**ABSTRACT**

The “Center Management System” is an application which is made using JAVA. In this application the main objective is to manage the candidates taking admission in various courses as well as managing staff and faculties of a computer training center.

This application needs JRE (Java Runtime Environment) which is easily available and is free. Here the concept of Oracle is used which is used while registering a candidate and / or a staff or a faculty member. The problem description, methodology, advantages and drawbacks of the application are decrypted. Moreover it also contains sample user interfaces or output screens with test data, which are actually the screen images of the application when executed.

It contains the future prospects, limitation and areas on the parts of program on which there is a need to work upon to make this software more feasible, interactive, user friendly and efficient.

**LIST OF FIGURES**

Title PageNo**.**

* 1. Gateway of CMS : 31
  2. Admin Menu option 31
  3. CMS Admin Login 32
  4. Admin Change Password 32
  5. CMS Software Registration 33
  6. CMS Staff Registration 33
  7. CMS Faculty Registration 34
  8. CMS Student Registration 34
  9. Search Student 35

**INTRODUCTION**

**ABOUT THE APPLICATION**

The “Center Management System” is an application which is made using JAVA. In this application the main objective is to manage candidates applying for particular courses as well as faculties and other staffs are managed using JDK (Java Development Toolkit). Here first window is the “CMS Window” which is actually the login window for the administrator and is a link to new student Registration as well as new staff and faculty registration.fom here we can also search a particular student by giving his/her id or name.

This Application uses the concept of Oracle as backend to store details of student as well as staff and faculty members by creating suitable tables with suitable attribute. There is also an admin table which contains admin id and password. the administrator has the facility to change its password for extra security.

Here the concept of JDBC (Java Database Connectivity) is used with Oracle having data base name as XE. Various tables namely admin, staff, faculty, student etc have been made to store respective data in Oracle

**OVERVIEW**

In this application the first window is a menu driven window that the first menu is login. The second menu that is registration menu is initially inactive. The admin must login using admin id and password in order to make the registration menu active. The registration menu contains three options for registering staff, faculty, and students separately. There is another menu for searching a student either by its name or registration no.

**DESCRIPTION**

**ADVANTAGES**

This application is very user friendly and is keeps a track record of each and every candidate who has registered. Moreover this application is build using JAVA so runs on every stand alone computers.

**DRAWBACKS**

The drawback of this application is that it can only be used on standalone computer and is not server based. Thus the wideness of this application is restricted.

**Packages Used:-**

Packages imported in “Placement Registration System “ are as follows :-

* java.awt.\*
* java.io.\*
* java.applet.\*
* javax.swing.\*
* java.sql.\*
* java.awt.event.\*

**Classes Used:-**

* Checkbox
* CheckboxGroup
* String
* Integer
* Label
* TextField
* TextArea
* Button
* Double
* FileInputStream
* FileOutputStream

USER

STRUCTURE OF PROGRAMME

CMS

ADMIN LOGIN

VERIFY

No

Yes

HELP

REGISTRATION

SEARCH

STUDENT NAME

STUDENT ID

NO

VERIFYY

STUDENT

STAFF

FACULTY

NAME, ADDRESS,

GENDER,DEPARTMENT

NAME, ADDRESS,

GENDER,EDQ-QUALIFICATION

NAME, COURSE NAME,GENDER,FATHER NAME,MB NO

Yes

FOUND

END

**SOURCE CODE:-**

**Source Code of CMS.java**

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.sql.\*;

public class CMS extends Frame implements ActionListener

{

MenuBar mb;

Menu m1,m2,m3,m4;

MenuItem m11,m12,m21,m22,m23;

public static void main(String s[])

{

CMS cm=new CMS();

cm.job();

}

public CMS()

{

mb=new MenuBar();

m1=new Menu("Admin");

m2=new Menu("Registration");

m3=new Menu("Search");

m4=new Menu("Help");

m11=new MenuItem("Login");

m12=new MenuItem("Change Password");

m21=new MenuItem("Staff Registration");

m22=new MenuItem("Faculty Registration");

m23=new MenuItem("Student Registration");

}

public void job()

{

setSize(400,200);

setTitle("Centre Management Software");

setLayout(new FlowLayout());

mb.add(m1);

mb.add(m2);

mb.add(m3);

mb.add(m4);

m2.setEnabled(false);

m1.add(m11);

m1.add(m12);

m2.add(m21);

m2.add(m22);

m2.add(m23);

m11.addActionListener(this);

m12.addActionListener(this);

m21.addActionListener(this);

m22.addActionListener(this);

m23.addActionListener(this);

addWindowListener(new AA());

setMenuBar(mb);

setVisible(true);

}

class AA extends WindowAdapter

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==m11)

{

MyForm mf=new MyForm();

mf.job();

}

if(ae.getSource()==m12)

{

ChangePass cp=new ChangePass();

cp.job();

}

if(ae.getSource()==m21)

{

MyFollow m=new MyFollow();

m.job();

}

if(ae.getSource()==m22)

{

MyFollow1 m1=new MyFollow1();

m1.job();

}

if(ae.getSource()==m23)

{

MyFollow2 m2=new MyFollow2();

m2.job();

}

}

class MyForm extends Frame implements ActionListener

{

Label l1,l2;

TextField t1,t2;

Button b1;

String s1,s2,str;

public MyForm()

