

**PROJECT REPORT ON**  
**PINNACLE DATABASE MANAGEMENT SYSTEM**  
**BY**

**MR. GAURAV HARWANI**

**MR. DHIRAJ TAWANI**

**MR. ASHISH INGALE**

**MR. SHUBHAM THORAT**

**GUIDED BY:-**

**PROF.MS. BHARTI KHEMANI**

**Department of Computer Technology**

**S.H.M.I.T**

**Ulhasnagar – 421003**

**Academic Year**

**2015-16**

# **CERTIFICATE**

# **INSTITUTE OF TECHNOLOGY**

## **ULHASNAGAR - 421003**

CERTIFIED that this Project Report entitled" PINNACLE DATABASE MANAGEMENT" submitted by

Mr. Gaurav Harwani Roll no 153116, a student of third year of the course COMPUTER TECHNOLOGY as a partial fulfillment of the Diploma Course Engineering/Technology as prescribed by MAHARASHTRA State Board of Technical Education(MSBTE), Mumbai, for the academic year 2015-2016.

And, that, I have instructed guided him for said work from time to time and found him to be satisfactorily progressive.

And, that, following student students are associated with him for this project work, however his contribution was proportionate.

1. Mr. Gaurav Harwani
2. Mr. Dhiraj Tawani
3. Mr. Ashish Ingale
4. Mr. Shubham Thorat

And, that, the said work has been accessed by me and I am satisfied that the same is up to the standard envisaged for the level of the course.

And, that, the said work may be presented to External Examiner.

DATE:

(Signature of guide)

(Internal Examiner)

(External Examiner)

(Head of Department)

(Principal)

# Submission

I, Gaurav Harwani Roll No.153116, a student of Third Year of the course Computer Technology humbly submit that I have completed from time to time Project Work as described in this Report by my own skills and that, following students(s) were associated with me for this work, however quantum of contribution has been approved by the teacher.

1. Mr. Gaurav Harwani
2. Mr. Dhiraj Tawani
3. Mr. Ashish Ingale
4. Mr. Shubham Thorat

And, that I have not copied the report or its any appreciate part from any other literature in contravention of the academic ethics.

Date:

(Signature of Student)

# INDEX

## Contents

ACKNOWLEDGEMENT.....	06
PREFACE.....	08
PROJECT DESCRIPTION.....	10
REQUIREMENT.....	12
SOWFTARE MAP.....	14
OUTPUT SCREEN.....	16
SOURCE CODE.....	16
ADVANTAGES.....	60
FUTURE SCOPE.....	62
CONCLUSION.....	64
REFERENCE.....	66

# **ACKNOWLEDGEMENT**

---

I would like to express my gratitude to all those who gave me the possibility to undertake our Project for final year i.e. "Pinnacle database management".

We would like to take this opportunity to express our deep sense of gratitude towards all of them who spans their time, energy and intelligence towards the beautification of our project. This justification will never sound good if we don't express our vote of thanks to **Prof. Bharti Khemani, Prof. Shilpa Ingoley & Prof. Sneha Lachhani** who provided us innumerable guidance and even extended their support in execution and preparation of our project. They motivated us and gave their suggestions to add new features in our software.

There are times in projects when your clock beats your time again and again and you run of the time and dangling to complete the project. Our H.O.D. **Mr D. V. Chawak** and Principal **Mr A. B. Patil** made us endure such times with their unfailing support and help.

Also we would like to add a few heart felt words to thank our librarian **Mr. Shardaprasad** for letting us borrow some books which helped us in our software. We would like to thank our parents for their financial and never ending support, for the help in our studies and for its success. And it would not be successful without God who guides us in everyday life and activities; we thank Him for all the good health he has given to us, and for success of our study. For all the people who helped us a lot, thank you very much and may god bless you all.

BY:

GAURAV HARWANI	16
ASHISH INGALE	19
DHIRAJ TAWANI	52
SHUBHAM THORAT	54

# **PREFACE**



19<sup>th</sup> Century is known for the Industrial Revolution, whereas the current era is well known among all as IT era i.e. Information Technology era. The Computers are responsible for major change in our lives.

Just one click and information is with you this is current situation. Computers play vital role in this process. Internet well known as source of knowledge, cloud of the information, has made life so easy that if someone would have said few years back that the life after some years going to be so easy, then people would have said him mad, but Yes! It's the only fact that Internet, and this computer Era has made huge difference in human life and made it so easy, better, fast.

We the engineering students are totally surrounded by a number of software which are quite user-friendly and easy to use. The very thought of how much hard work to design the software made us to take up this project. The extent to which the software is easy to use, to the same extent they require hard work to design. This project aims to bridge the gap between theory and application.

The aim of our project is to help people leave the manual approach and move towards the modern approach of maintaining student records of their classes. Not only a classes but also colleges and schools can also use the modern approach of maintaining the student details. which would be useful even after many years. Our project report has the ability to satisfy the needs of classes who is going to use the software to maintain the records of student details.

Our project has been accomplished in Java language. Our project report gives the detailed idea about a software covering most of the details that can be useful for a classes. We have put our all the efforts to take it as simple as anyone can get at the first glance. As any person, if once he has a look at all the INS & OUTS of the project will get all the topics very easily and quickly. For better understanding our project report in a better way regardless of whether he is technical or nontechnical person.

We have covered an in – depth description of every module in our project. With the valuable guidance given by Mrs. BhartiKhemani, Mr. D. V. Chawak, and our Principal Mr. A. B. Patil sir, we decided the project topic as “Pinnacle database management”, which is accomplished using Java and MS-Access.

# 1.PROJECT DESCRIPTION

Pinnacle Database Management System (PDMS) is a modern day software to control the activities of a classroom. Ever dreamed of a software to track your classroom activity in real time? Ever thought of how useful it would be to know what your employees or students do when they are supposed to work or study? Ever lost money and/or confidential information, lacking control over your staff irresponsibility? PDMS is the solution you needed. This powerful software allows you to track your work, giving you the most detailed information on what, how and when your staff performed. Whether it is a library public network, university or commercial organization network – PDMS offers efficient control. The system consists of server and client parts. PDMS server can be installed on any computer in the PDMS grid. It can be installed even remotely from the PC with PDMS on it or via Active Directory Group Policy in a Windows domain. You may tune this classroom monitoring software so that it will record activity on all grid computers. This information can be later used for the deeper analysis and advanced report generation. This powerful software from us saves your money.

At first in PINNACLE GROUP TUITIONS, all the storage of data was done through paperwork. From calculating a student's fees to attendance, from messaging to storing information was done through sheets of modularized paper where it would be stored in cupboards and storage rooms.

Doing paperwork was very hectic and cause a lot of mess often when one needed to find some information. Whenever a search of student information or student databases was done, the mess in the office would go from top to down crossing ever limit. At first with less no of students it seemed pretty easy but with growing popularity and increasing number of students this became a trouble on the staff's head. Natural calamities when struct PINNACLE caused the data to vanish causing a havoc in the collection of fees and giving salaries to staff.

