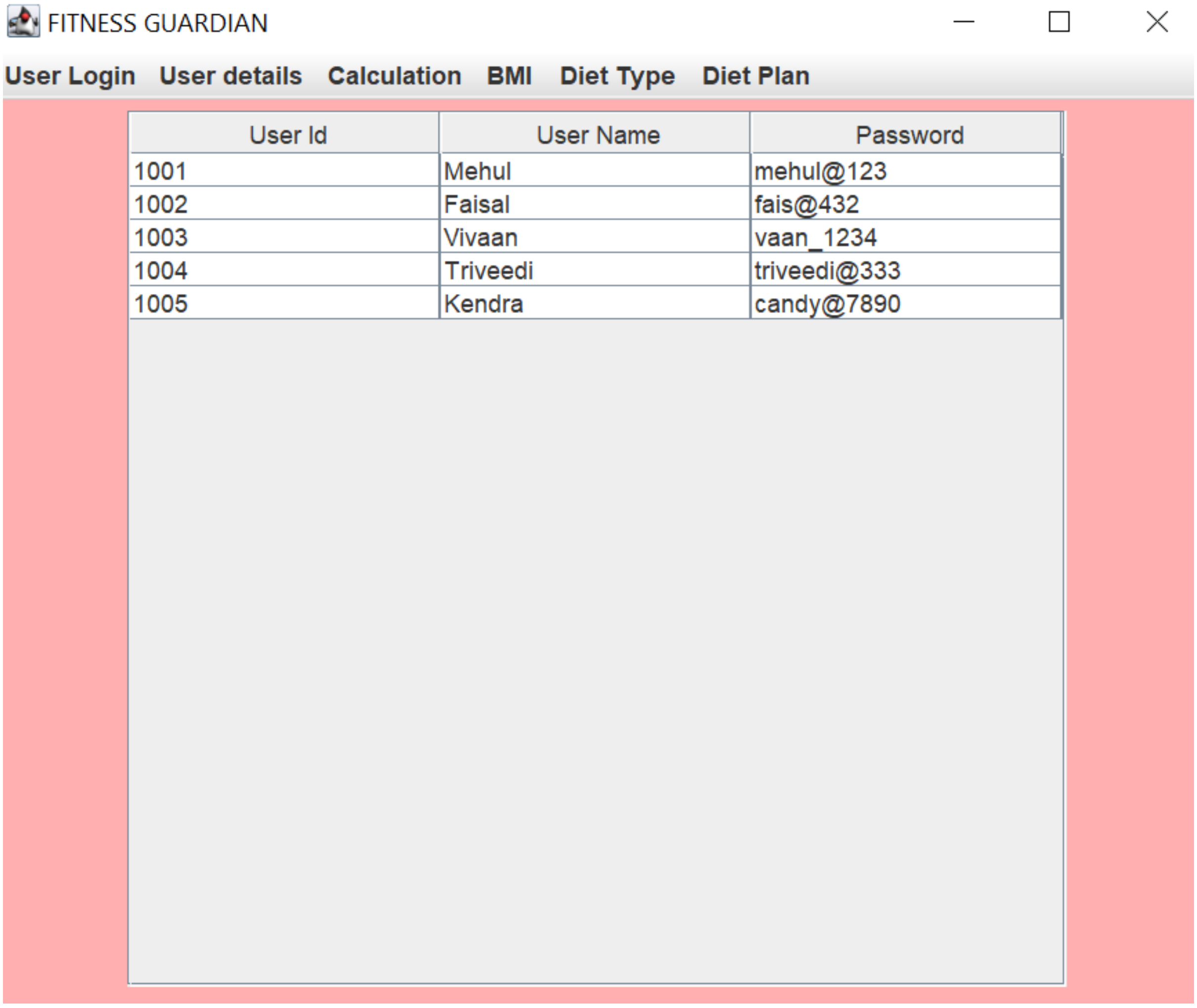


1001
1002
1003
1004
1005
1006

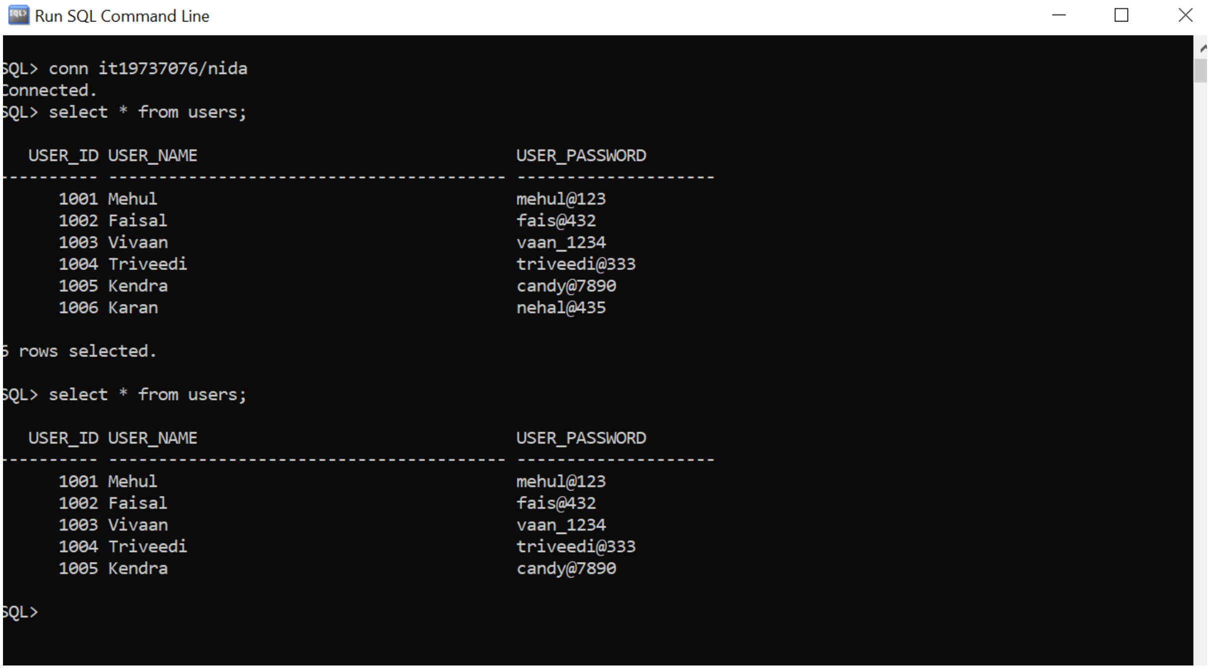
User Id: 1006

User Name: Karan

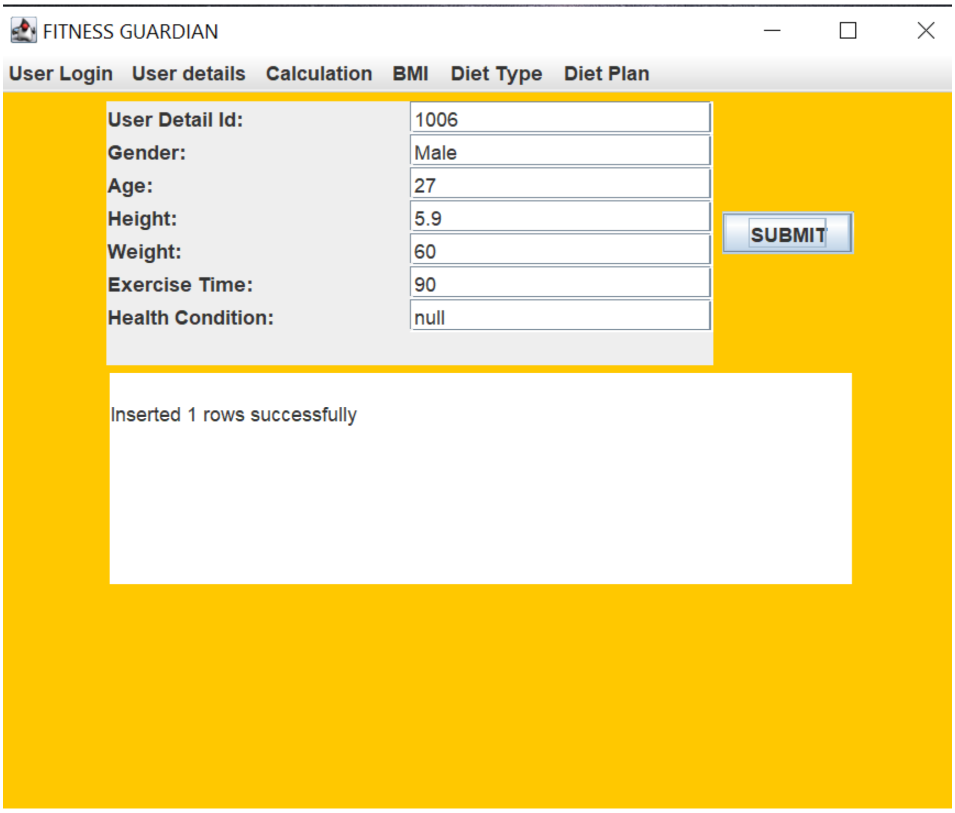
Password: nehal@435



User Id	User Name	Password
1001	Mehul	mehul@123
1002	Faisal	fais@432
1003	Vivaan	vaan_1234
1004	Triveedi	triveedi@333
1005	Kendra	candy@7890

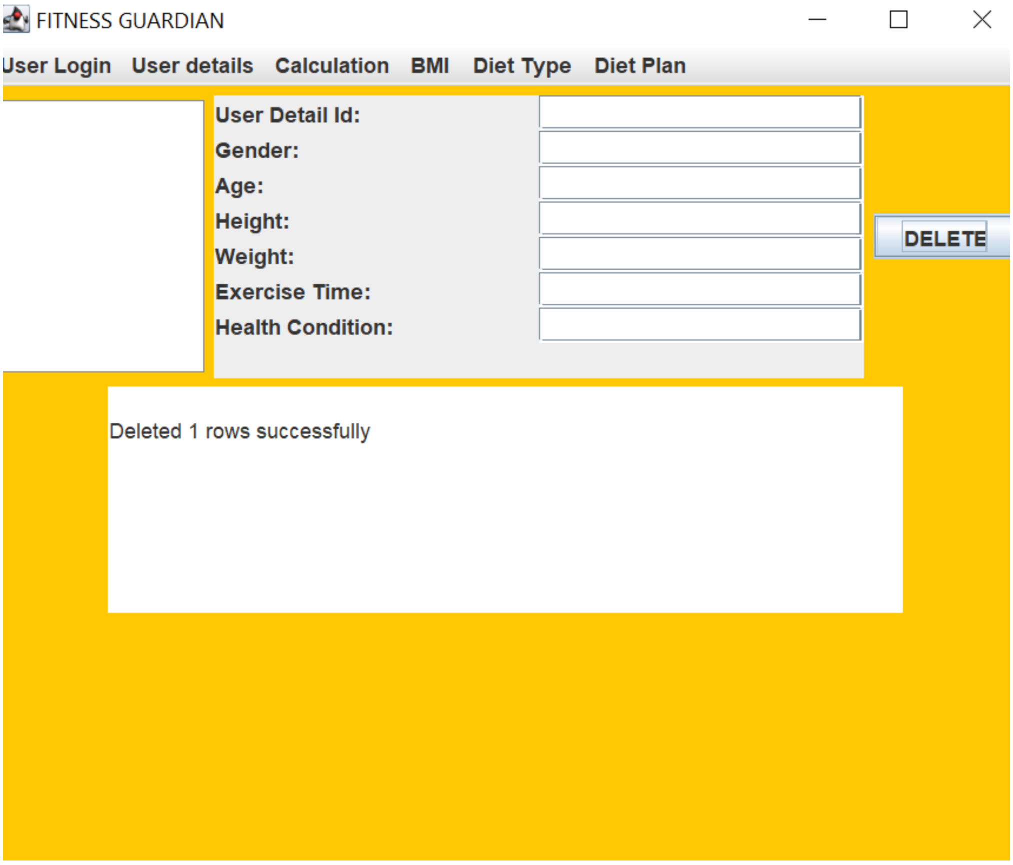


```
SQL> conn it19737076/nida
Connected.
SQL> select * from users;

   USER_ID USER_NAME                                USER_PASSWORD
---------- ---------------------------------------- --------------------
      1001 Mehul                                    mehul@123
      1002 Faisal                                   fais@432
      1003 Vivaan                                   vaan_1234
      1004 Triveedi                                 triveedi@333
      1005 Kendra                                   candy@7890
      1006 Karan                                    nehal@435

6 rows selected.

SQL> select * from users;

   USER_ID USER_NAME                                USER_PASSWORD
---------- ---------------------------------------- --------------------
      1001 Mehul                                    mehul@123
      1002 Faisal                                   fais@432
      1003 Vivaan                                   vaan_1234
      1004 Triveedi                                 triveedi@333
      1005 Kendra                                   candy@7890

SQL>
```