{

l1=new Label("EMAIL ID");

l2=new Label("PASSWORD");

t1=new TextField(20);

t2=new TextField(20);

b1=new Button("OK");

}

public void job()

{

setTitle("WEB ENTER");

setSize(300,200);

setLayout(null);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

});

l1.setBounds(40,40,80,30);

add(l1);

l2.setBounds(40,90,80,30);

add(l2);

t1.setBounds(135,40,100,20);

add(t1);

t2.setBounds(135,90,100,20);

add(t2);

t2.setEchoChar('\*');

b1.setBounds(100,140,100,20);

add(b1);

b1.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent ae)

{

try

{

if(ae.getSource()==b1)

{

s1=t1.getText();

s2=t2.getText();

str="select \* from admin";

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

Connection conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","scott","tiger");

Statement st=conn.createStatement();

ResultSet rs;

int flag=0;

rs=st.executeQuery(str);

while(rs.next())

{

flag=0;

String str1=rs.getString("id");

String str12=rs.getString("pass");

if(s1.equals(str1) && s2.equals(str12))

{

flag=1;

break;

}

}

if(flag==1)

{

System.out.println("Connection granted");

m2.setEnabled(true);

m11.setEnabled(false);

this.dispose();

}

else

{

System.out.println("Connection not granted");

this.dispose();

}

}

}

catch(Exception e)

{System.out.println(e);}

}

}

}

**Source Code of AdminLogin.java**

import java.awt.\*;

import java.awt.event.\*;

public class Password extends Frame implements WindowListener

{

Label l1,l2;

TextField t1,t2;

Button b1;

public static void main(String s[])

{

Password mf=new Password();

mf.job();

}

public Password()

{

l1=new Label("EMAIL ID");

l2=new Label("PASSWORD");

t1=new TextField(20);

t2=new TextField(20);

b1=new Button("OK");

}

public void job()

{

setTitle("WEB ENTER");

setSize(300,200);

setLayout(null);

addWindowListener(this);

l1.setBounds(40,40,80,30);

add(l1);

l2.setBounds(40,90,80,30);

add(l2)

t1.setBounds(135,40,100,20);

add(t2);

t2.setBounds(135,90,100,20);

b1.setBounds(100,240,100,20);

setVisible(true);

}

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

public void windowClosed(WindowEvent we)

{ }

public void windowOpened(WindowEvent we)

{ }

public void windowIconified(WindowEvent we)

{ }

public void windowDeiconified(WindowEvent we)

{ }

public void windowActivated(WindowEvent we)

{ }

public void windowDeactivated(WindowEvent we)

{ }

}

**Source Code of ChagePass.java**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

public class ChangePass extends Frame implements ActionListener

{

Label l1,l2,l3,l4;

TextField tf1,tf2,tf3,tf4;

Button b1;

String s1,s2,s3,s4,str;

public static void main(String s[])

{

ChangePass mf=new ChangePass();

mf.job();

}

public ChangePass()

{

l1=new Label("EMAIL ID:");

l2=new Label("OLD PASSWORD");

l3=new Label("NEW PASSWORD");

l4=new Label("CONFIRM PASSWORD");

tf1=new TextField(20);

tf2=new TextField(20);

tf3=new TextField (20);

tf4=new TextField (20);

b1=new Button("CHANGE");

}

public void job()

{

setTitle("ADMIN LOGIN");

setSize(300,300);

setLayout(null);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

});

l1.setBounds(40,40,75,25);

add(l1);

l2.setBounds(40,90,75,25);

add(l2);

l3.setBounds(40,140,75,25);

add(l3);

l4.setBounds(40,190,75,25);

add(l4);

tf1.setBounds(135,40,100,25);

add(tf1);

tf2.setBounds(135,90,100,25);

add(tf2);

tf2.setEchoChar('\*');

tf3.setBounds(135,140,100,25);

add(tf3);

tf3.setEchoChar('\*');

tf4.setBounds(135,190,100,25);

add(tf4);

tf4.setEchoChar('\*');

b1.setBounds(100,240,100,25);

add(b1);

b1.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent ae)

{

try

{

if(ae.getSource()==b1)

{

s1=tf1.getText();

s2=tf2.getText();

s3=tf3.getText();

s4=tf4.getText();

str="select \* from admin";

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

Connection conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","scott","tiger");

Statement st=conn.createStatement();

ResultSet rs;

int flag=0;

rs=st.executeQuery(str);

while(rs.next())

{

flag=0;

String str1=rs.getString("id");

String str12=rs.getString("pass");

if(s1.equals(str1) && s2.equals(str12))

{

flag=1;

break;

}

}

if(flag==1 && s3.equals(s4))

{

str="update admin set pass='"+s3+"' where id='"+s1+"'";

st.executeUpdate(str);

this.dispose();

}

else

{

System.out.println("Connection not granted");

this.dispose();

}

}

}

catch(Exception e)

{System.out.println(e);}

}

**Source Code of StudentRegistration.java**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

public class MyFollow extends Frame implements ActionListener,WindowListener

{

Label l1,l2,l3,l4;

TextField t1,t2;

Checkbox c1,c2;

CheckboxGroup cbg;

Choice c;

Button b1;

String s1,s2,s3,s4,str;

public static void main(String s[])

{

MyFollow mf=new MyFollow();

mf.job();

}

public MyFollow()