# **REQUIREMENTS**

## **Software Required**

- ❖ Notepad
- ❖ Microsoft ACCESS

## **Hardware Required**

- ❖ 32 Bit processor
- ❖ 1 GB RAM
- ❖ 10 GB Hard Disk

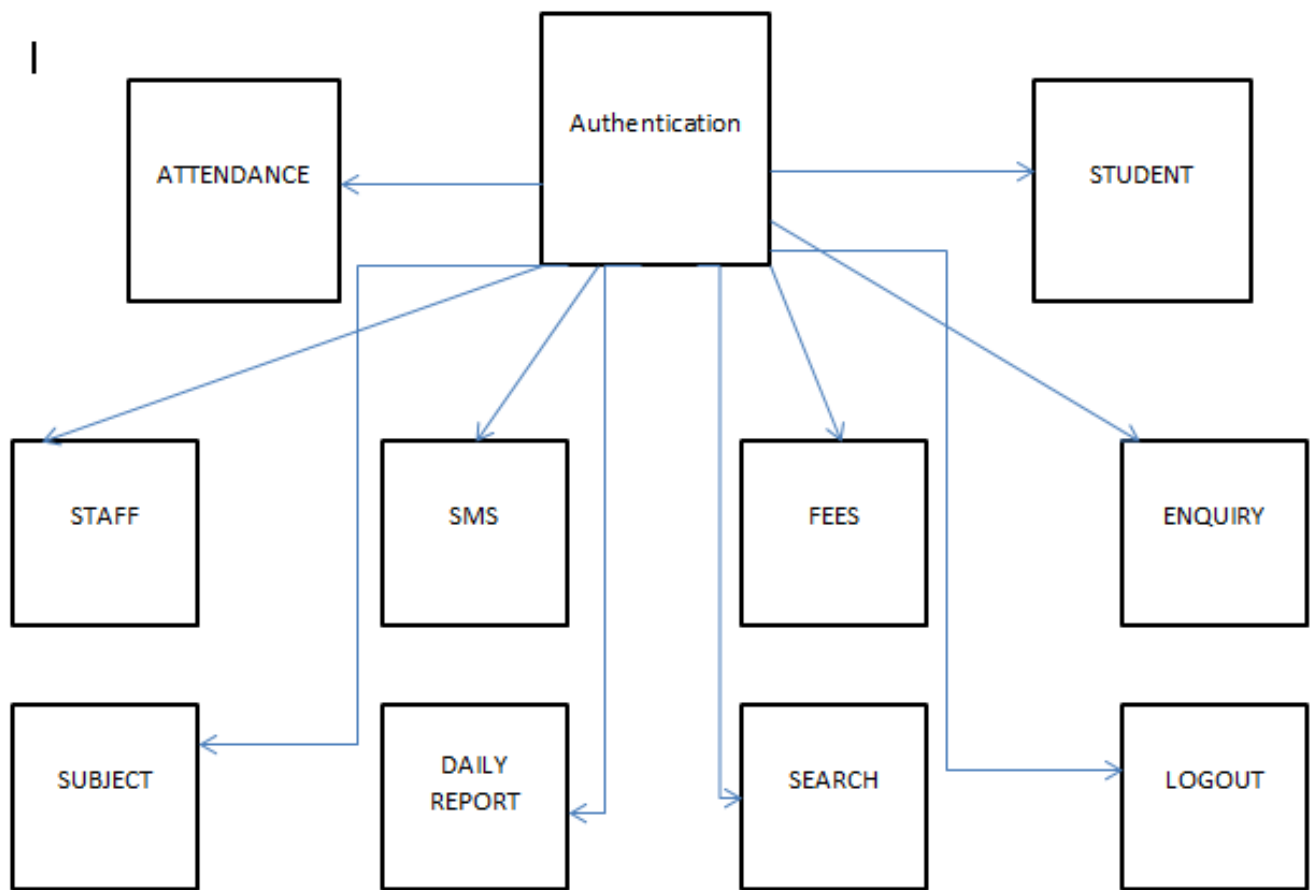
## **Languages Used**

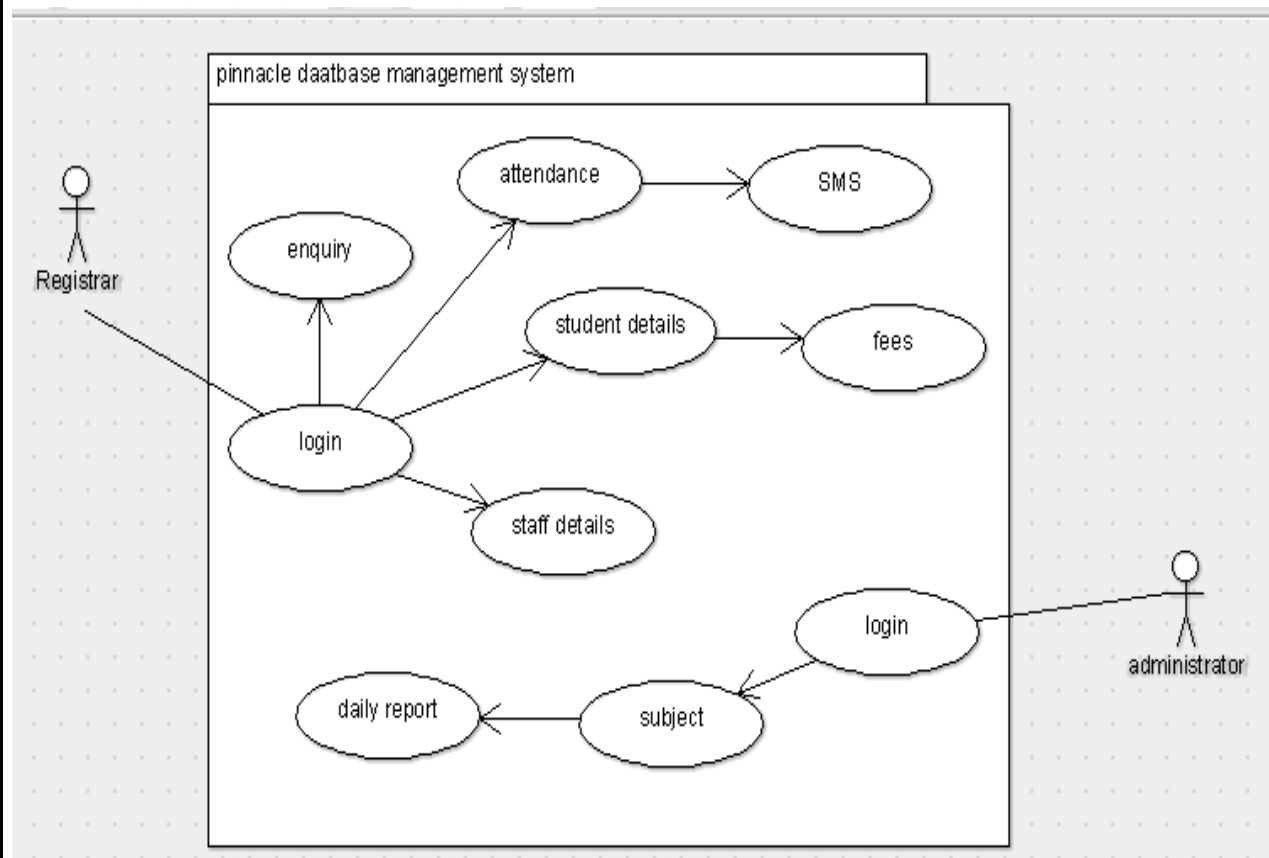
- ❖ JAVA
- ❖ SQL

## **SYSTEM REQUIREMENTS**

- ❖ Windows 32/64 bit 7/8/10
- ❖ Microsoft Access 8/10
- ❖ JDK 1.7

# SOFTWARE MAP







# OUTPUT SCREENS

## AND

# SOURCE CODE

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
import javax.swing.*;

public class Base extends JFrame implements ActionListener
{
    static JButton
    btnAttendance, btnStudent, btnStaff, btnSMS, btnFees, btnEnquiry, btnStudAdd, btnStudUp, btnStudDel, btnStaffUp, btnStaffD
    el, btnStaffAdd, btnSubAdd, btnSubDel, btnSubUp, btnSubject, btnReport, btnAbout;
    static JPanel areapanel, bpanel, mpanel, studpanel, staffpanel, subpanel, panel1;
    static String username = "anonymous";
    static String password = "guest";
    static String url = "jdbc:odbc:Students";
    static Statement st;
    static Connection con;
    CardLayout cl ;
    Base()
    {
        super("Pinnacle Management System");
        try {
            UIManager.setLookAndFeel("com.seaglasslookandfeel.SeaGlassLookAndFeel");
        } catch (Exception e) {
            e.printStackTrace();
        }
        areapanel = new JPanel();
        areapanel.setLayout(new GridLayout(12,1,5,5));
        btnAttendance=new JButton("Attendance");
        btnStudent=new JButton("Student");
        btnStaff=new JButton("Staff");
        btnSMS=new JButton("SMS");
        btnFees=new JButton("Fees");
        btnEnquiry=new JButton("Enquiry");
        btnSubject=new JButton("Subject");
        btnReport=new JButton("Daily Report");
        btnAbout=new JButton("About");

        areapanel.add(btnAttendance);
        areapanel.add(btnStudent);
        areapanel.add(btnStaff);

```

```

        areapanel.add(btnSMS);
        areapanel.add(btnFees);
        areapanel.add(btnEnquiry);
        areapanel.add(btnSubject);
        areapanel.add(btnReport);
        areapanel.add(btnAbout);
bpanel = new JPanel();
        mpanel = new JPanel();
        panel1=new JPanel();
        mpanel.setLayout(new CardLayout());
        c1= (CardLayout)(mpanel.getLayout());
        studpanel=new JPanel();
        studpanel.add(btnStudAdd=new JButton("Add Student"));
        studpanel.add(btnStudDel=new JButton("Delete Student"));
        studpanel.add(btnStudUp=new JButton("Update Student"));
        staffpanel=new JPanel();
        staffpanel.add(btnStaffAdd=new JButton("Add Staff"));
        staffpanel.add(btnStaffDel=new JButton("Delete Staff"));
        staffpanel.add(btnStaffUp=new JButton("Update Staff"));
        subpanel=new JPanel();
        subpanel.add(btnSubAdd=new JButton("Add Subject"));
        subpanel.add(btnSubDel=new JButton("Delete Subject"));
        subpanel.add(btnSubUp=new JButton("Update Subject"));
        mpanel.add(studpanel,"student");
        mpanel.add(subpanel,"subjects");
        mpanel.add(staffpanel,"staff");
        mpanel.add(panel1,"Empty");
bpanel.setLayout(new BorderLayout());
bpanel.add(areapanel,BorderLayout.LINE_START);
        bpanel.add(mpanel,BorderLayout.LINE_END);
add(bpanel);

btnAttendance.addActionListener(this);
        btnStudent.addActionListener(this);
        btnStaff.addActionListener(this);
        btnSMS.addActionListener(this);
        btnFees.addActionListener(this);
        btnEnquiry.addActionListener(this);
        btnAbout.addActionListener(this);
        btnSubject.addActionListener(this);
        btnReport.addActionListener(this);
        btnStudAdd.addActionListener(this);
        btnStudDel.addActionListener(this);
        btnStudUp.addActionListener(this);
        btnStaffAdd.addActionListener(this);
        btnStaffDel.addActionListener(this);
        btnStaffUp.addActionListener(this);

```