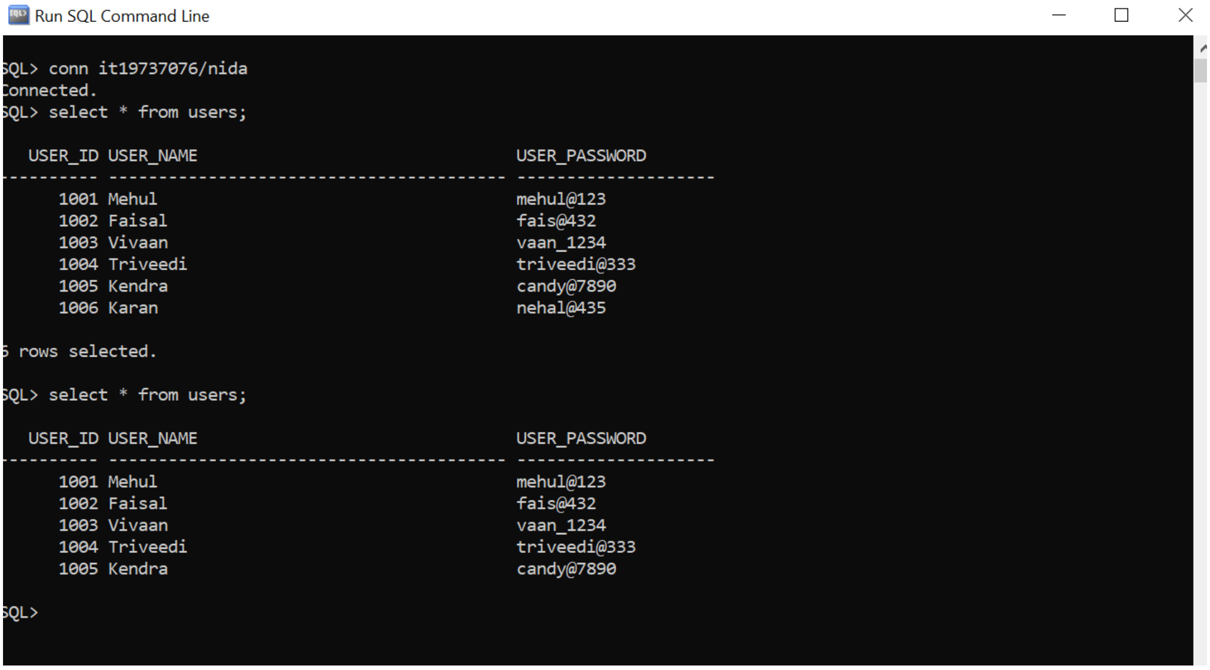
User Detail Id:	1006
Gender:	Male
Age:	27
Height:	5.9
Weight:	60
Exercise Time:	90
Health Condition:	null

```
Inserted 1 rows successfully
```

