

A Project Report on
“EMPLOYEE MANAGEMENT SYSTEM”

*Submitted in partial fulfilment
for the award of the Degree in*

BACHELOR OF COMPUTER APPLICATION

By

“MD YASIR ZAFAR”

AJU/190367

**Under the guidance of
PROF.MR.AKASH BHAGAT**



ARKA JAIN UNIVERSITY

JAMSHEDPUR, JHARKHAND

**DEPARTMENT OF COMPUTER SCIENCE & INFORMATION
TECHNOLOGY**

2019-22

JAMSHEDPUR, JHARKHAND
DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled "EMPLOYEE MANAGEMENT SYSTEM", is bona-fide work of MD YASIR ZAFAR bearing Registration Number: AJU/190367 submitted in partial fulfilment of the requirement for the award of degree in BACHELOR OF COMPUTER APPLICATION from ARKA JAIN UNIVERSITY, JHARKHAND.


Internal Guide


Head of the Department

Date: 25/5/22



University Seal

ABSTRACT

Employee management system is an application-based system, having two applications developed, one for employers to manage employee details and another for employees to mark their attendance. Every organisation whether government or private uses an information system [2.] to store data of their staff. However, in India it is found that many small-scale industries use pen and paper to keep a record. However, there are many advanced technology systems available that can do this work but they all are costly for these low-level industries. This paper discusses making a system for solving problems for them at a cheaper cost. This system will mark attendance of each employee and calculate the salary of them at the end of month. It also calculates overtime and total working hours of each employee. As in small scale each company has their own holidays preference and variable week off for employees, so all this power is given to the employer to manage holidays and week days of each employee separately. It saves lots of time and has no error in pay calculation hence preventing clashes between HR Team and employees. So that both employer and employee can focus on their work to develop their company.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to several individuals and organization for supporting me throughout the completion of my project.

First, I wish to express my sincere gratitude to my mentor (AKASH BHAGAT) for his/her enthusiasm patience, insightful comments, helpful information, practical advices and unceasing ideas that have helped me tremendously at all times in my Project and writing of these thesis. His/her immense knowledge, profound experience and professional expertise in Backend has enabled me to complete this project successfully. Without his/her support and guidance, this project would not have been possible.

I am also thankful to our respected H.O.D and all faculty members for loving inspiration and timely guidance. I also wish to express my sincere thanks to the Department of Computer science & Information technology of ARKA JAIN UNIVERSITY for accepting this project.

Thanks for all your encouragement!

DECLARATION

I hereby declare that the Project entitled "EMS" done at ARKA JAIN UNIVERSITY has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

This project is done in partial fulfilment of the requirement for the award of degree of BACHELOR OF COMPUTER APPLICATION to be submitted as final semester project as part of our curriculum.

MD YASIR ZAFAR
26.05.22

Signature

(MD YASIR ZAFAR)

CONTENT

1. INTRODUCTION
2. OBJECTIVES
3. HARDWARE SOFTWARE REQUIREMENTS
4. PROBLEM DIFNITION
5. OVERVIEW
6. ER DIAGRAM
7. DFD (CONTEXT LEVEL AND FIRST LEVEL)
8. DATA BASE STRUCTURE
9. MODULE EXPECTED TO BE DEVELOPED
10. SYSTEM SECURITY
11. CODING
12. SCREENSHOTS
13. IMPLEMENTATION AND SYSTEM TESTING
14. FUTURE SCOPE
15. CONCLUSION

INTRODUCTION

The system to be developed i.e. **Employee Management System (EMS)** is a system for managing data of the employees in a company. Once developed, this system will keep track of all the employees. It will monitor the registration of new employees, daily attendance of the employees, and other handlings of the employee records. EMS comprises the personal details of an employee. EMS keeps the record of employees joining date, leaving date, assets provided. It manages salary of employee according to the attendance. So we have decided to investigate the use of an Employee Management System. This system would be used by the HR Department of the corporation. The purpose of this document is to analyse and elaborate on the high-level needs and features of the Employee Management System. It focuses on the capabilities and facilities provided by a company. Visual Basic is used as front end and Oracle is for back end. Various functions and structures are used to make a complete use of this language. This project is well versed with the programming. Without an employee management system, it's a tedious job for the Human Resource department of any company. The Software Requirements Specification captures all the requirements in a single document. The Employee Management System that is to be developed provides the information of the employees and many other facilities. Employee Management system will store all the details of the employees.