```

        btnSubAdd.addActionListener(this);
        btnSubDel.addActionListener(this);
        btnSubUp.addActionListener(this);
        setVisible(true);
        setSize(500,500);
        c1.show(mpanel,"Empty");
    }

    public void actionPerformed(ActionEvent ae){
        if(ae.getSource()==btnStudent)
        {
            c1.show(mpanel,"student");

        }
        if(ae.getSource()==btnStaff)
        {
            c1.show(mpanel,"staff");

        }
        if(ae.getSource()==btnSubject)
        {
            c1.show(mpanel,"subjects");

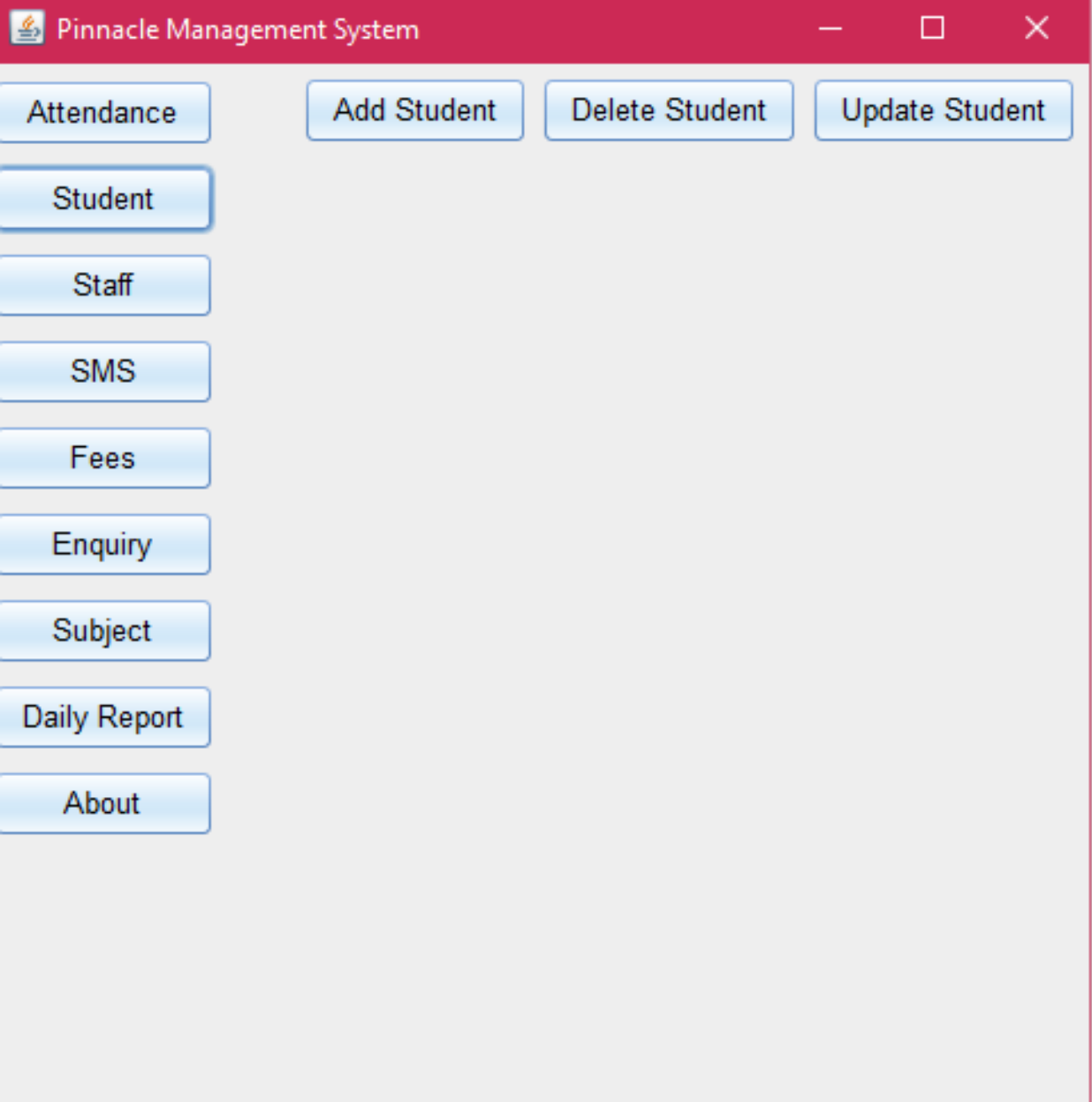
        }
        if(ae.getSource()==btnFees)
        {
            new Fees();

        }
        if(ae.getSource()==btnAbout)
            new About();

        if(ae.getSource()==btnAttendance);
        //new Attendance();
        if(ae.getSource()==btnEnquiry)
            new AddEnquiry();
        if(ae.getSource()==btnSMS)
            new SMS();
        if(ae.getSource()==btnReport)
            new DailyRecord().GUI();
        if(ae.getSource()==btnStaffAdd)
            new AddRecord1();
        if(ae.getSource()==btnStaffDel)
            new DeleteRecord1();
        if(ae.getSource()==btnStaffUp)
            new UpdateRecord1();
        if(ae.getSource()==btnStudAdd)
            new AddRecord();
        if(ae.getSource()==btnStudDel)
            new DeleteRecord();
    }

```

```
        if(ae.getSource()==btnStudUp)
            newUpdateRecord();
        if(ae.getSource()==btnSubAdd)
            newAddSubject();
        if(ae.getSource()==btnSubDel)
            newDeleteSubject();
        if(ae.getSource()==btnSubUp)
            newUpdateSubject();
    }
    public static void main(String[] args) {
        Base lapp =new Base();
        lapp.addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e){
                System.exit(0);
            }
        });
    }
```



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

```

import java.util.*;
import javax.swing.*;

public class DeleteRecord extends JFrame implements ActionListener{

    static JTextField txtID;
        static JLabel txtFname, txtLname, txtMname, txtCourse;
    static JButton btnDelete, btnSearch;
    static JPanel areapanel, bpanel, tpanel;
    static String username = "anonymous";
    static String password = "guest";
    static String url = "jdbc:odbc:Students";
    static Statement st;
    static Connection con;

    DeleteRecord(){
        super("Search the record to Delete. . .");
        try {
            UIManager.setLookAndFeel("com.seaglasslookandfeel.SeaGlassLookAndFeel");
        } catch (Exception e) {
            e.printStackTrace();
        }
        areapanel = new JPanel();
        tpanel = new JPanel();
        tpanel.setLayout(new GridLayout(1,1));
        tpanel.add(btnDelete = new JButton("Delete Record"));

        areapanel.setLayout(new GridLayout(6,2,5,5));
        areapanel.add(new JLabel("UNIQUE ID"));
        areapanel.add(txtID = new JTextField(10));
        areapanel.add(new JLabel(""));
        areapanel.add(btnSearch = new JButton("Search"));
        areapanel.add(new JLabel("Last Name:"));
        areapanel.add(txtLname = new JLabel());
        areapanel.add(new JLabel("First Name:"));
        areapanel.add(txtFname = new JLabel());
        areapanel.add(new JLabel("Middle Name:"));
        areapanel.add(txtMname = new JLabel());
        areapanel.add(new JLabel("Course:"));
        areapanel.add(txtCourse = new JLabel());

        bpanel = new JPanel();
        bpanel.setLayout(new BorderLayout());
        bpanel.add(areapanel, BorderLayout.NORTH);
        bpanel.add(new JLabel(""), BorderLayout.CENTER);
        bpanel.add(tpanel, BorderLayout.SOUTH);
    }
}

```