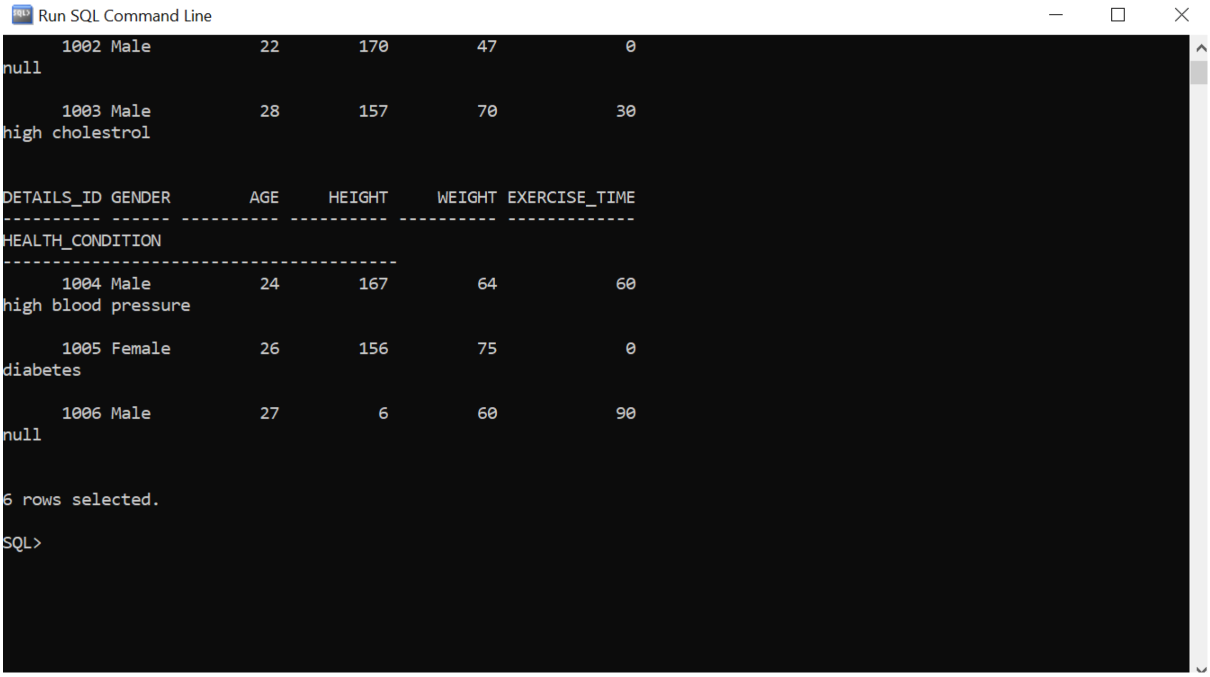


User Detail Id:	
Gender:	
Age:	
Height:	
Weight:	
Exercise Time:	
Health Condition:	

