ATTENDANCE-MANAGEMENT SYSTEM (USING JAVA)

School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY PHAGWARA, PUNJAB

April 2023

S.NO	NAME	SECTION	REG.NO	ROLL.NO
1	B. Leeladhar Reddy	K21HP	12103455	RK21HPA17
2	K. Lakshmieswar Reddy	K21HP	12107096	RK21HPA16
3	P. Karthik	K21HP	12006339	RK21HPA29

INTRODUCTION :-

An attendance management system is a software application that allows organizations to track the attendance of their employees or students. It provides an efficient and automated way to record and manage attendance, reducing the need for manual processes such as taking roll calls or maintaining spreadsheets.

The system typically uses some form of identification, such as a biometric scanner, RFID card, or password, to verify the identity of the individual and record their attendance. The data is then stored in a centralized database, which can be accessed and analyzed by authorized personnel to generate reports, monitor attendance trends, and identify patterns or issues that may require attention.

An attendance management system can offer numerous benefits, including increased accuracy, efficiency, and accountability, as well as improved compliance with labor laws and regulations. It can also help organizations save time and resources, reduce errors and discrepancies, and improve communication and transparency with employees or students.

Overall, an attendance management system can help organizations streamline their attendance tracking processes and make more informed decisions based on real-time data, ultimately contributing to better performance, productivity, and profitability.

OBJECTIVES :-

Accurate and Timely Attendance Recording: One of the primary objectives of an attendance management system is to accurately and efficiently record the attendance of employees or students. This helps ensure that accurate records are maintained, and time and attendance data is available in real-time.

Reduction in Manual Processes: Attendance management systems aim to reduce the need for manual processes such as taking roll calls or maintaining spreadsheets. This results in time savings, reduces errors and discrepancies, and improves efficiency.

Compliance with Labor Laws and Regulations: Attendance management systems help organizations comply with labor laws and regulations related to attendance tracking and management. By automating attendance tracking, organizations can ensure that they are meeting legal requirements.

Improved Accountability: An attendance management system can help improve accountability by ensuring that employees or students are aware of their attendance responsibilities and are held accountable for attendance-related issues such as absences or tardiness.

Improved Communication and Transparency: Attendance management systems can improve communication and transparency between employees or students and management. For example, employees or students can access their attendance records, view their schedule, and receive alerts about upcoming shifts or classes.

Enhanced Reporting and Analytics: Attendance management systems can provide organizations with valuable data and insights about attendance trends, patterns, and issues. This data can be used to generate reports, identify areas of improvement, and make informed decisions about attendance policies and practices.

IMPLEMENTATION PROCESS :-

ADDING NEW FACULTY MEMBER

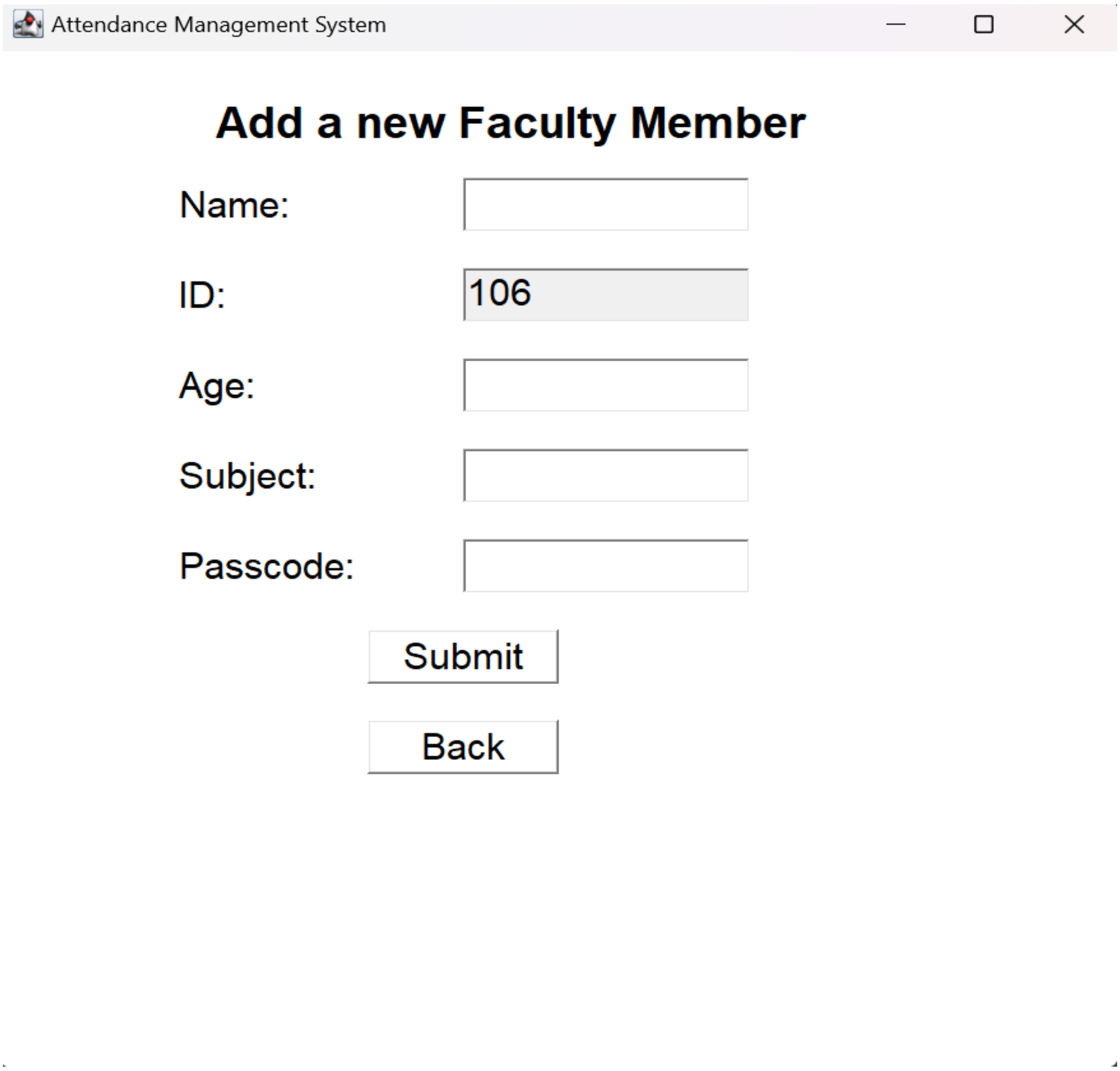
The Add_Faculty class is initialized with a frame and a username as parameters. It first removes all components from the frame and repaints it. Then, it creates various components such as labels, text fields, and buttons to allow the user to enter information about the new faculty member. The ifNumber class is used to search for an available ID for the new faculty member. The Add_to_base class is called when the user submits the information, and it creates a connection to the database and inserts the new faculty member's information into it.