{

l1=new Label("Name");

l2=new Label("Address");

l3=new Label("Gender");

l4=new Label("Department");

t1=new TextField(20);

t2=new TextField(30);

cbg=new CheckboxGroup();

c1=new Checkbox("Male",true,cbg);

c2=new Checkbox("Female",false,cbg);

c=new Choice();

c.add("cst");

c.add("Mechanical");

c.add("Civil");

b1=new Button("OK");

}

public void job()

{

setTitle("WEB Browsing");

setSize(400,400);

setLayout(null);

addWindowListener(this);

l1.setBounds(40,40,80,30);

add(l1);

l2.setBounds(40,90,80,30);

add(l2);

l3.setBounds(40,140,80,30);

add(l3);

l4.setBounds(40,190,80,30);

add(l4);

t1.setBounds(135,40,100,30);

add(t1);

t2.setBounds(135,90,100,30);

add(t2);

c1.setBounds(135,140,80,30);

add(c1);

c2.setBounds(230,140,80,30);

add(c2);

c.setBounds(135,190,80,30);

add(c);

b1.setBounds(100,240,100,30);

add(b1);

b1.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent ae)

{

try

{

s1=t1.getText();

s2=t2.getText();

if(c1.getState())

s3=c1.getLabel();

else

if(c2.getState())

s3=c2.getLabel();

s4=c.getSelectedItem();

str="Insert into staff values('"+s1+"','"+s2+"','"+s3+"','"+s4+"')";

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

Connection conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","scott","tiger");

Statement st=conn.createStatement();

if(ae.getSource()==b1)

{

st.executeUpdate(str);

t1.setText("");

t2.setText("");

}

}

catch(Exception e)

{System.out.println(e);}

}

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

public void windowClosed(WindowEvent we)

{ }

public void windowOpened(WindowEvent we)

{ }

public void windowIconified(WindowEvent we)

{ }

public void windowDeiconified(WindowEvent we)

{ }

public void windowActivated(WindowEvent we)

{ }

public void windowDeactivated(WindowEvent we)

{ }

}

**Source Code of FacultyRegistration.java**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

public class MyFollow1 extends Frame implements ActionListener,WindowListener

{

Label l1,l2,l3,l4;

TextField t1,t2;

Checkbox c1,c2;

CheckboxGroup cbg;

Choice c;

Button b1;

String s1,s2,s3,s4,str;

public static void main(String s[])

{

MyFollow1 mf1=new MyFollow1();

mf1.job();

}

public MyFollow1()

{

l1=new Label("Name");

l2=new Label("Address");

l3=new Label("Gender");

l4=new Label("Educational Qualification");

t1=new TextField(20);

t2=new TextField(30);

cbg=new CheckboxGroup();

c1=new Checkbox("Male",true,cbg);

c2=new Checkbox("Female",false,cbg);

c=new Choice();

c.add("Graduate");

c.add("Post Graduate");

c.add("PHD");

b1=new Button("OK");

}

public void job()

{

setTitle("Google Chrome");

setSize(400,400);

setLayout(null);

addWindowListener(this);

l1.setBounds(40,40,80,30);

add(l1);

l2.setBounds(40,90,80,30);

add(l2);

l3.setBounds(40,140,80,30);

add(l3);

l4.setBounds(40,190,80,30);

add(l4);

t1.setBounds(135,40,100,30);

add(t1);

t2.setBounds(135,90,100,30);

add(t2);

c1.setBounds(135,140,80,30);

add(c1);

c2.setBounds(230,140,80,30);

add(c2);

c.setBounds(135,190,80,30);

add(c);

b1.setBounds(100,240,100,30);

add(b1);

b1.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent ae)

{

try

{

s1=t1.getText();

s2=t2.getText();

if(c1.getState())

s3=c1.getLabel();

else

if(c2.getState())

s3=c2.getLabel();

s4=c.getSelectedItem();

str="Insert into faculty values('"+s1+"','"+s2+"','"+s3+"','"+s4+"')";

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

Connection conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","scott","tiger");

Statement st=conn.createStatement();

if(ae.getSource()==b1)

{

st.executeUpdate(str);

t1.setText("");

t2.setText("");

}

}

catch(Exception e)

{System.out.println(e);}

}

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

public void windowClosed(WindowEvent we)

{ }

public void windowOpened(WindowEvent we)

{ }

public void windowIconified(WindowEvent we)

{ }

public void windowDeiconified(WindowEvent we)

{ }

public void windowActivated(WindowEvent we)

{ }

public void windowDeactivated(WindowEvent we)

{ }

}

**Source Code of StaffRegistration .java**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

public class MyFollow2 extends Frame implements ActionListener,WindowListener

{

Label l1,l2,l3,l4,l5;

TextField t1,t2,t3;

Checkbox c1,c2;

CheckboxGroup cbg;

Choice c;

Button b1;

String s1,s2,s3,s4,s5,str;

public static void main(String s[])

{

MyFollow2 mf2=new MyFollow2();

mf2.job();

}

public MyFollow2()

{

l1=new Label("Student Name");

l2=new Label("Course Name");

l3=new Label("Gender");

l4=new Label("Father's Name");

l5=new Label("Mobile No");

t1=new TextField(30);

t2=new TextField(30);

t3=new TextField(30);

cbg=new CheckboxGroup();

c1=new Checkbox("Male",true,cbg);

c2=new Checkbox("Female",false,cbg);

c=new Choice();

c.add("BCA");

c.add("MCA");

c.add("BBA");

c.add("MBA");

b1=new Button("Submit");