```

add(bpanel);
btnSearch.addActionListener(this);
btnDelete.addActionListener(this);
        setVisible(true);
        setSize(500,500);
    }

public static void retrieveRecords(){
con = getConnection();
    String result = null;
    String fname = null,mname = null, lname = null, course = null;
    String query = "Select * from tblStudents where ID like " + txtID.getText() + "";
    try{
st = con.createStatement();
ResultSetrs = st.executeQuery(query);
while(rs.next()){
lname = rs.getString("LastName");
fname = rs.getString("FirstName");
mname = rs.getString("MiddleName");
course = rs.getString("Course");
        //result = name;
    }
st.close();
con.close();
    }
catch(SQLException ex){
System.err.println(ex.getMessage());
    }
txtFname.setText(fname);
txtCourse.setText(course);
txtLname.setText(lname);
txtMname.setText(mname);

    if(txtFname.getText().equals("") &&txtMname.getText().equals("") &&txtLname.getText().equals("")
&&txtCourse.getText().equals("")){
JOptionPane.showMessageDialog(null,"Record Not Found!");

txtFname.setText("");
txtCourse.setText("");
txtLname.setText("");
txtMname.setText("");
txtID.setText("");}
    }

public static Connection getConnection(){
    try{

```



```

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con = DriverManager.getConnection(url,username,password);
    }
catch(java.lang.ClassNotFoundException e){
System.err.println(e.getMessage());
    }
catch(SQLException ex){
System.err.println(ex.getMessage());
    }
return con;
    }

public static void deleteRecords(){
con = getConnection();
    String delete1 = "Delete from tblStudents where ID='"+ txtID.getText() +"'";

    try{
st = con.createStatement();
st.executeUpdate(delete1);
st.close();
con.close();
    }
catch(SQLException ex){
System.err.println(ex.getMessage());
    }
JOptionPane.showMessageDialog(null,"Update Finished!");
txtFname.setText("");
txtCourse.setText("");
txtLname.setText("");
txtMname.setText("");
txtID.setText("");
    }

public void actionPerformed(ActionEvent e){
if(e.getSource() == btnSearch){
retrieveRecords();
    }
if(e.getSource() == btnDelete){
deleteRecords();
    }
}

```

Search the record to Delete. . .

UNIQUE ID

Last Name:

First Name:

Middle Name:

Course:

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

```

import javax.swing.*;

public class UpdateRecord1 extends JFrame implements ActionListener
{
    static JTextField txtID,txtFname,txtLname,txtMname,txtCourse;
    static JButton btnUpdate,btnSearch;
    static JPanel areapanel,bpanel,tpanel;
    static String username = "anonymous";
    static String password = "guest";
    static String url = "jdbc:odbc:Students";
    static Statement st;
    static Connection con;
    UpdateRecord1()
    {
        super("Search the record to update. . .");
        try {
            UIManager.setLookAndFeel("com.seaglasslookandfeel.SeaGlassLookAndFeel");
        } catch (Exception e) {
            e.printStackTrace();
        }
        areapanel = new JPanel();
        tpanel = new JPanel();
        tpanel.setLayout(new GridLayout(1,1));
        tpanel.add(btnUpdate = new JButton("Update Record"));
        areapanel.setLayout(new GridLayout(6,2,5,5));
        areapanel.add(new JLabel("UNIQUE ID"));
        areapanel.add(txtID = new JTextField(10));
        areapanel.add(new JLabel(""));
        areapanel.add(btnSearch = new JButton("Search"));
        areapanel.add(new JLabel("Last Name:"));
        areapanel.add(txtLname = new JTextField(10));
        areapanel.add(new JLabel("First Name:"));
        areapanel.add(txtFname = new JTextField(10));
        areapanel.add(new JLabel("Middle Name:"));
        areapanel.add(txtMname = new JTextField(10));
        areapanel.add(new JLabel("Course:"));
        areapanel.add(txtCourse = new JTextField(10));
        bpanel = new JPanel();
        bpanel.setLayout(new BorderLayout());
        bpanel.add(areapanel,BorderLayout.NORTH);
        bpanel.add(new JLabel(""),BorderLayout.CENTER);
        bpanel.add(tpanel,BorderLayout.SOUTH);
        add(bpanel);
        btnSearch.addActionListener(this);
        btnUpdate.addActionListener(this);
        setVisible(true);
        setSize(500,500);
    }
}

```

```

}
public static void retrieveRecords()
{
    con = getConnection();
    String result = null;
    String fname = null, mname = null, lname = null, course = null;
    String query = "Select * from tblStaff where ID like '" + txtID.getText() + "'";
    try
    {
        st = con.createStatement();
        ResultSet rs = st.executeQuery(query);
        while(rs.next())
        {
            lname = rs.getString("LastName");
            fname = rs.getString("FirstName");
            mname = rs.getString("MiddleName");
            course = rs.getString("Course");
            //result = name;
        }
        st.close();
        con.close();
    }
    catch(SQLException ex)
    {
        System.err.println(ex.getMessage());
    }
    txtFname.setText(fname);
    txtCourse.setText(course);
    txtLname.setText(lname);
    txtMname.setText(mname);

    if(txtFname.getText().equals("") && txtMname.getText().equals("") && txtLname.getText().equals("") &&
    txtCourse.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Record Not Found!");

        txtFname.setText("");
        txtCourse.setText("");
        txtLname.setText("");
        txtMname.setText("");
        txtID.setText("");
    }
}

public static Connection getConnection()
{
    try{
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con = DriverManager.getConnection(url, username, password);
    }
}

```

```

catch(java.lang.ClassNotFoundException e){
    System.err.println(e.getMessage());
}
catch(SQLException ex){
    System.err.println(ex.getMessage());
}
return con;
}

public static void updateRecords()
{
    con = getConnection();
    String update1 = "Update tblStaff set FirstName = " + txtFname.getText() + " where ID = " + txtID.getText() + "" ;
    String update2 = "Update tblStaff set MiddleName = " + txtMname.getText() + " where ID = " + txtID.getText() + "" ;
    "" ;
    String update3 = "Update tblStaff set LastName = " + txtLname.getText() + " where ID = " + txtID.getText() + "" ;
    String update4 = "Update tblStaff set Course = " + txtCourse.getText() + " where ID = " + txtID.getText() + "" ;
    try
    {
        st = con.createStatement();
        st.executeUpdate(update1);
        st.executeUpdate(update2);
        st.executeUpdate(update3);
        st.executeUpdate(update4);
        st.close();
        con.close();
    }
    catch(SQLException ex)
    {
        System.err.println(ex.getMessage());
    }
    JOptionPane.showMessageDialog(null,"Update Finished!");
    txtFname.setText("");
    txtCourse.setText("");
    txtLname.setText("");
    txtMname.setText("");
    txtID.setText("");
}

public void actionPerformed(ActionEvent e)
{
    if(e.getSource() == btnSearch)
    {
        retrieveRecords();
    }
    if(e.getSource() == btnUpdate)
    {
        updateRecords();
    }
}

```

}

}

▼

Search the record to update. . .

—

□

✕

UNIQUE ID

Search

Last Name:

First Name:

Middle Name:

Course:

Update Record

Cancel

```

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import java.sql.*;
import java.io.*;
import java.net.*;

class SMS extends JFrame implements ActionListener
{
    static JTextArea msg;
    static JComboBox jbranch,jyear,jsubject;
    static String username = "anonymous";
    static String password = "guest";
    static String url = "jdbc:odbc:Students";
    static Statement st;
    static int count;
    static JButton btnSubmit;
    static String s[]={"CM","CO","IF"};;
    static Connection con;
    static StringBuffer phno;
    static String phno1="";
    JPanel areapanel,bpanel,mpanel;
    SMS()
    {
        super("Pinnacle Management System");
        msg=new JTextArea(10,20);
        areapanel = new JPanel();
        mpanel=new JPanel();
        bpanel=new JPanel();
        areapanel.setLayout(new GridLayout(1,6,5,5));
        areapanel.add(new JLabel("Select Year"));
        String[] y={"1","2","3"};
        areapanel.add(jyear=new JComboBox(y));
        areapanel.add(new JLabel("Select Branch"));
        areapanel.add(jbranch =new JComboBox(s));
        areapanel.add(new JLabel("Select Subject"));
        areapanel.add(jsubject=new JComboBox());
        mpanel.add(new JLabel("Enter Message here"));
        mpanel.add(msg);
        msg.setLineWrap(true);
        bpanel.setLayout(new BorderLayout());
        bpanel.add(areapanel,BorderLayout.NORTH);
        bpanel.add(mpanel,BorderLayout.CENTER);
    }
}

```

```

        bpanel.add(btnSubmit=new JButton("Submit"),BorderLayout.SOUTH);
add(bpanel);
btnSubmit.addActionListener(this);
    jbranch.addActionListener(this);
    jsubject.addActionListener(this);
    jyear.addActionListener(this);
    setVisible(true);
    setSize(300,300);

}
public static void main(String args[])
{
    new SMS();
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==jyear)
    {
        call_branch();
    }
    if(ae.getSource()==jbranch)
    {
        int itemCount = jsubject.getItemCount();
        for(int i=0;i<itemCount;i++){
            jsubject.removeItemAt(0);}
        call_Sub();
    }
    if(ae.getSource()==btnSubmit)
    {
        con=getConnection();
        String query="Select Phone from tblStudents where id in( select id from subjects where
subjects='"+ jsubject.getSelectedItemAt(0) + "')";
        try
        {
            st= con.createStatement();
            ResultSet rs=st.executeQuery(query);
            while(rs.next())
            {
                phno1+= "91" +rs.getString("Phone") +
";";
            }
            st.close();

            con.close();
            System.out.println(phno1);
            phno=new StringBuffer(phno1);
            System.out.println(phno.length());
            phno1=phno.substring(0,(phno.length()-1));

```