CODE:-

```
package Test;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.JOptionPane;

public class Add_Faculty {
        Add_Faculty(Frame f, String username){
                f.removeAll();
                f.repaint();
                int ID = new ifNumber().searchIdforFaculty();
                Label l1 = new Label("Add a new Faculty Member");
                l1.setFont(new Font("Arial", Font.BOLD, 24));
          l1.setAlignment(Label.CENTER);
          l1.setBounds(100,50,350,40);
          f.add(l1);
                Label nameLabel = new Label("Name:");
    Label idLabel = new Label("ID:");
    Label ageLabel = new Label("Age:");
    Label subjectLabel = new Label("Subject:");
    Label passcodeLabel = new Label("Passcode:");
    TextField nameField = new TextField();
    TextField idField = new TextField(""+ID);
    idField.setEditable(false);
    TextField ageField = new TextField();
    TextField subjectField = new TextField();
    TextField passcodeField = new TextField();
    Font font = new Font("Times new roman", Font.PLAIN, 20);
    nameLabel.setBounds(100, 100, 100, 30);
    nameLabel.setFont(font);
    idLabel.setBounds(100, 150, 100, 30);
    idLabel.setFont(font);
    ageLabel.setBounds(100, 200, 100, 30);
    ageLabel.setFont(font);
    subjectLabel.setBounds(100, 250, 100, 30);
    subjectLabel.setFont(font);
    passcodeLabel.setBounds(100, 300, 100, 30);
    passcodeLabel.setFont(font);
    nameField.setBounds(250, 100, 150, 30);
    nameField.setFont(font);
    idField.setBounds(250, 150, 150, 30);
    idField.setFont(font);
    ageField.setBounds(250, 200, 150, 30);
    ageField.setFont(font);
    subjectField.setBounds(250, 250, 150, 30);
    subjectField.setFont(font);
    passcodeField.setBounds(250, 300, 150, 30);
    passcodeField.setFont(font);
    f.add(nameLabel);
    f.add(idLabel);
    f.add(ageLabel);
    f.add(subjectLabel);
    f.add(passcodeLabel);
    f.add(nameField);
    f.add(idField);
    f.add(ageField);
    f.add(subjectField);
    f.add(passcodeField);
    Button b1 = new Button("Back");
    b1.setFont(font);
    b1.setBounds(200,400,100,30);
    f.add(b1);
    b1.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        new Admin_after_login(f,username);
      }
    });
    Button b2 = new Button("Submit");
    b2.setFont(font);
    b2.setBounds(200,350,100,30);
    f.add(b2);
    b2.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        String id = idField.getText();
        String age = ageField.getText();
        String subject = subjectField.getText();
        String pass = passcodeField.getText();
        if(name.length() == 0) {
                JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
        }
        else if(idField.getText().length() == 0) {
                JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
        }
        else if(ageField.getText().length() == 0) {
                JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
        }
        else if(subject.length() == 0) {
                JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
        }
        else if(pass.length() == 0) {
                JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
        }
        else {
                if(!name.matches("^[a-zA-Z]*$")) {
                        JOptionPane.showMessageDialog(f, " Only Alphabets are allowed in Name", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else if(!new ifNumber().ifnumber(id)) {
                        JOptionPane.showMessageDialog(f, " Only numbers are allowed in Id", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else if(!new ifNumber().ifnumber(age)) {
                        JOptionPane.showMessageDialog(f, " Only numbers are allowed in Age", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else if(!subject.matches("^[a-zA-Z]*$")) {
                        JOptionPane.showMessageDialog(f, " Only Alphabets are allowed in Name", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else {
                        new Add_to_base(f,name,Integer.parseInt(id),Integer.parseInt(age),subject,pass,username);
                }
        }
      }
```

```
  });
    }
}

class Add_to_base{
    Add_to_base(Frame f,String name,int id,int age,String subject,String password,String username){
        Connection con;
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/ams","root","@LEELadhar2719");
            PreparedStatement pstmt = con.prepareStatement("select user_id from faculty_login_details where
user_id = ?;");
            pstmt.setInt(1, id);
            ResultSet r = pstmt.executeQuery();
            if(r.next()) {
                JOptionPane.showMessageDialog(f, " Duplicate Value ", "Message",
JOptionPane.INFORMATION_MESSAGE);
                con.close();
            }
            pstmt = con.prepareStatement("insert into faculty_details values (?, ?, ?, ?, ?);");
            PreparedStatement pstmt1 = con.prepareStatement("insert into faculty_login_details values (?, ?);");
         pstmt.setString(1, name);
         pstmt.setInt(2, id);
         pstmt.setInt(3, age);
         pstmt.setString(4, subject);
         pstmt.setString(5, password);
         pstmt1.setInt(1, id);
         pstmt1.setString(2, password);
         int rs = pstmt.executeUpdate();
         int rs1 = pstmt1.executeUpdate();
         JOptionPane.showMessageDialog(f, " Details Saved ", "Message", JOptionPane.INFORMATION_MESSAGE);
         new Add_Faculty(f, username);
         con.close();
        }
        catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

class ifNumber{
    boolean ifnumber(String s) {
        char[] a = s.toCharArray();
        for(char t : a) {
            if(t <'0' || t>'9') return false;
        }
        return true;
    }
    int searchIdforFaculty() {
        int cnt = 0;
        Connection con;
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/ams","root","@LEELadhar2719");
            PreparedStatement p = con.prepareStatement("SELECT COUNT(*) as cnt FROM faculty_details;");
            ResultSet r = p.executeQuery();
            if(r.next()) {
                cnt = r.getInt(1);
            }
            cnt += 100;
        }
        catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return cnt;
    }
    int searchIdforStudent() {
        int cnt = 0;
        Connection con;
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/ams","root","@LEELadhar2719");
            PreparedStatement p = con.prepareStatement("SELECT COUNT(*) as cnt FROM students;");
            ResultSet r = p.executeQuery();
            if(r.next()) {
                cnt = r.getInt(1);
            }
            cnt += 1000;
        }
        catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return cnt;
    }
}
```