}

public void job()

{

setTitle("WEB Browsing");

setSize(400,400);

setLayout(null);

addWindowListener(this);

l1.setBounds(40,40,80,30);

add(l1);

l2.setBounds(40,90,80,30);

add(l2);

l3.setBounds(40,140,80,30);

add(l3);

l4.setBounds(40,190,80,30);

add(l4);

l5.setBounds(40,240,80,30);

add(l5);

t1.setBounds(135,40,100,30);

add(t1);

t2.setBounds(135,190,100,30);

add(t2);

t3.setBounds(135,240,100,30);

add(t3);

c1.setBounds(135,140,80,30);

add(c1);

c2.setBounds(230,140,80,30);

add(c2);

c.setBounds(135,90,80,30);

add(c);

b1.setBounds(100,290,100,30);

add(b1);

b1.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent ae)

{

try

{

s1=t1.getText();

s2=t2.getText();

s3=t3.getText();

if(c1.getState())

s4=c1.getLabel();

else

if(c2.getState())

s4=c2.getLabel();

s5=c.getSelectedItem();

str="Insert into student values('"+s1+"','"+s2+"','"+s3+"','"+s4+"','"+s5+"')";

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

Connection conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","scott","tiger");

Statement st=conn.createStatement();

if(ae.getSource()==b1)

{

st.executeUpdate(str);

t1.setText("");

t2.setText("");

t3.setText("");

}

}

catch(Exception e)

{System.out.println(e);}

}

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

public void windowClosed(WindowEvent we)

{ }

public void windowOpened(WindowEvent we)

{ }

public void windowIconified(WindowEvent we)

{ }

public void windowDeiconified(WindowEvent we)

{ }

public void windowActivated(WindowEvent we)

{ }

public void windowDeactivated(WindowEvent we)

{ }

}

**Source Code of SearchStudent .java**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import javax.swing.\*;

public class Search extends Frame implements ActionListener

{

Checkbox rd1,rd2;

CheckboxGroup cbg;

Label l1,l2,l3;

TextField t1,t2;

TextArea ta;

Button jb,jb1;

JFrame fm=new JFrame();

String sd,s1,s2,s3,s4,s5,s6,item="rd1",st;

Connection con;

ResultSet rs;

int j=0;

Statement stmt;

Search()

{

setSize(600,400);

setTitle("Search Student");

l1=new Label("STUDENT NAME");

l1.setBounds(5,40,200,20);

t1=new TextField(20);

t1.setBounds(220,40,200,20);

t1.setEnabled(false);

l2=new Label("STUD ID");

l2.setBounds(5,80,200,20);

t2=new TextField(20);

t2.setBounds(220,80,200,20);

t2.setEnabled(false);

jb=new Button("SEARCH");

//jb.setMnemonic('S');

jb.setActionCommand("ser");

jb.addActionListener(this);

jb.setBounds(210,120,200,25);

jb1=new Button("RESET");

//jb1.setMnemonic('R');

jb1.setActionCommand("r");

jb1.addActionListener(this);

jb1.setBounds(5,120,200,25);

l3=new Label("Result:");

l3.setBounds(5,160,75,25);

ta=new TextArea(10,20);

ta.setBounds(100,160,400,200);

cbg=new CheckboxGroup();

rd1=new Checkbox("Student Name",false,cbg);

//rd1.setActionCommand("rd1");

rd1.addItemListener(new RB());

rd1.setBounds(180,0,100,15);

rd2=new Checkbox("Student ID",false,cbg);

//rd2.setActionCommand("rd2");

rd2.addItemListener(new RD());

rd2.setBounds(180,15,200,15);

Panel jp=new Panel();

jp.setLayout(null);

jp.add(l1);

jp.add(t1);

jp.add(l2);

jp.add(t2);

jp.add(jb1);

jp.add(jb);

jp.add(rd1);

jp.add(rd2);

jp.add(l3);

jp.add(ta);

add(jp);

addWindowListener(new WIND());

setVisible(true);

}

class RB implements ItemListener

{

public void itemStateChanged(ItemEvent i1)

{

if(cbg.getSelectedCheckbox()==rd1)

{

item="rd1";

t1.setEnabled(true);

t2.setEnabled(false);

t1.setText("");

}

}

}

class RD implements ItemListener

{

public void itemStateChanged(ItemEvent i1)

{

if(cbg.getSelectedCheckbox()==rd2)

{

item="rd2";

t2.setEnabled(true);

t1.setEnabled(false);

t2.setText("");

}

}

}

/\*===========================SEARCH BUTTON=============================\*/

public void actionPerformed(ActionEvent t)

{

String sd=(String)t.getActionCommand();

s1=t1.getText();

//s1=s1.toLowerCase();

s2=t2.getText();

//s2=s2.toLowerCase();

if(sd.equals("ser"))

{

try

{

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

}

catch(Exception ee)

{

}

try

{

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","scott","tiger");

con.setAutoCommit(false);

}

catch(Exception sqe)

{

}

if((item.equals("rd1") && (s1==null||s1.equals(""))) || (item.equals("rd2") && (s2==null||s2.equals(""))))

{

try

{

con.rollback();

con.close();

JOptionPane.showMessageDialog(fm,"INPUT FIELD WITHOUT DATA !");