```

        }
        catch(SQLException ex)
        {
            System.out.println("jjks67");
            System.err.println(ex);
        }
        sendMessage();
    }

}

public void call_branch()
{
    try
    {
        con = getConnection();
        String query="Select distinct(Branches) from course_details ";
        try
        {
            st= con.createStatement();
            ResultSet rs=st.executeQuery(query);
            while(rs.next())
            {
                count++;
            }
            s=new String[count];
            int i=0;
            rs=st.executeQuery(query);
            while(rs.next())
            {
                String temp=rs.getString("Branches");
                if(temp.equals("All"))
                {}
                else{
                    jbranch.addItem(temp);
                }
            }
        }
        catch(SQLException ex)
        {
            System.out.println("jjks");
            System.err.println(ex);
        }

        st.close();
        con.close();
    }
}

```

```

        }
        catch(Exception ob){System.out.println(ob);}

    }
    public void call_Sub()
    {

        try
        {
            con = getConnection();
            String query="Select Subjects from course_details where
Year="+jyear.getSelectedItem()+"and (Branches="+jbranch.getSelectedItem()+"");
            try
            {
                st= con.createStatement();
                ResultSet rs=st.executeQuery(query);
                while(rs.next())
                {
                    jsubject.addItem(rs.getString("Subjects"));
                }
            }
            catch(SQLException ex)
            {
                System.out.println("jjks67");
                System.err.println(ex);
            }

            st.close();
            con.close();
        }
        catch(Exception ob){System.out.println(ob);}

    }

    public static Connection getConnection(){
try{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con = DriverManager.getConnection(url,username,password);
}
catch(java.lang.ClassNotFoundException e){
    System.err.println(e.getMessage());
}
catch(SQLException ex){
    System.err.println(ex.getMessage());
}
return con;
}

```

```

public void sendMessage()
{
    //Your authentication key
String authkey = "99471ATtECiptFe5682e1b8";
//Multiple mobiles numbers separated by comma
String mobiles = phno1;
//Sender ID,While using route4 sender id should be 6 characters long.
String senderId = "PINACD";
//Your message to send, Add URL encoding here.
String message = msg.getText();
//define route
String route="4";

//Prepare Url
URLConnection myURLConnection=null;
URL myURL=null;
BufferedReader reader=null;

//encoding message
String encoded_message=URLEncoder.encode(message);

//Send SMS API
String mainUrl="http://api.msg91.com/api/sendhttp.php?";

//Prepare parameter string
StringBuilder sbPostData= new StringBuilder(mainUrl);
sbPostData.append("authkey="+authkey);
sbPostData.append("&mobiles="+mobiles);
sbPostData.append("&message="+encoded_message);
sbPostData.append("&route="+route);
sbPostData.append("&sender="+senderId);

//final string
mainUrl = sbPostData.toString();
try
{
    //prepare connection
    myURL = new URL(mainUrl);
    myURLConnection = myURL.openConnection();
    myURLConnection.connect();
    reader= new BufferedReader(new InputStreamReader(myURLConnection.getInputStream()));
    //reading response
    String response;
    while ((response = reader.readLine()) != null)
    //print response
    System.out.println(response);
}

```

```
//finally close connection  
reader.close();  
}  
catch (IOException e)  
{  
    e.printStackTrace();  
}  
}
```

The screenshot shows a Java Swing window titled "Pinnacle Management S...". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there are three "Select..." dropdown menus at the top, each followed by a small blue button with a double-headed arrow. Below these is a text label "Enter Message here" in a red, monospaced font. Underneath the label is a large, empty text area with horizontal lines for input. At the bottom of the window is a large, light blue button with the text "Submit" in a bold, black font.

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
import javax.swing.*;

public class Fees extends JFrame implements ActionListener
{
    static String id=null,fname = null,tot = null, paid = null, bal = null,branch=null,pay=null;
    static JButton btnSubmit,btnCancel,btnSearch;
    static JTextField txtSearch,txtID,txtFname,txtLname,txtMname,txtBranch,txtTot,txtBal,txtPaid,txtPay,txtchqno;
    static JRadioButton cash,chq;
    static JPanel areapanel,bpanel,mpanel;
    static JLabel j1;
    static String username = "anonymous";
    static String password = "guest";
    static String url = "jdbc:odbc:Students";
    static Statement st;
    static int count;
    static String s[]={"CM","CO","IF"};;
    static Connection con;
    Fees()
    {
        super("Pinnacle Management System");
        try {
            UIManager.setLookAndFeel("com.seaglasslookandfeel.SeaGlassLookAndFeel");
        } catch (Exception e) {
            e.printStackTrace();
        }

        areapanel = new JPanel();
        areapanel.setLayout(new GridLayout(14,2,5,5));
        areapanel.add(new JLabel("Enter Id Or Name"));
        areapanel.add(txtSearch=new JTextField());
        areapanel.add(new JLabel("ID"));
        areapanel.add(txtID=new JTextField());
        areapanel.add(new JLabel("Name"));
        areapanel.add(txtFname=new JTextField());
        areapanel.add(new JLabel("Branch"));
        areapanel.add(txtBranch=new JTextField());
        areapanel.add(new JLabel("Total Fees"));
    }
}

```

```

        areapanel.add(txtTot=new JTextField());
        areapanel.add(new JLabel("Fees Paid"));
        areapanel.add(txtPaid=new JTextField());
        areapanel.add(new JLabel("Balance"));
        areapanel.add(txtBal=new JTextField());
        areapanel.add(new JLabel("Enter the amount Paid"));
        areapanel.add(txtPay=new JTextField());
        areapanel.add(new JLabel("Paymode"));
        areapanel.add(new JLabel());
        areapanel.add(cash=new JRadioButton("Cash",true));
        areapanel.add(chq=new JRadioButton("Cheque"));
        areapanel.add(j1=new JLabel("Enter Cheque Number"));
        j1.setVisible(false);
        areapanel.add(txtchqno=new JTextField());
        txtchqno.setVisible(false);
        btnSearch=new JButton("Search");
        areapanel.add(btnSearch);
        btnSubmit=new JButton("Submit");
        areapanel.add(btnSubmit);
        ButtonGroup b=new ButtonGroup();
        b.add(cash);
        b.add(chq);
        areapanel.add(btnCancel=new JButton("Cancel"));
        bpanel = new JPanel();
        bpanel.setLayout(new BorderLayout());
        bpanel.add(areapanel,BorderLayout.NORTH);
        add(bpanel);

        btnSearch.addActionListener(this);
        btnCancel.addActionListener(this);
        chq.addActionListener(this);
        cash.addActionListener(this);
        btnSubmit.addActionListener(this);
        setVisible(true);
        setSize(500,600);
    }

    public void actionPerformed(ActionEvent ae){
        String str =ae.getActionCommand();
        if(ae.getSource()== btnCancel)
        {
            this.dispose();
        }
        if(ae.getSource()== btnSubmit)
        {
            pay_fees();
        }
        if(ae.getSource()== btnSearch)
        {

```

```

        retrieveFees();
    }
    if(ae.getSource()==cash)
    {
        j1.setVisible(false);
        txtchqno.setVisible(false);
    }
    if(ae.getSource()==chq)
    {
        j1.setVisible(true);
        txtchqno.setVisible(true);
    }
}
public void pay_fees()
{
    System.out.println("Exec");
    try
    {
        System.out.println(txtPay.getText());
        pay=txtPay.getText();

        int paid1=Integer.parseInt(paid)+Integer.parseInt(pay);
        paid=Integer.toString(paid1);
        int bal1=Integer.parseInt(tot)-Integer.parseInt(paid);
        bal=Integer.toString(bal1);
        System.out.println("56");
        con = getConnection();
        String query="Update tblStudents set fees_paid='"+ paid + "',bal='"+ bal
+"" where ID='"+txtID.getText()+"'";

        try
        {
            st= con.createStatement();
            st.executeUpdate(query);

        }
        catch(SQLException ex)
        {
            System.out.println("jjks");
            System.err.println(ex);
        }

        st.close();
        con.close();
    }
    catch(Exception ob){System.out.println(ob);}
}