<u>OUTPUT BY RUNNING</u> :-



Add a new Faculty Member

Name:

ID: 106

Age:

Subject:

Passcode:

ADDING NEW STUDENT

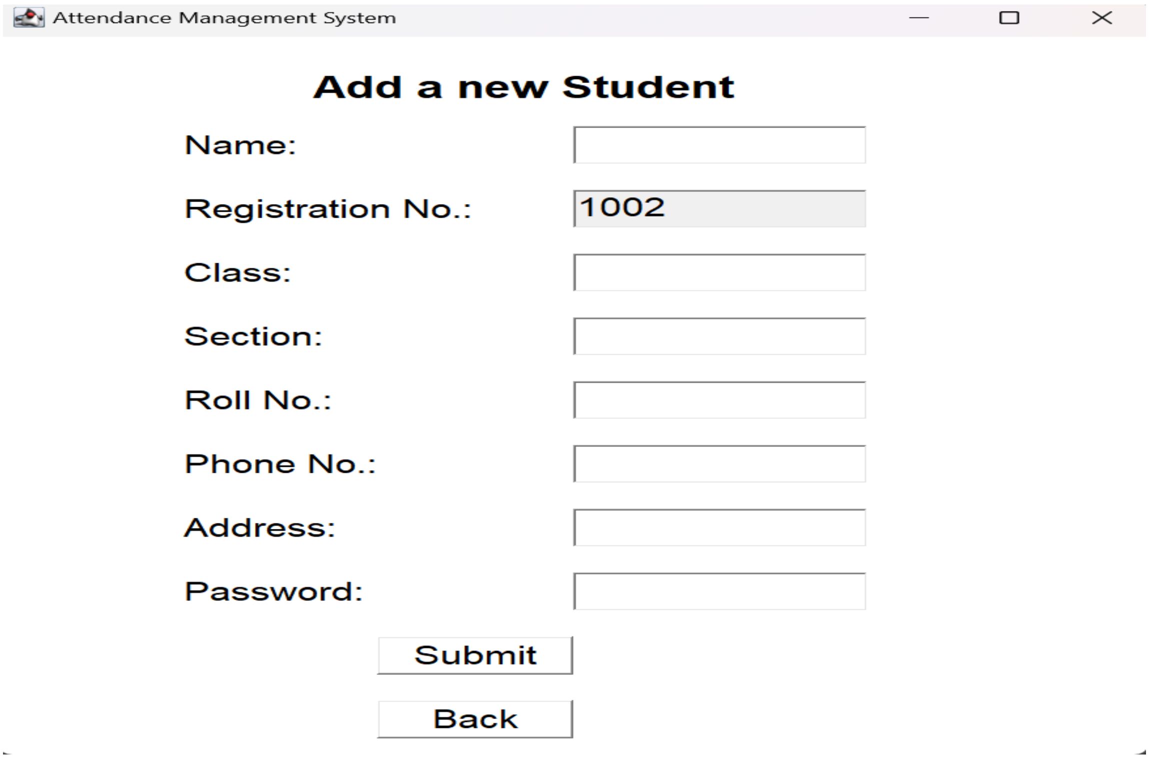
The program uses event listeners to handle button clicks. The "Back" button opens a new window for the admin dashboard, and the "Submit" button performs validation of the entered data and inserts the data into a database. If any of the required fields are left blank, the program displays an error message. Additionally, the program checks if the name field only contains alphabets and displays an error message if it doesn't.