}

catch(Exception g)

{

}

}

else

{

if(item.equals("rd1"))

{

//s4=s2;

try

{

ta.setText("");

stmt=con.createStatement();

rs=stmt.executeQuery("select Name,Course from student where Name='"+s1+"'");

while(rs.next())

{

s3=rs.getString("Name");

s4=rs.getString("Course");

s5=s3+" "+s4;

if(s1.equals(s3))

{

ta.append(s5+"\n");

j++;

}

}

rs.close();

stmt.close();

con.close();

}

catch(Exception m)

{

}

}

if(item.equals("rd2"))

{

//s4=s2;

try

{

ta.setText("");

stmt=con.createStatement();

rs=stmt.executeQuery("select Name,Course,MobileNo from student where MobileNo='"+s2+"'");

while(rs.next())

{

s3=rs.getString("Name");

s4=rs.getString("Course");

s6=rs.getString("MobileNo");

s5=s3+" "+s4;

if(s2.equals(s6))

{

ta.append(s5+"\n");

j++;

}

}

rs.close();

stmt.close();

con.close();

}

catch(Exception m)

{

}

}

if((s3==null)||(s4==null)||(s3.equals(""))||(s4.equals("")))

{

JOptionPane.showMessageDialog(fm," Student NOT FOUND !");

j=0;

s3=null;

s4=null;

}

if(s3!=null)

{

JOptionPane.showMessageDialog(fm," Student FOUND ! \nNumber of Students found = "+j);

j=0;

s3=null;

s4=null;

}

}

}

/\*===========================RESET BUTTON=============================\*/

if(sd.equals("r"))

{

rd1.setState(true);

rd2.setState(false);

t1.setText("");

t1.setEnabled(true);

t2.setText("");

t2.setEnabled(false);

ta.setText("");

}

}

/\*=========================WINDOW CLOSING EVENT=========================\*/

private class WIND extends WindowAdapter

{

public void windowClosing(WindowEvent we)

{

dispose();

}

}

public static void main(String s[])

{

Search sc=new Search();

}

}

**OUTPUT SCREENS AND DESCRIPTION WITH TEST DATA**

****

Figure 1: Gateway of Centre Management Software

This is the gateway of center management software. it is menu driven. The registration menu is initially inactive. Once the administrator enters administrative id and password then only the registration menu is activated. The admin menu contains two options (a) login (b) change password. This is shown in figures 2 below.