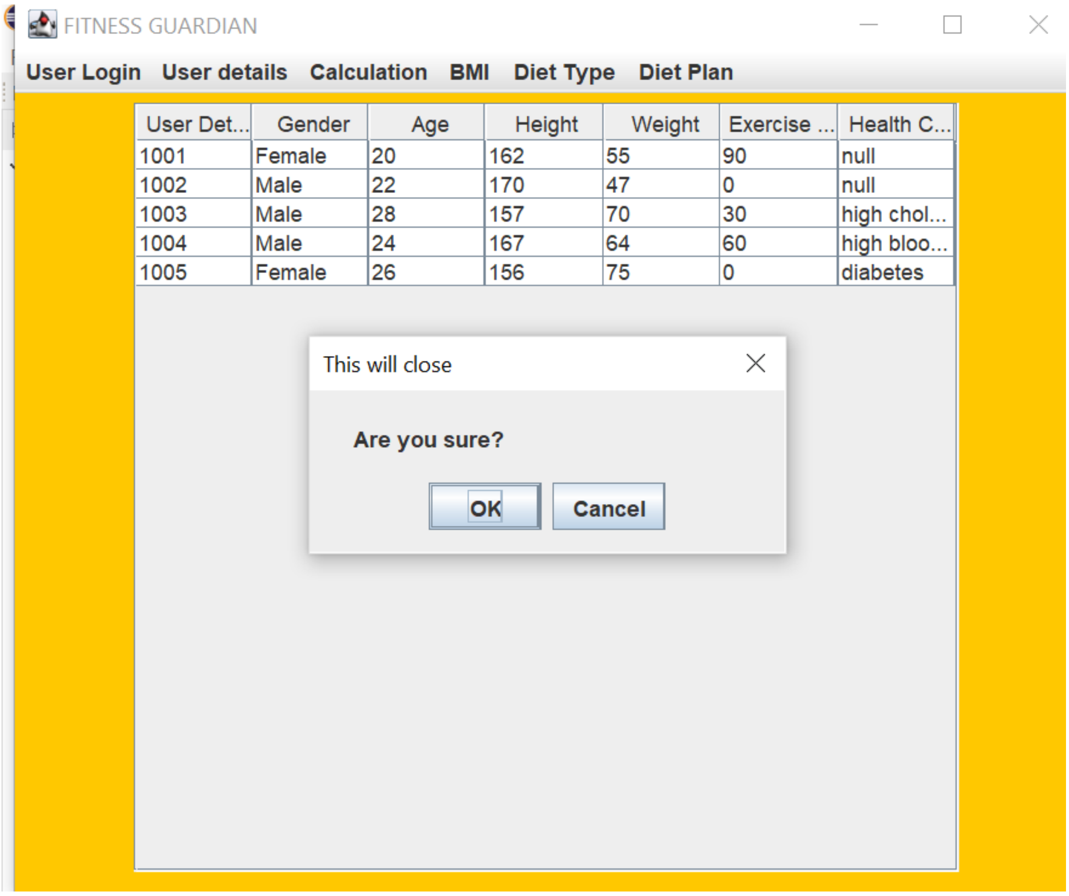
```
Deleted 1 rows successfully
```



```
SQL> conn it19737076/nida
Connected.
SQL> select * from users;

   USER_ID USER_NAME                                USER_PASSWORD
---------- ---------------------------------------- --------------------
      1001 Mehul                                    mehul@123
      1002 Faisal                                   fais@432
      1003 Vivaan                                   vaan_1234
      1004 Triveedi                                 triveedi@333
      1005 Kendra                                   candy@7890
      1006 Karan                                    nehal@435

6 rows selected.

SQL> select * from users;

   USER_ID USER_NAME                                USER_PASSWORD
---------- ---------------------------------------- --------------------
      1001 Mehul                                    mehul@123
      1002 Faisal                                   fais@432
      1003 Vivaan                                   vaan_1234
      1004 Triveedi                                 triveedi@333
      1005 Kendra                                   candy@7890

SQL>
```



```
      1002 Male           22        170        47            0
null

      1003 Male           28        157        70           30
high cholestrol


DETAILS_ID GENDER        AGE     HEIGHT     WEIGHT EXERCISE_TIME
---------- ------ ---------- ---------- ---------- -------------
HEALTH_CONDITION
----------------------------------------
      1004 Male           24        167        64           60
high blood pressure

      1005 Female         26        156        75            0
diabetes

      1006 Male           27          6        60           90
null


6 rows selected.

SQL>
```



User Det...	Gender	Age	Height	Weight	Exercise ...	Health C...
1001	Female	20	162	55	90	null
1002	Male	22	170	47	0	null
1003	Male	28	157	70	30	high chol...
1004	Male	24	167	64	60	high bloo...
1005	Female	26	156	75	0	diabetes

This will close

Are you sure?

Results:

I successfully completed this MINI PROJECT “Fitness Guardian”.

Discussion and Future work

While doing this project I got new ideas I understood how to work on projects. Now to further extend this project I want to create a android app by which I can control my project on my hand and connect to it.

References:

✦ https://www.academia.edu/36893248/Ramakrishnan_-_Database_Management_Systems_3rd_Edition

✦ https://docs.oracle.com/javase/7/docs/index.html

✦ https://www.javatpoint.com/dbms-tutorial

✦ http://www.sqlines.com/articles/java/sql_server_jdbc_connection