CODE :-

```
package Test;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.JOptionPane;

public class Add_Student_Details {
        Add_Student_Details(Frame f, String username){
                f.removeAll();
                f.repaint();
                int ID = new ifNumber().searchIdforStudent();
                Label l1 = new Label("Add a new Student");
                l1.setFont(new Font("Arial", Font.BOLD, 24));
          l1.setAlignment(Label.CENTER);
          l1.setBounds(100,50,350,40);
          f.add(l1);
                Label name = new Label("Name:");
    Label reg = new Label("Registration No.:");
    Label Class = new Label("Class:");
    Label section = new Label("Section:");
    Label rollno = new Label("Roll No.:");
    Label phone = new Label("Phone No.:");
    Label address = new Label("Address:");
    Label pass = new Label("Password:");
    TextField namet = new TextField();
    TextField regt = new TextField(""+ID);
    regt.setEditable(false);
    TextField Classt = new TextField();
    TextField sectiont = new TextField();
    TextField rollnot = new TextField();
    TextField phonet = new TextField();
    TextField addresst = new TextField();
    TextField passt = new TextField();
    Font font = new Font("Times new roman", Font.PLAIN, 20);
    name.setBounds(100, 100, 150, 30);
    name.setFont(font);
    reg.setBounds(100, 150, 150, 30);
    reg.setFont(font);
    Class.setBounds(100, 200, 150, 30);
    Class.setFont(font);
    section.setBounds(100, 250, 150, 30);
    section.setFont(font);
    rollno.setBounds(100, 300, 150, 30);
    rollno.setFont(font);
    phone.setBounds(100, 350, 150, 30);
    phone.setFont(font);
    address.setBounds(100, 400, 150, 30);
    address.setFont(font);
    pass.setBounds(100, 450, 150, 30);
    pass.setFont(font);
    namet.setBounds(300, 100, 150, 30);
    namet.setFont(font);
    regt.setBounds(300, 150, 150, 30);
    regt.setFont(font);
    Classt.setBounds(300, 200, 150, 30);
    Classt.setFont(font);
    sectiont.setBounds(300, 250, 150, 30);
    sectiont.setFont(font);
    rollnot.setBounds(300, 300, 150, 30);
    rollnot.setFont(font);
    phonet.setBounds(300, 350, 150, 30);
    phonet.setFont(font);
    addresst.setBounds(300, 400, 150, 30);
    addresst.setFont(font);
    passt.setBounds(300, 450, 150, 30);
    passt.setFont(font);
    f.add(name);
    f.add(reg);
    f.add(Class);
    f.add(section);
    f.add(rollno);
    f.add(phone);
    f.add(address);
    f.add(namet);
    f.add(regt);
    f.add(Classt);
    f.add(sectiont);
    f.add(rollnot);
    f.add(phonet);
    f.add(addresst);
    f.add(pass);
    f.add(passt);
    Button b1 = new Button("Back");
    b1.setFont(font);
    b1.setBounds(200,550,100,30);
    f.add(b1);
    b1.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        new Admin_after_login(f,username);
      }
    });
    Button b2 = new Button("Submit");
    b2.setFont(font);
    b2.setBounds(200,500,100,30);
    f.add(b2);
    b2.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        if(namet.getText().length() == 0) {
                JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
        }
        else if(regt.getText().length() == 0) {
                JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
        }
        else if(Classt.getText().length() == 0) {
```

```
                        JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else if(sectiont.getText().length() == 0) {
                        JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else if(rollnot.getText().length() == 0) {
                        JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else if(phonet.getText().length() == 0) {
                        JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else if(addresst.getText().length() == 0) {
                        JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else if(passt.getText().length() == 0) {
                        JOptionPane.showMessageDialog(f, " All Fields are compulsory", "Message",
JOptionPane.INFORMATION_MESSAGE);
                }
                else {
                        if(!namet.getText().matches("^[a-zA-Z]*$")) {
                                JOptionPane.showMessageDialog(f, " Only Alphabets are allowed in Name", "Message",
JOptionPane.INFORMATION_MESSAGE);
                        }
                        else if(!new ifNumber().ifnumber(regt.getText())) {
                                JOptionPane.showMessageDialog(f, " Only numbers are allowed in Id", "Message",
JOptionPane.INFORMATION_MESSAGE);
                        }
                        else if(!Classt.getText().matches("^[a-zA-Z]*$")) {
                                JOptionPane.showMessageDialog(f, " Only Alphabets are allowed in Class", "Message",
JOptionPane.INFORMATION_MESSAGE);
                        }
                        else if(!sectiont.getText().matches("^[a-zA-Z]*$")) {
                                JOptionPane.showMessageDialog(f, " Only Alphabets are allowed in Section", "Message",
JOptionPane.INFORMATION_MESSAGE);
                        }
                        else if(!new ifNumber().ifnumber(rollnot.getText())) {
                                JOptionPane.showMessageDialog(f, " Only numbers are allowed in Roll No", "Message",
JOptionPane.INFORMATION_MESSAGE);
                        }
                        else if(!new ifNumber().ifnumber(phonet.getText())) {
                                JOptionPane.showMessageDialog(f, " Only numbers are allowed in Phone number", "Message",
JOptionPane.INFORMATION_MESSAGE);
                        }
                        else {
                                new Add_to_bases(f,namet.getText(),Integer.parseInt(regt.getText()),Classt.getText(),
                                                sectiont.getText(),Integer.parseInt(rollnot.getText()),phonet.getText(),
                                                addresst.getText(),passt.getText(),username);
                        }
                }
            }
        });
            }
}

class Add_to_bases{
        Add_to_bases(Frame f,String name,int reg,String Class,String section,int rollno,String phone,String address,String
password, String username){
                Connection con;
                try {
                        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/ams","root","@LEELadhar2719");
                        PreparedStatement pstmt = con.prepareStatement("select user_id from student_login_details where
user_id = ?;");
                        pstmt.setInt(1, reg);
                        ResultSet r = pstmt.executeQuery();
                        if(r.next()) {
                                JOptionPane.showMessageDialog(f, " Duplicate Value ", "Message",
JOptionPane.INFORMATION_MESSAGE);
                                con.close();
                        }
                        pstmt = con.prepareStatement("insert into students values (?, ?, ?, ?, ?, ?, ?, ?);");
                        PreparedStatement pstmt1 = con.prepareStatement("insert into student_login_details values (?, ?);");
                  pstmt.setString(1, name);
                  pstmt.setInt(2, reg);
                  pstmt.setString(3, Class);
                  pstmt.setString(4, section);
                  pstmt.setInt(5, rollno);
                  pstmt.setString(6, phone);
                  pstmt.setString(7, address);
                  pstmt.setString(8, password);
                  pstmt1.setInt(1, reg);
                  pstmt1.setString(2, password);
                  int rs = pstmt.executeUpdate();
                  int rs1 = pstmt1.executeUpdate();
                  JOptionPane.showMessageDialog(f, " Details Saved ", "Message",
JOptionPane.INFORMATION_MESSAGE);
                  new Add_Student_Details(f,username);
                  con.close();
                }
                catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
```