Figure 2: Admin Menu option

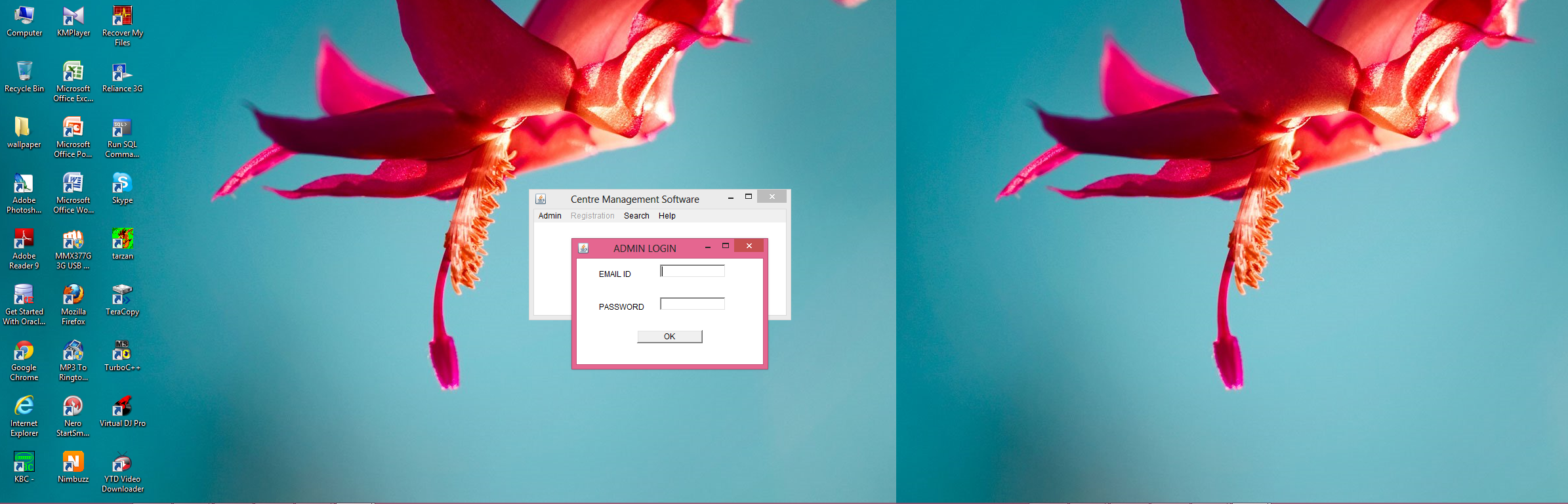


Figure 3 : CMS Admin Login

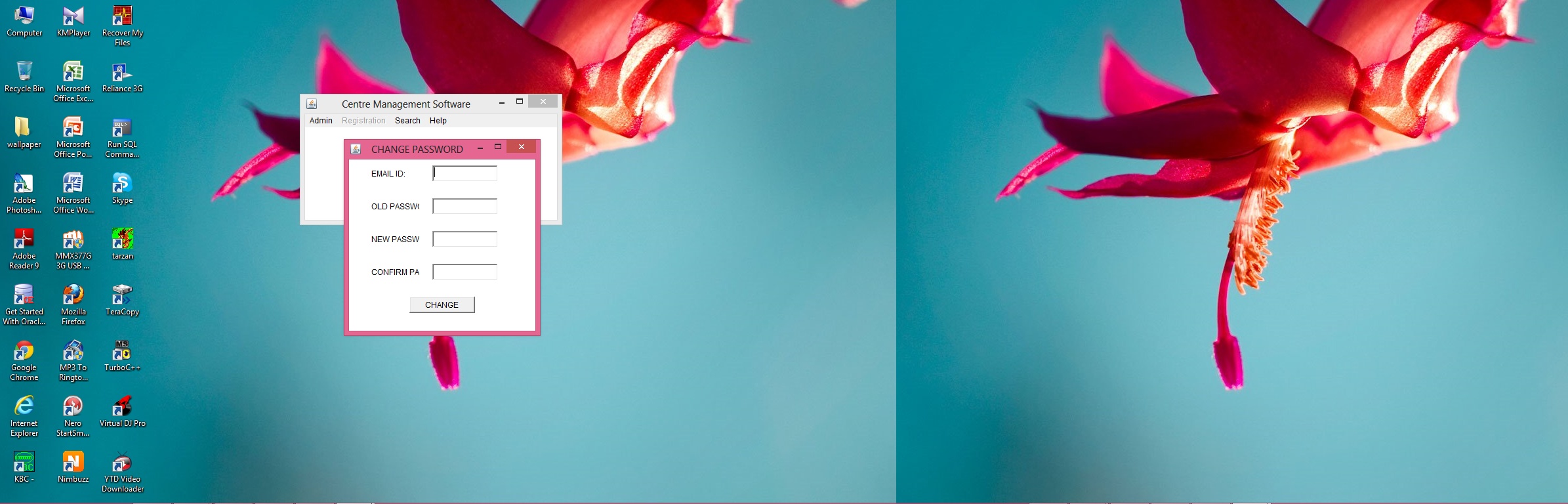


Figure 4: Administrator Change Password.

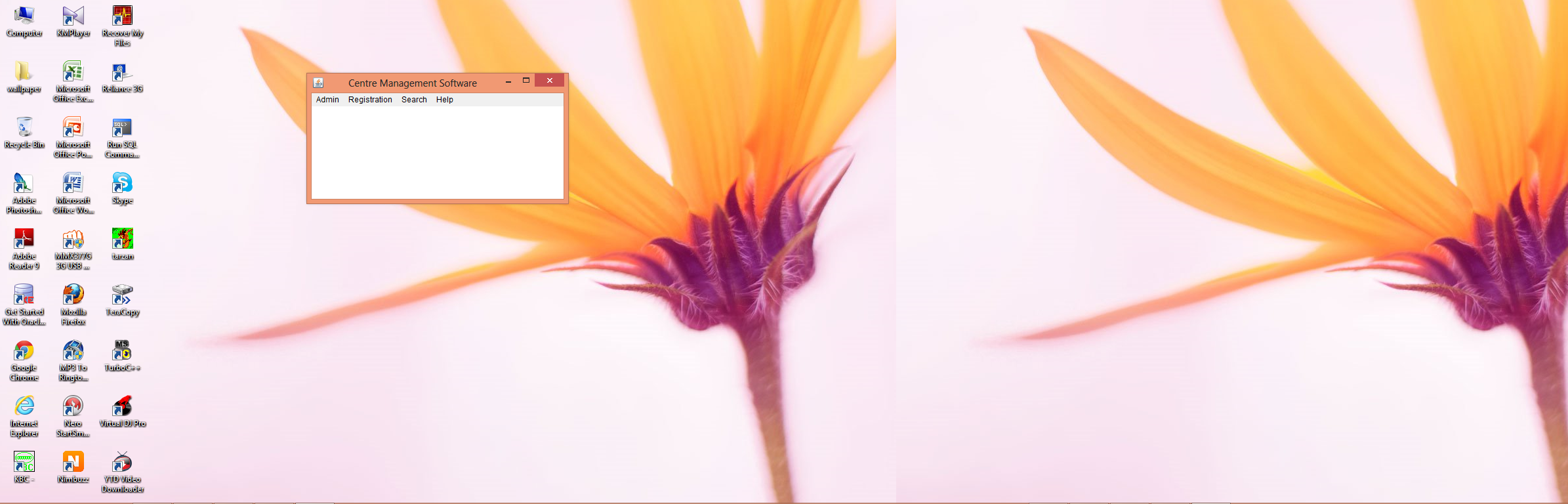


Figure 5: Center Management Software Registration.

This menu contains options for new staff, Faculty, Student registration.