```

```

    }
    public static Connection getConnection(){
try{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con = DriverManager.getConnection(url,username,password);
}
catch(java.lang.ClassNotFoundException e){
    System.err.println(e.getMessage());
}
catch(SQLException ex){
    System.err.println(ex.getMessage());
}
return con;
}

    public static void retrieveFees(){
con = getConnection();
String s=txtSearch.getText();
String query = "Select * from tblStudents where ID = '" + s + "' or FirstName = '" + s + "'";
    System.out.println(query);
try{
    st = con.createStatement();
    ResultSet rs = st.executeQuery(query);
    while(rs.next()){
        id=rs.getString("ID");
        branch=rs.getString("Branch");
        fname = rs.getString("FirstName");
        tot = rs.getString("tot_fees");
        paid = rs.getString("fees_paid");
        bal = rs.getString("bal");
    }

st.close();
con.close();
}

catch(SQLException ex){
    System.err.println(ex.getMessage());
}

        txtID.setText(id);
        txtID.setEnabled(false);
txtFname.setText(fname);
        txtFname.setEnabled(false);
txtTot.setText(tot);
        txtTot.setEnabled(false);
txtPaid.setText(paid);
        txtPaid.setEnabled(false);
        txtBal.setText(bal);

```



```
        txtBal.setEnabled(false);
txtBranch.setText(branch);
        txtBranch.setEnabled(false);

    }

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

Pinnacle Management System

X

Enter Id Or Name

ID

Name

Branch

Total Fees

Fees Paid

Balance

Enter the amount Paid

Paymode

☒ Cash

☐ Cheque

Search

Submit

Cancel

i

```
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
import java.awt.*;
```

```

import java.awt.event.*;
import java.sql.*;
import java.util.Date;
import javax.swing.*;

public class AddEnquiry extends JFrame implements ActionListener
{

static JTextField txtDate,txtSem,txtSname,txtPhno,txtEmail,txtCollege,txtBranch,txtCourse,txtStaff;
    static String date;
    static JTextArea txtAddr;
static JButton btnAdd;
    static JComboBox Sem,Branch,Course;
    static JCheckBox sub_check[]=new JCheckBox[6];
    static String[] sub;
static JPanel areapanel,bpanel;
static String username = "anonymous";
static String password = "guest";
static String url = "jdbc:odbc:Students";
static Statement st;
static Connection con;
    static JCheckBox sub1,sub2,sub3,sub4,sub5,sub6;
    static int count=0,tot_fees=0,i=0,count1=0;
    static String msg="";
public AddEnquiry()
    {
super("Enquiry");
        try {
UIManager.setLookAndFeel("com.seaglasslookandfeel.SeaGlassLookAndFeel");
        } catch (Exception e) {
            e.printStackTrace();
        }

        areapanel = new JPanel();
areapanel.setLayout(new GridLayout(8,1,5,5));

        areapanel.add(new JLabel("Student Name:"));
areapanel.add(txtSname = new JTextField(10));

        String DATE_FORMAT_NOW = "yyyy-MM-dd";
        Date date = new Date();
        SimpleDateFormat sdf = new SimpleDateFormat(DATE_FORMAT_NOW);
        String stringDate = sdf.format(date );

        areapanel.add(new JLabel("Date:"));
areapanel.add(txtDate = new JTextField(10));
        txtDate.setEditable(false);

```

```

        txtDate.setText(stringDate);

        areapanel.add(new JLabel("Semester:"));
        String [] a={"I","II","III","IV","V","VI"};
areapanel.add(Sem = new JComboBox(a));

        areapanel.add(new JLabel("Course:"));
        String [] c={"Diploma","Degree"};
areapanel.add(Course = new JComboBox(c));

areapanel.add(new JLabel("Branch:"));
        String [] b={"CM","CO","EJ","ET"};
areapanel.add(Branch = new JComboBox(b));
        call_branch();
        areapanel.add(new JLabel("Email:"));
        areapanel.add(txtEmail = new JTextField(10));

        areapanel.add(new JLabel("College:"));
areapanel.add(txtCollege = new JTextField(10));

        areapanel.add(new JLabel(" Address:"));
        areapanel.add(txtAddr = new JTextArea(5,10));

        areapanel.add(new JLabel("Phone Number:"));
        areapanel.add(txtPhno = new JTextField(10));

areapanel.add(new JLabel("Guiding Staff:"));
        areapanel.add(txtStaff = new JTextField(10));

areapanel.add(btnAdd = new JButton("Add Enquiry"));

bpanel = new JPanel();
bpanel.setLayout(new BorderLayout());
bpanel.add(areapanel,BorderLayout.NORTH);
add(bpanel);

btnAdd.addActionListener(this);
        Branch.addActionListener(this);
        setVisible(true);
        setSize(500,500);
    }

    public void actionPerformed(ActionEvent e)
    {
if(e.getSource() == btnAdd)
        {
                insertEnquiry();

```

```

    }

    }

    public void itemStateChanged(ItemEventie)
    {

    }

public static Connection getConnection(){
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con = DriverManager.getConnection(url,username,password);
    }
catch(java.lang.ClassNotFoundException e){
System.err.println(e);
    }
catch(SQLException ex){
System.err.println(ex);
    }
return con;
    }

public static void insertEnquiry()
{
    System.out.println("called");
con = getConnection();
    String q = "Insert into Enquiry values('"+ txtDate.getText() + "','"+ txtSname.getText() + "','"+
Sem.getSelectedItem() + "','"+ Course.getSelectedItem() + "','"+ Branch.getSelectedItem() + "','"+ txtEmail.getText() +
 "','"+ txtCollege.getText() + "','"+ txtAddr.getText() + "','"+ txtPhno.getText() + "','"+ txtStaff.getText() + "')";
try{
st= con.createStatement();
st.executeUpdate(q);
st.close();
con.close();
    }
catch(SQLException ex){
System.err.println(ex);
    }
JOptionPane.showMessageDialog(null,"Successfully inserted a enquiry!");
txtSname.setText("");
txtEmail.setText("");
txtCollege.setText("");
txtAddr.setText("");
txtPhno.setText("");
    txtStaff.setText("");
}

```

```

public void call_branch()
{

    try
    {

        con = getConnection();
        String query="Select distinct(Branches) from course_details ";
        try
        {

            st= con.createStatement();
            ResultSets=st.executeQuery(query);
            while(rs.next())
            {

                String temp=rs.getString("Branches");
                if(temp.equals("All"))
                {}

                else{

                    Branch.addItem(temp);

                }

            }

        }
        catch(SQLException ex)
        {

            System.out.println("jjks");
            System.err.println(ex);

        }

        st.close();
        con.close();

    }
    catch(Exception ob){System.out.println(ob);}

}

public static void main(String args[])
{

    newAddEnquiry();

}

```

Enquiry

Student Name:

Date:

2016-03-26

Semester:

I

Course:

Diploma

Branch:

CM

Email:

College:

Address:

Attendance Add Staff Delete Staff Update Staff

Student  ... Search

Staff

SMS

Fees

Enquiry

Subject

Daily Repo

About

```
import java.awt.*;  
import javax.swing.*;
```



```
public class About extends JFrame{
```

```
    About(){
```

```
        super("Pinnacle Management System");
```

```
        setLayout(new GridLayout(5,1));
```

```
        add(new JLabel("Developer: Ashish Ingale-19,Gaurav Harwani-16,Shubham Thorat-54,Dhiraj Tawani-52"));
```

```
        add(new JLabel("Institute - S.H.M. I.T."));
```

```
        add(new JLabel("Instructor: Mrs. Bharti "));
```

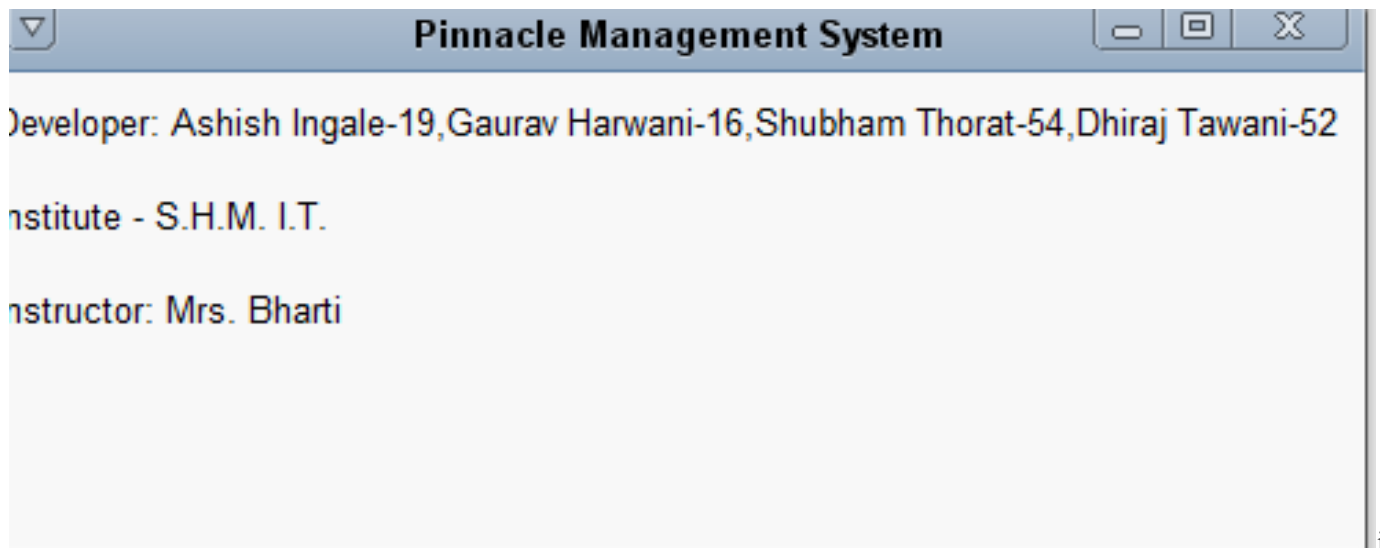
```
            setVisible(true);
```