OUTPUT BY RUNNING :-



Attendance Management System

Add a new Student

Name:

Registration No.: 1002

Class:

Section:

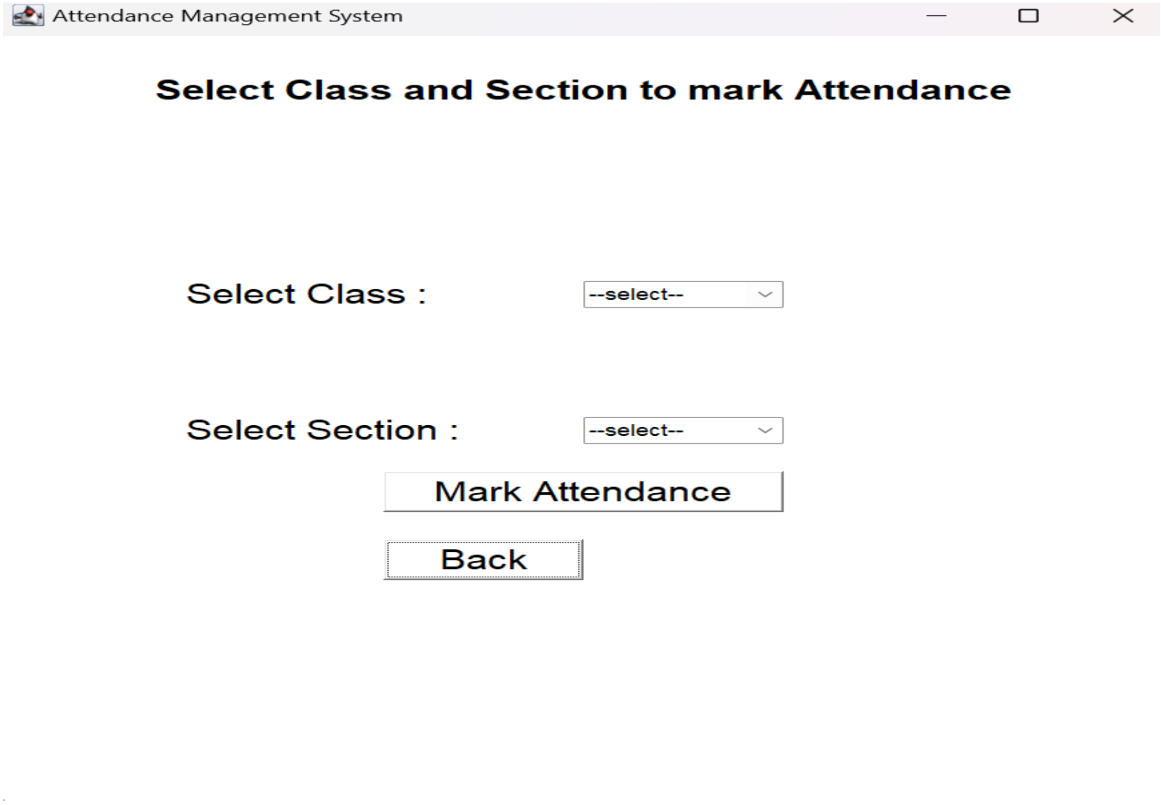
Roll No.:

Phone No.:

Address:

Password:

ATTENDANCE MARKING :-

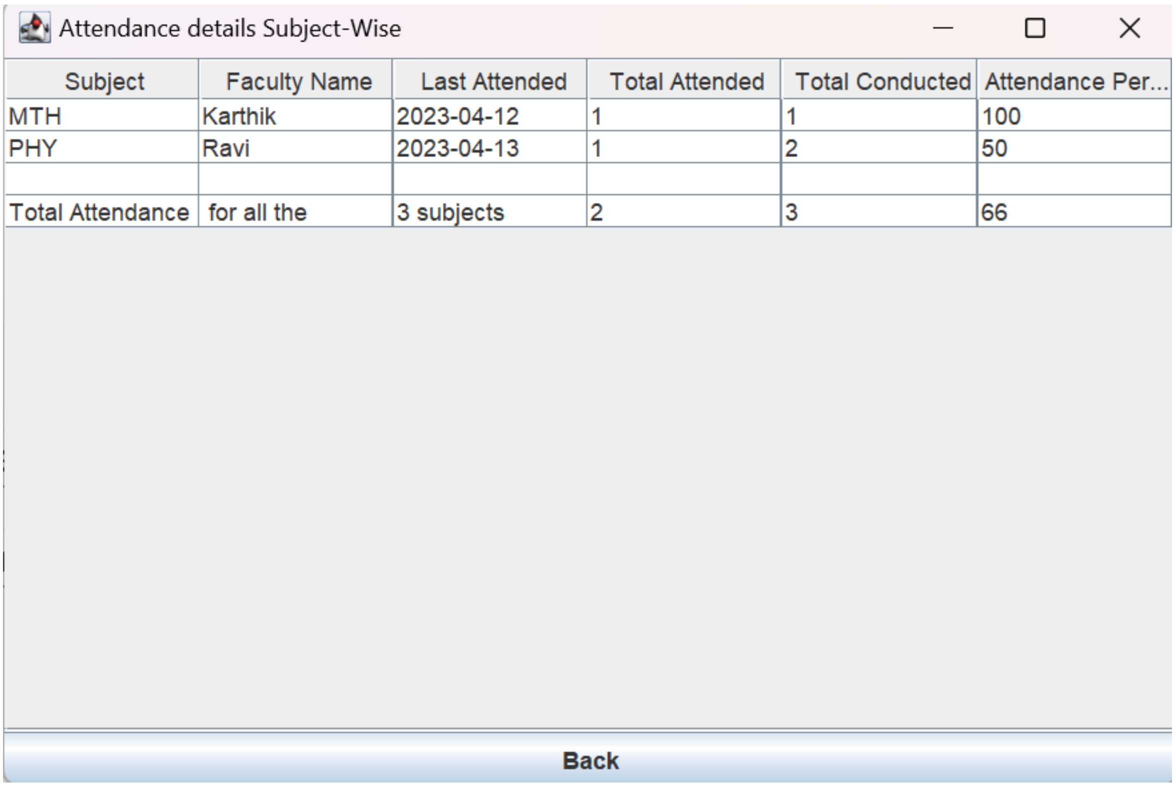


Select Class and Section to mark Attendance

Select Class : --select--

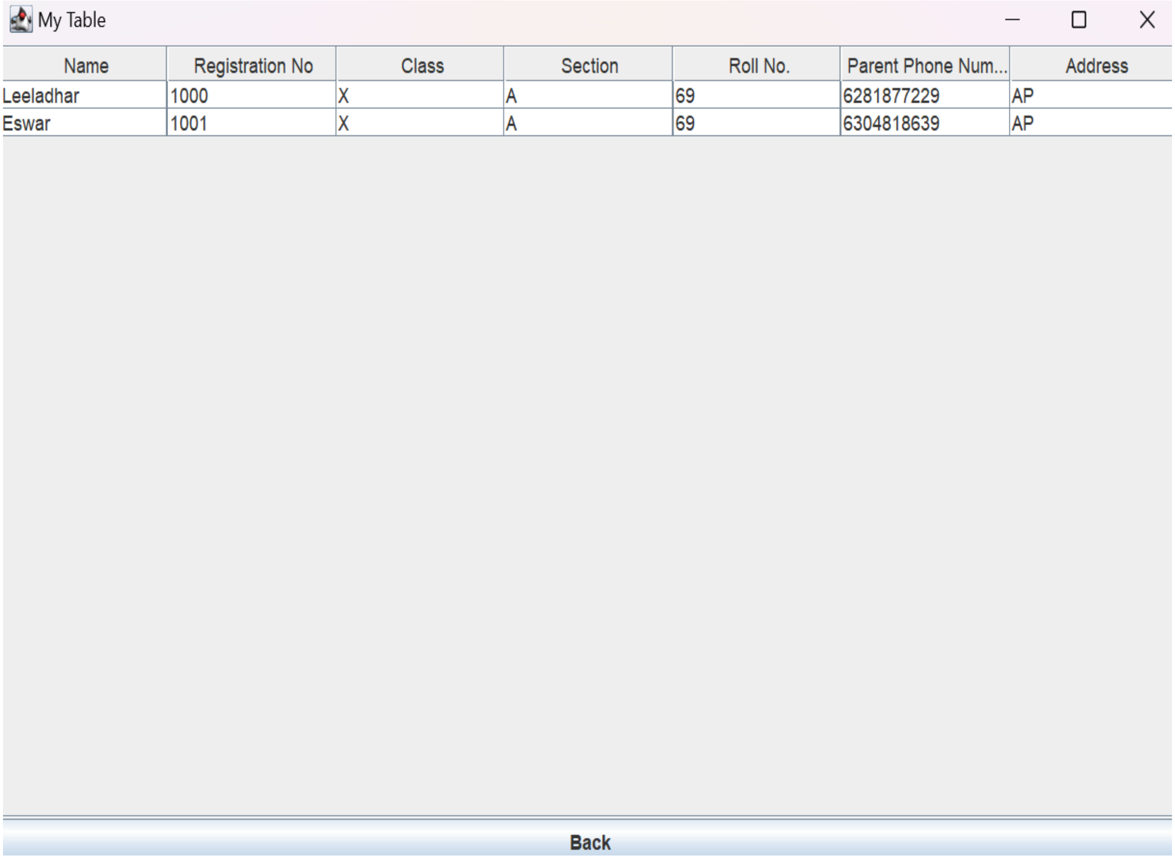
Select Section : --select--

DATA BASE OF FACULTY MEMBERS :-



Subject	Faculty Name	Last Attended	Total Attended	Total Conducted	Attendance Per...
MTH	Karthik	2023-04-12	1	1	100
PHY	Ravi	2023-04-13	1	2	50
Total Attendance	for all the	3 subjects	2	3	66