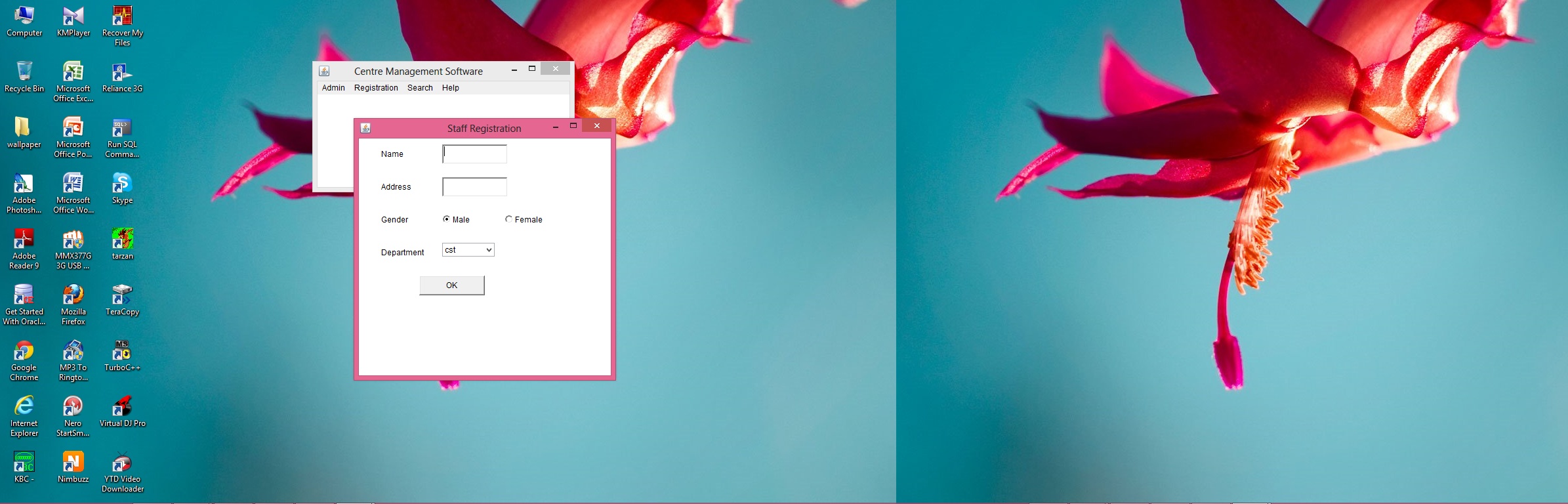


Figure 6: CMS Staff Registration

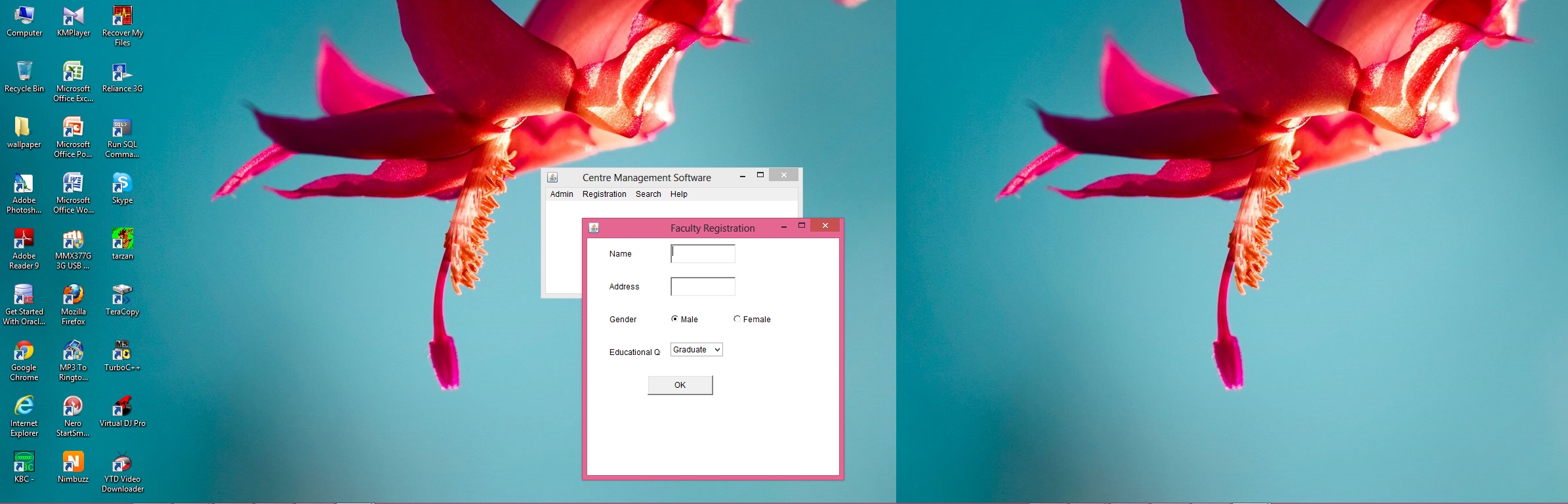


Figure 7: CMS Faculty Registration.