```
            setSize(500,200);
```

```
            setResizable(false);
```

```
    }
```

```
}
```



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

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

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

```

import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.Statement;
import java.util.*;
import javax.swing.*;
import java.text.*;

public class AddSubject extends JFrame implements ActionListener{
    static JTextField txtID,txtFname,txtLname,txtMname,txtFees,txtBal,txtTot,txtCollege;
    static JButton btnAdd,btnCancel;
        static JComboBox year,Branch,Course;
        static JCheckBox sub_check[]=new JCheckBox[6];
        static String[] sub;
    static JPanel areapanel,bpanel;
    static String username = "anonymous";
    static String password = "guest";
    static String url = "jdbc:odbc:Students";
    static Statement st;
    static Connection con;
        static JCheckBox sub1,sub2,sub3,sub4,sub5,sub6;
        static int count=0,tot_fees=0,i=0,count1=0,a=0;
        static String msg="",msg1="";
    AddSubject(){
        super("ADD Subject");
        try {
            UIManager.setLookAndFeel("com.seaglasslookandfeel.SeaGlassLookAndFeel");
        } catch (Exception e) {
            e.printStackTrace();
        }
        areapanel = new JPanel();
        areapanel.setLayout(new GridLayout(12,2,5,5));
        areapanel.add(new JLabel("UNIQUE ID"));
        areapanel.add(txtID = new JTextField(10));
        areapanel.add(new JLabel("Last Name:"));
        areapanel.add(txtLname = new JTextField(10));
        areapanel.add(new JLabel("First Name:"));
        areapanel.add(txtFname = new JTextField(10));
        areapanel.add(new JLabel("Middle Name:"));
        areapanel.add(txtMname = new JTextField(10));
        areapanel.add(new JLabel("Branch:"));
        String [] b={"CM","CO","EJ","ET"};
        areapanel.add(Branch = new JComboBox(b));
        areapanel.add(new JLabel("Course:"));
        String [] s={"Diploma","Degree"};
        areapanel.add(Course = new JComboBox(s));
        areapanel.add(new JLabel("College:"));

```

```

areapanel.add(txtCollege = new JTextField(10));
areapanel.add(new JLabel("Select Year"));
    String[] y1={"1","2","3"};
    areapanel.add(year = new JComboBox(y1));
    for(i=0;i<sub_check.length;i++){
        sub_check[i]=new JCheckBox();
        areapanel.add(sub_check[i]);
        sub_check[i].setVisible(false);
    }

areapanel.add(btnAdd = new JButton("Add Record"));
    areapanel.add(btnCancel = new JButton("Cancel"));
bpanel = new JPanel();
bpanel.setLayout(new BorderLayout());
bpanel.add(areapanel,BorderLayout.NORTH);
add(bpanel);
    init();

btnAdd.addActionListener(this);
    btnCancel.addActionListener(this);
    year.addActionListener(this);
    Branch.addActionListener(this);
    setVisible(true);
    setSize(500,500);
}
public void actionPerformed(ActionEvent e){
    if(e.getSource() == btnAdd){
        select();

        insertRecord();

    }

    if(e.getSource() == btnCancel)
    {
        new Base();
        this.dispose();
    }
    if(e.getSource() == year){
        for(i=0;i<count1;i++)
            sub[i]="";
        System.out.println(year.getSelectedItem());
        retrieveRecords();
    }
    if(e.getSource() == Branch){
        for(i=0;i<count1;i++)
            sub[i]="";
        System.out.println(Branch.getSelectedItem());
        retrieveRecords();
    }
}

```

```

    }
}

public void select()
{
    tot_fees=0;
    try
    {
        con = getConnection();
        System.out.println(count1);
        for(int i=0;i<count1;i++)
        {
            if(sub_check[i].isSelected())
            {
                msg=sub_check[i].getText()+"";
                String q = "Insert into subjects values('"+ txtID.getText() + "','"+ msg +
""");
                System.out.println(q);
                String query="Select Fees from course_details where Subjects='"+ msg
+ "' and Branches='"+ Branch.getSelectedItemAt() + "'";
                System.out.println(query);
                try
                {
                    st= con.createStatement();
                    st.executeUpdate(q);
                    ResultSet rs=st.executeQuery(query);
                    while(rs.next())
                    {
                        int fees=Integer.parseInt(rs.getString("Fees"));
                        System.out.println(fees);
                        tot_fees = tot_fees + fees;
                    }
                }
                catch(SQLException ex)
                {
                    System.err.println(ex);
                }
            }
        }
        st.close();
        con.close();
    }
    catch(Exception ob){System.out.println(ob);}
}

public static Connection getConnection(){
    try{
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

```

```

        con = DriverManager.getConnection(url,username,password);
    }
    catch(java.lang.ClassNotFoundException e){
        System.err.println(e);
    }
    catch(SQLException ex){
        System.err.println(ex);
    }
    return con;
}

public static void insertRecord(){
    con = getConnection();
        DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd ");
        String q = "Insert into tblStudents Values('"+ msg1 +"','"+ txtLname.getText() + "','" + txtFname.getText() + "','" +
txtMname.getText() + "','" + Branch.getSelectedItemAt() + "','" + Course.getSelectedItemAt() + "','" + tot_fees + "','" +
txtCollege.getText() + "','" +0+"','"+ +0+"','"+ year.getSelectedItemAt() + "','" +dateFormat.format(new Date())+"')";
        try{
            st= con.createStatement();
            st.executeUpdate(q);
            st.close();
            con.close();
        }
        catch(SQLException ex){
            System.err.println(ex);
        }
        JOptionPane.showMessageDialog(null,"Successfully inserted a record!");
        txtFname.setText("");
        txtLname.setText("");
        txtMname.setText("");
        txtID.setText("");
    }

    public static void retrieveRecords(){
        con = getConnection();

        sub=new String[5];

        String query = "Select * from course_details where Year = " + year.getSelectedItemAt() + " and Branches = " +
Branch.getSelectedItemAt() + " ";
        try{
            st = con.createStatement();
            ResultSet rs = st.executeQuery(query);
            while(rs.next()){

                sub[count]=new String(rs.getString("Subjects"));

                System.out.println(sub[count++]);
            }
        }
    }
}

```

```

st.close();
con.close();
}
catch(SQLException ex){
    System.err.println(ex);
}
try{
    System.out.println(count+" "+sub.length);
    count1=count;
    System.out.println(count1);
    for (int i = 0; i <count; i++)
    {
        sub_check[i].setText(sub[i]);
        sub_check[i].setVisible(true);
    }
    count=0;
}
catch(Exception e)
{}

}

public static void init()
{
    System.out.println("Called");
    con = getConnection();

    String q1="Select max(ID) as max from tblStudents";
try{
    st= con.createStatement();
    System.out.println("Called1");
    ResultSet r1=st.executeQuery(q1);
    System.out.println("Called2");
    while(r1.next())
    {
        msg1=r1.getString("max");
        if(msg1==null)
            a=0;
        else
            a=Integer.parseInt(msg1);
        System.out.println(a);
        a++;
        msg1=a+"";
    }
    txtID.setText(msg1);
    txtID.setEnabled(false);
}
}

```

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

ADD Subject

UNIQUE ID

Last Name:

First Name:

Middle Name:

Branch:

CM

Course:

Diploma

College:

Select Year

1

Add Record

Cancel

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;
import javax.swing.*;

public class UpdateSubject extends JFrame implements ActionListener{

    static JTextField txtID,txtname,txtFees,txtBranch;
    static JComboBox Year,Course,Time1,Day;
    static JButton btnUpdate,btnSearch,btnCancel;
    static JPanel areapanel,bpanel,tpanel,mpanel;
    static String username = "anonymous";
    static String password = "guest";
    static String url = "jdbc:odbc:Students",msg="",msg1="";
    static Statement st;
    static Connection con;
    static int count=0,c=0,i=0,tot_fees=0,j=0,bal=0;

    UpdateSubject(){
        super("Search the record to update. . .");
        try {
            UIManager.setLookAndFeel("com.seaglasslookandfeel.SeaGlassLookAndFeel");
        } catch (Exception e) {
            e.printStackTrace();
        }
        areapanel = new JPanel();
        areapanel.setLayout(new GridLayout(14,2,5,5));
        areapanel.add(new JLabel("UNIQUE ID"));
        areapanel.add(txtID = new JTextField(10));
        areapanel.add(new JLabel(""));
        areapanel.add(btnSearch = new JButton("Search"));
        areapanel.add(new JLabel("Subject Name:"));
        areapanel.add(txtname = new JTextField(10));
        areapanel.add(new JLabel("Fees:"));
        areapanel.add(txtFees = new JTextField(10));
        areapanel.add(new JLabel("Day:"));
        String [] day={"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"};
        areapanel.add(Day = new JComboBox(day));
        areapanel.add(new JLabel("Course:"));
        String [] course={"Diploma","Degree"};
        areapanel.add(Course = new JComboBox(course));
        areapanel.add(new JLabel("Select Year"));
        String[] y1={"1","2","3"};
        areapanel.add(Year = new JComboBox(y1));
    }
}