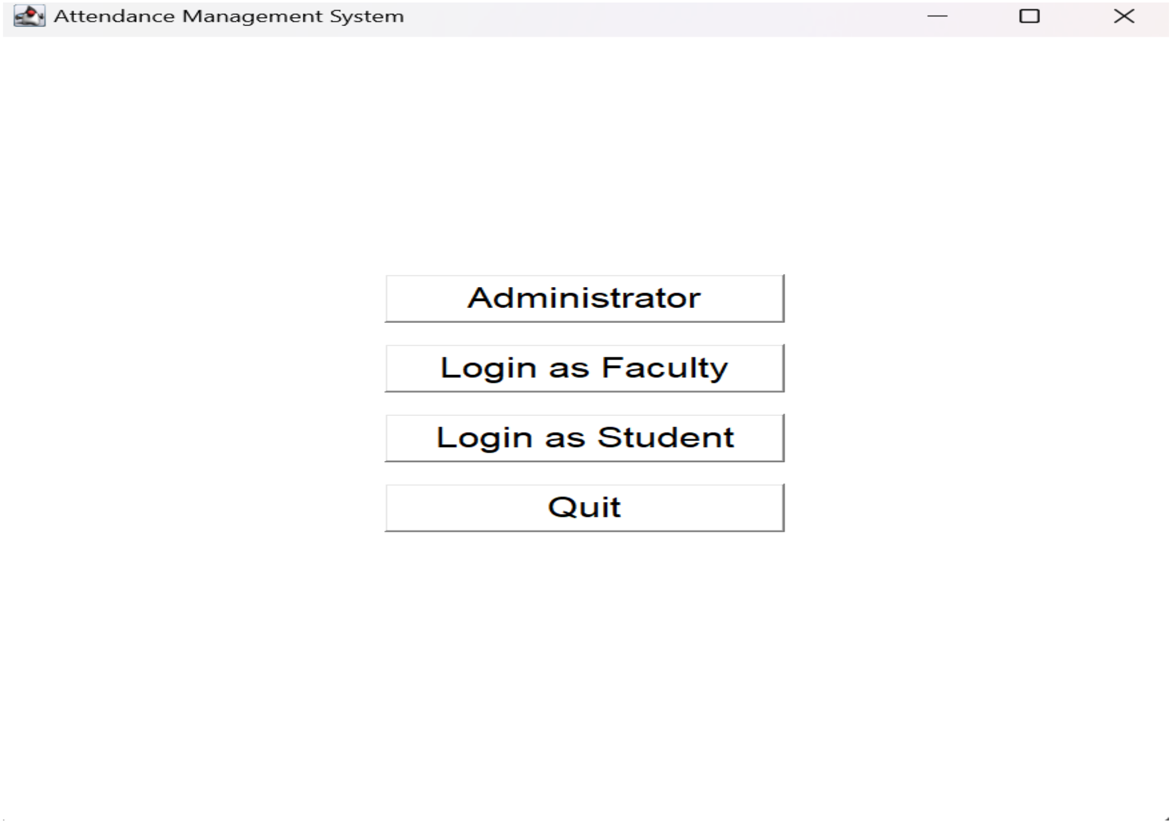
DATA BASE OF STUDENT MEMBERS :-



My Table

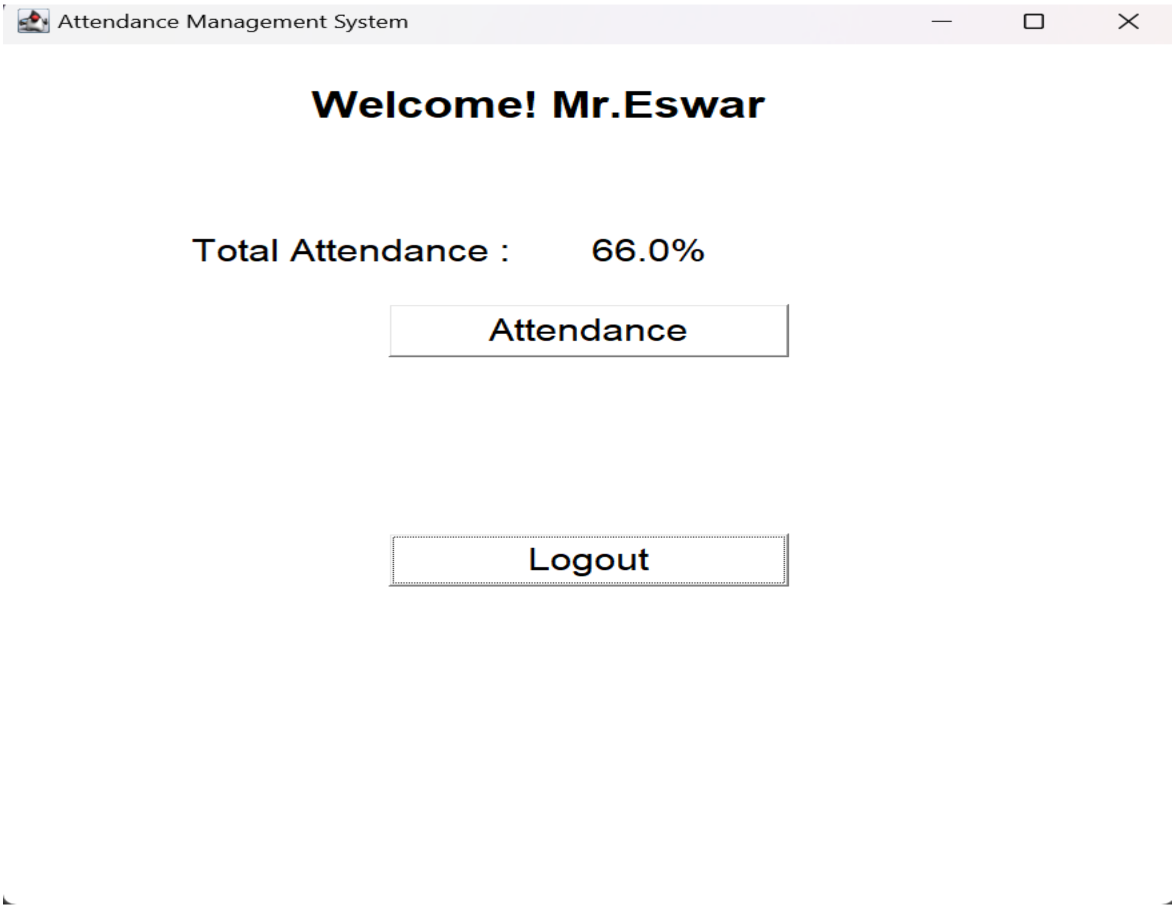
Name	Registration No	Class	Section	Roll No.	Parent Phone Num...	Address
Leeladhar	1000	X	A	69	6281877229	AP
Eswar	1001	X	A	69	6304818639	AP

HOME :-



Attendance Management System

Student attendance viewing :-



Attendance Management System

Welcome! Mr.Eswar

Total Attendance : 66.0%

CONCLUSION :-

The Attendance Management System is a useful tool for managing and keeping track of student attendance in a school or institution. The system provides a user-friendly interface for teachers to mark attendance for their students and store the data in a database for future reference.

The system makes use of Java programming language and MySQL database to develop the application. It has several features such as the ability to add and remove students from a class, view attendance reports, and mark attendance for individual students.

Overall, the Attendance Management System is an efficient and effective solution for schools and institutions to manage student attendance and keep track of student performance. It can help teachers to monitor student attendance, identify any attendance-related issues, and take necessary action to improve student performance.

GITHUB ID’S :-

B.Leeladhar Reddy 12103455
Github id :- Leeladharreddyb

K.Lakshmieswar Reddy 12107096
Github id :- eswar@143

P.Karthik 12006339
Github id :- Pothinakarthik03