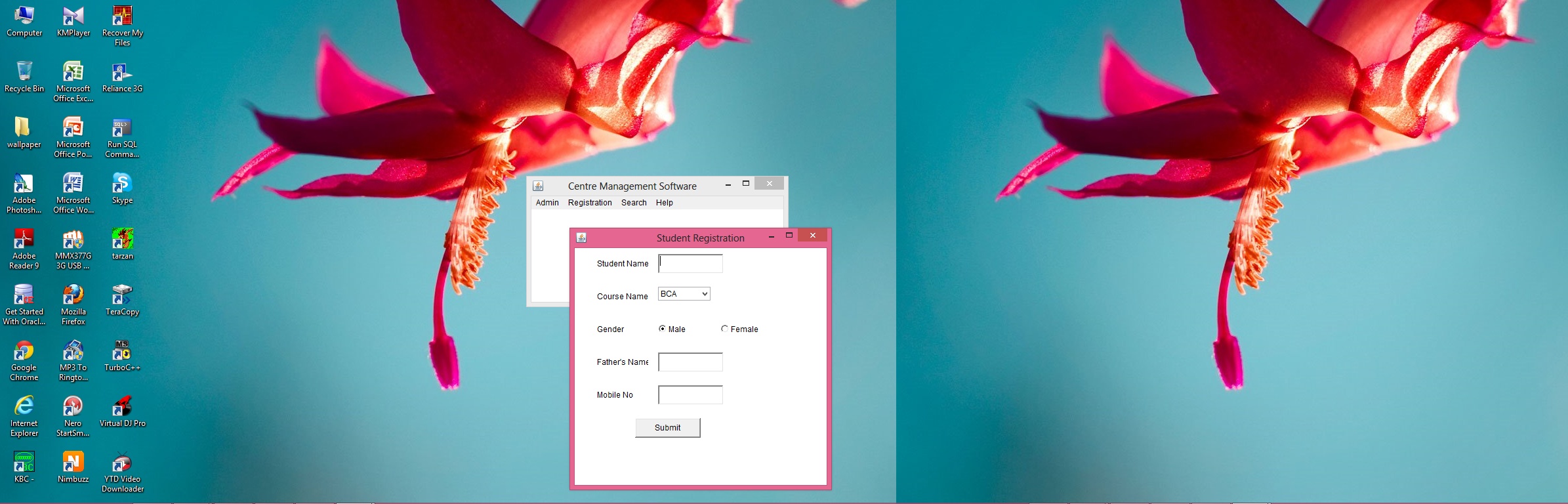


Figure 8: CMS Student Registration.

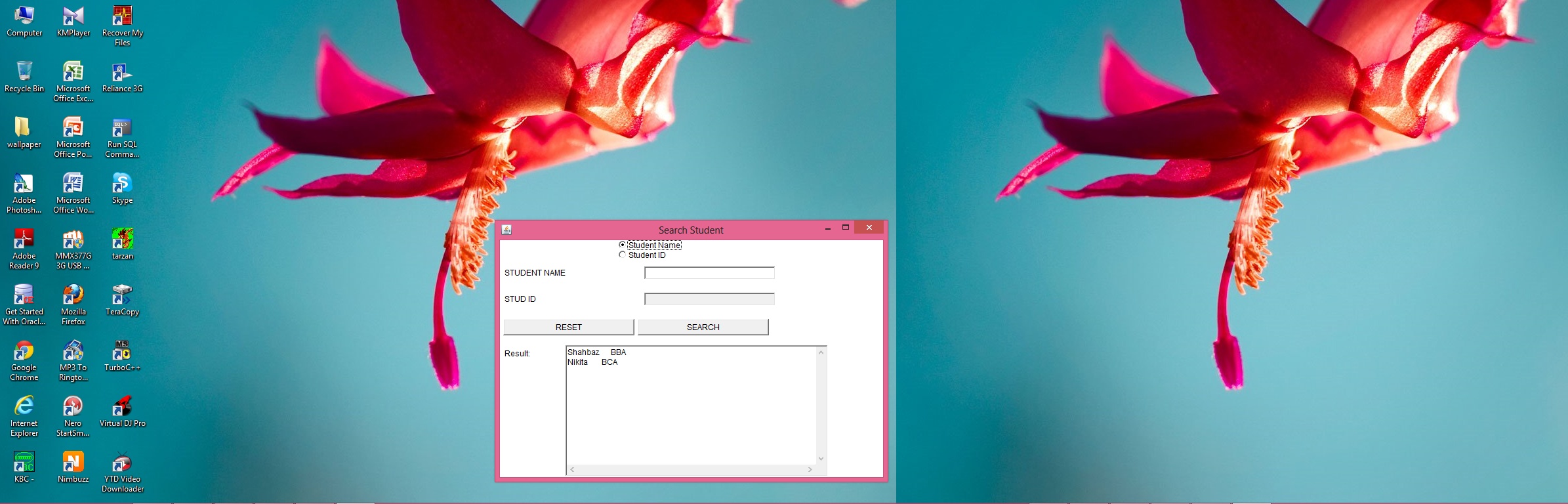


Figure 9 : Student Search

**FUTURE PROSPECTS AND WORKS**

* The application of this Software is that, it helps in maintaining and keeping records in very organized way.
* Very user friendly so can be used in Database related fields of center management software.
* The total control of the application is in the hands of the administrator, therefore the data once input and is finalized will be changed with the help of the administrator only. Thus regular alteration of the data is restricted
* Scope Of Improvement:-
  + This application is not server based so works on standalone computers.
  + The scope of the details of the candidate entered is in fewer amounts, so ways must be adopted to enhance the detailing of a candidate.
  + There is an option to create the software for web and mobile application.

**BIBLIOGRAPHY**

* Schildt,Herbert“*TheComplete.Reference.7th.Edition”,* Delhi-110006 : The McGraw-Hill,2007.
* Thomas M,Todd “*Java Data Access - JDBC, JNDI, and JAXP*”,M & T Books An Imprints of Hungry Minds,Inc. 2003.