```



```

        areapanel.add(new JLabel("Select Time"));
        areapanel.add(Time1 = new JComboBox());
        areapanel.add(new JLabel("Enter Branch"));
        areapanel.add(txtBranch = new JTextField(10));
areapanel.add(btnUpdate = new JButton("Update Record"));
        areapanel.add(btnCancel = new JButton("Cancel"));
bpanel = new JPanel();
bpanel.setLayout(new BorderLayout());
bpanel.add(areapanel,BorderLayout.NORTH);
add(bpanel);

        call_time();
btnUpdate.addActionListener(this);
        btnCancel.addActionListener(this);
        btnSearch.addActionListener(this);
        Year.addActionListener(this);
        setVisible(true);
        setSize(500,500);
}
public static void retrieveRecords(){
        System.out.println("Called");
con = getConnection();
String result = null;
String name=null,branch=null,day=null,time=null,fee=null,year=null,id=null,course=null;
String query = "Select * from course_details where id =" + txtID.getText() + "";
try{
        st = con.createStatement();
        ResultSet rs = st.executeQuery(query);
        while(rs.next()){
                name = rs.getString("Subjects");
                branch = rs.getString("Branches");
                day = rs.getString("Days");
                course = rs.getString("Course");
                                time = rs.getString("Time");
                                fee = rs.getString("Fees");
                                year = rs.getString("Year");
                                id = rs.getString("id");

        }
st.close();
con.close();
}
catch(SQLException ex){
        System.err.println(ex.getMessage());
}

        System.out.println(name+" "+branch);
txtname.setText(name);
txtBranch.setText(branch);
txtFees.setText(fee);

```

```

txtID.setText(id);

if(txtname.getText().equals("") && txtFees.getText().equals("") && txtBranch.getText().equals("") &&
txtID.getText().equals("")){
    JOptionPane.showMessageDialog(null,"Record Not Found!");

}
}

public static Connection getConnection(){
    try{
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con = DriverManager.getConnection(url,username,password);
    }
    catch(java.lang.ClassNotFoundException e){
        System.err.println(e.getMessage());
    }
    catch(SQLException ex){
        System.err.println(ex.getMessage());
    }
    return con;
}

public static void updateRecords(){
    con = getConnection();
    String update1 = "Update course_details set Subjects = " + txtname.getText() + " where ID = " + txtID.getText() +
"" ;
    String update2 = "Update course_details set Branches = " + txtBranch.getText() + " where ID = " + txtID.getText()
+ "" ;
    String update3 = "Update course_details set Fees = " + txtFees.getText() + " where ID = " + txtID.getText() + "" ;
    String update4 = "Update course_details set Course = " + Course.getSelectedItem() + " where ID = " +
txtID.getText() + "" ;
    String update5 = "Update course_details set Year = " + Year.getSelectedItem() + " where ID = " +
txtID.getText() + "" ;
    String update6 = "Update course_details set Days = " + Day.getSelectedItem() + " where ID = " +
txtID.getText() + "" ;
    try{

        st = con.createStatement();
        st.executeUpdate(update1);
        st.executeUpdate(update2);
        st.executeUpdate(update3);
        st.executeUpdate(update4);
        st.executeUpdate(update5);
        st.executeUpdate(update6);

        st.close();
        con.close();
    }
}

```

```

    }
    catch(SQLException ex){
        System.err.println(ex.getMessage());
    }
    JOptionPane.showMessageDialog(null,"Update Finished!");
    txtname.setText("");
    txtFees.setText("");
    txtBranch.setText("");
    txtID.setText("");
}

public void actionPerformed(ActionEvent e){
    if(e.getSource() == btnSearch){
        retrieveRecords();

    }
    if(e.getSource() == btnUpdate){
        updateRecords();
    }
}

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

public static void call_time()
{
    try
    {
        con = getConnection();
        String query="Select distinct(Time) from course_details ";
        try
        {
            st= con.createStatement();
            ResultSet rs=st.executeQuery(query);
            while(rs.next())
            {
                String temp=rs.getString("Time");
                Time1.addItem(temp);
            }
        }
        catch(SQLException ex)
        {
            System.out.println("jjks");
            System.err.println(ex);
        }
    }
}

```

```
        st.close();
        con.close();
    }
    catch(Exception ob){System.out.println(ob+"123");}
}
```

The image shows a Java Swing window titled "Search the record to update. . .". The window contains a search form with the following fields and controls:

- UNIQUE ID**: A text input field.
- Search**: A button to initiate the search.
- Subject Name:**: A text input field.
- Fees:**: A text input field.
- Day:**: A dropdown menu currently showing "Monday".
- Course:**: A dropdown menu currently showing "Diploma".
- Select Year**: A dropdown menu currently showing "1".
- Select Time**: A dropdown menu.
- Enter Branch**: A text input field.
- Update Record**: A button to update the record.
- Cancel**: A button to cancel the operation.

# **ADVANTAGES**

1. PDMS helps solve the regular problem of updating and entering in a very generic and easy solution. With the help of PDMS one can enter data any time into the database without much problems by just few clicks.
2. There would be no issue of caring while storing data, all we would need to do is just enter the data in the given syntax and your database would be created with a proper manner and security.
3. With the use of PDMS one can easily rely on the software to provide enough security of data, there would be no worry of data loss and data changing or misplace of data, one can just trust this software to ensure the data has been provided with modern securities and proper authentication. One need not to be worried about data loss as regular backups is the topmost priority of the software PDMS.
4. PDMS also is a cost effective software which can be run on our regular computers, there would be no need of external resources, PDMS is a one time investment with guarantee of life time services without the wear and tear of software. It is comparatively very less costly compared to Paperwork.
5. With PDMS one can search for information of any student or enquiry with ease as PDMS would provide searching of data through table parameters like Name, Subject, Course etc. giving the user a very fast response every time they need data and saving their time to go through the Papers..
6. The main aim of PDMS is to provide hassle free methods to update the content in the databases stored by the user. This facility would prove beneficial in updating information like contact no, whenever we want.
7. PDMS would be included with an attendance application which would help the institute track the record of students and their attendance in the lectures and PDMS would notify the parents of the absent students whenever a new record is entered.
8. PDMS would be provided with an all time manual messaging application which would message the next day lecture schedule to the students manual where the user would just have to select the batch and the subject and the students in those batches would be messaged of their schedule automatically, this would also provide of a manual messaging service incase of an irregular change in the schedule.
9. PDMS would also contain a particular institutes fees structure and would be provided with the updated fee whenever fee structure changes. It would ask the user for a discount and would calculate the final fee for the student and update it on their personal accounts where fees balance and marks of regular tests would be shown.

# **FUTURE SCOPE**

- 1. Multiple operable versions of the software.**
- 2. Connection to cloud storage.**
- 3. Connected to client server.**
- 4. Automatic messaging.**
- 5. Platform change to web-apps.**



# **Conclusion**

The Conclusion reached after creating the current version of the software to meet the system objectives. The comparison is done between the system that was built and original requirements that were designed at the beginning of the project. It also describes the Future Work that is intended to be accomplished with later versions of the software.

The main objective of the application is to help computer science students understand the basics of Java, JavaScript, and HTML. By browsing through the application and looking at the code for each graphical interpretation, students should be able to easily understand the implementation. The following results have been achieved after the completing the system and relate back to the system's objective.

- 1. Should allow computer science students to browse through the code and application:** This is achieved when users, i.e., computer science students, are able to run and install the application. When they run the application, they can browse through the implementation of different objects.
- 2. Should allow users to browse through different product categories:** This is achieved when the user first runs the application and is directed to a home page where they can enter truck details and view the services provided.

# **REFERENCES**

### **Books Referred:**

- 1. Complete reference – JAVA**
- 2. Java API 1.7**
- 3. Balguruswamy**
- 4. And internet websites such as: [www.oracle.com](http://www.oracle.com)**