Objective

This project is developed as a part of a course curriculum but it also has great significance pertaining to knowledge. The project was developed keeping in mind its importance for an organization or company.

Once developed, this project can serve as a prototype for other similar such projects.

Developing such a system which has a market value would increase our confidence and will acquaint us with all the related aspects of software implementation in the market.

Once approved the project can be developed keeping in mind the following objectives: -

Improved service through greater access to accurate information.

Increased productivity and job satisfaction among staff members as the extra efforts is eliminated due to automation.

More economical and safer means of storing and keeping track of information.

Easier access to information like performance reports, shift details, etc, as well as more accurate and faster results from statistical analysis.

Reduces errors and eliminating the long and repetitive manual processing.

Greater accountability and transparency in operations.

Improved efficiency and effectiveness in administration and management as it has unprecedented access to real-time information.

More reliable security for sensitive and confidential information.

Appropriate knowledge-based action and intervention can now take place in a timelier manner.

Hardware/Software requirement

Specification

Hardware Requirement:

RAM : 2GB or above

Processor: i3 and above

HDD: 500 GB or above

Printer: HP Laser

Monitor: 18' LED

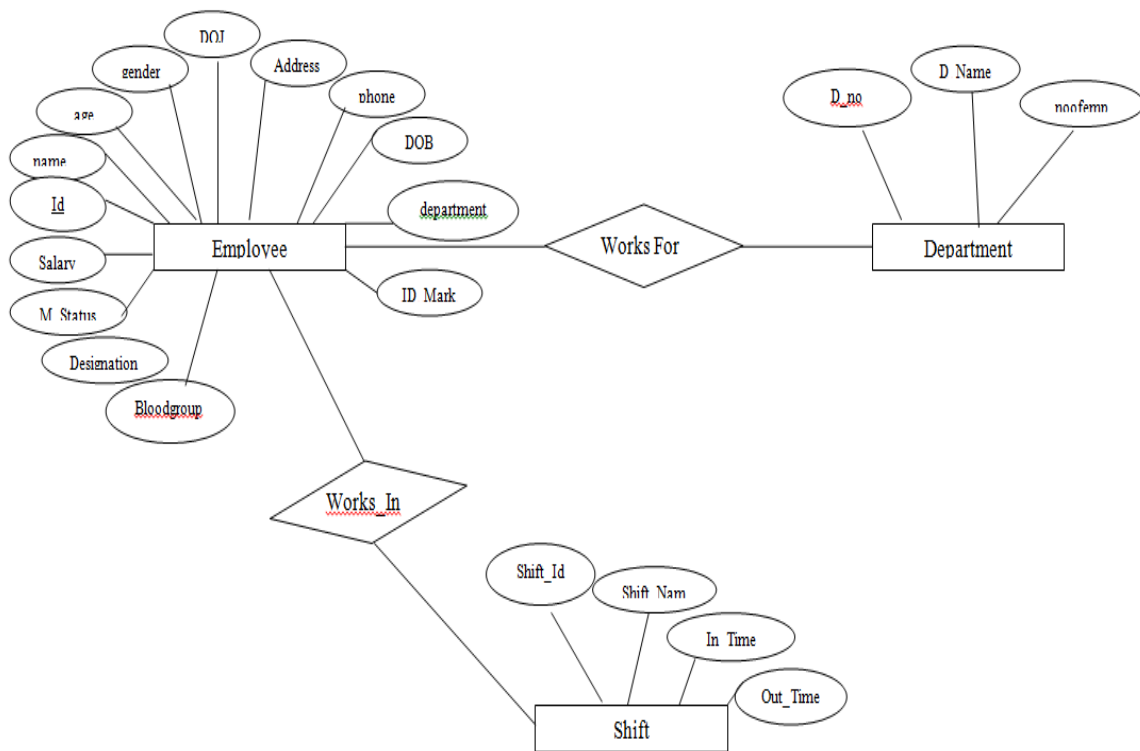
Software Requirement:

Front-End : JAVA (version17)

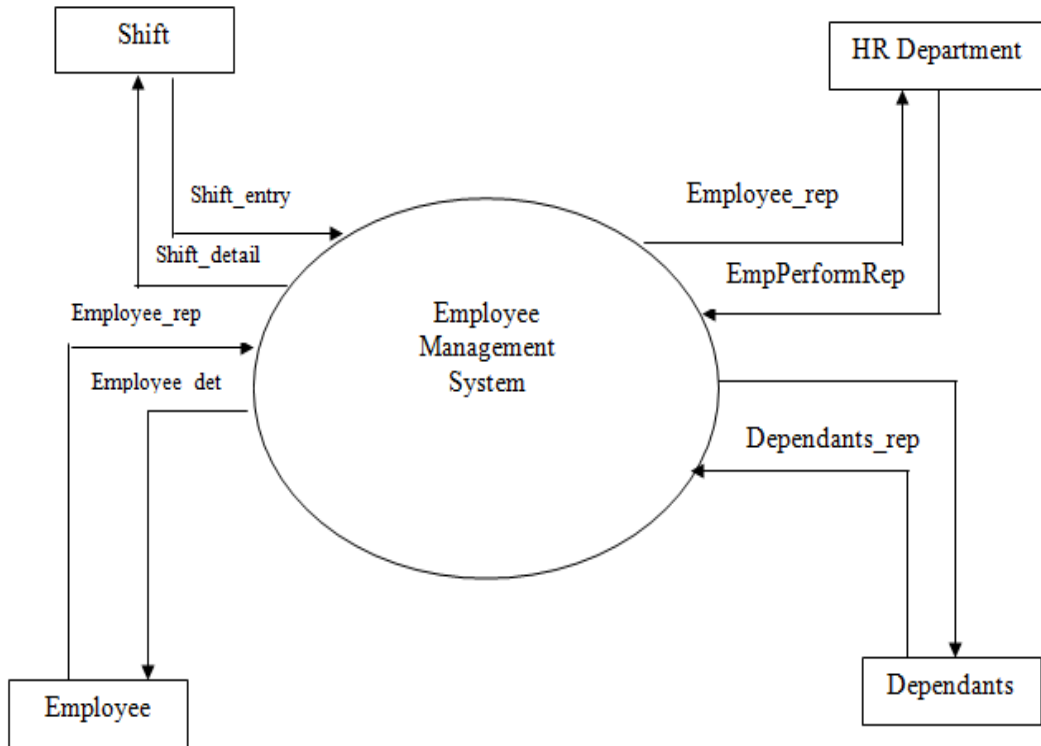
Back-End : MS ACCESS

Operating System: Windows7 or above

ER DIAGRAM



Context Diagram



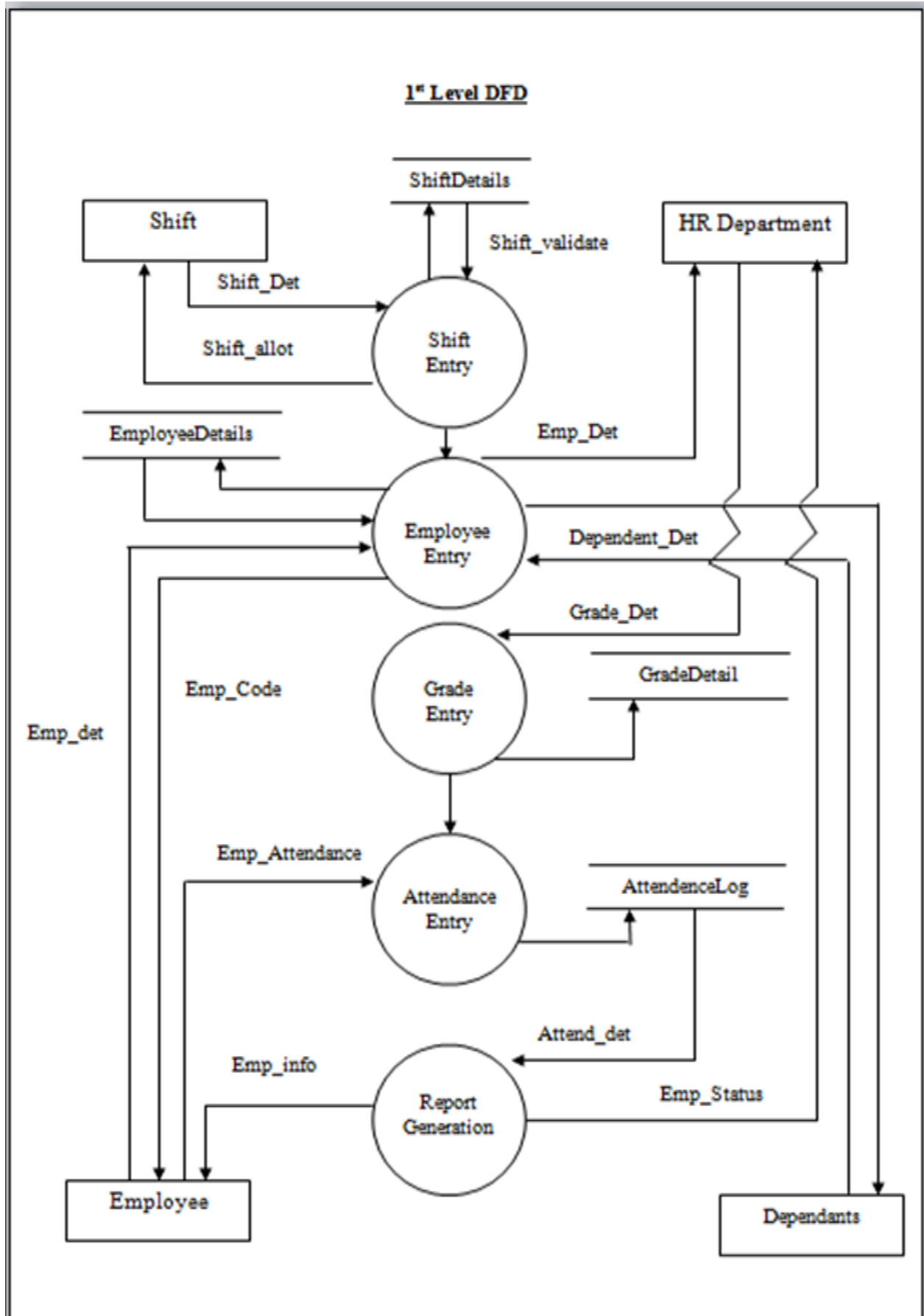
Employee: The system takes information from the employee and in-turn gives details to the employee as and when required.

1. Dependents: The details of the dependents are also maintained because in case of any claims and benefits to be given to them according to the norms of the company.
2. HR Department: This entity plays an integral role as it judge the employee on many grounds by the help of this system.
3. Shift: In big organizations/companies the work is done in shifts and the details of the same should be monitored properly.

•

- Details about the employee's qualifications are recorded which could be used as and when required.
 - Manages Employee's Fund Nominee
- Nominee records are kept which could be used in case of transfer of the funds or any mishap.

•



- Manages Employee Training Record
- It monitors the training record of the employees. This will help the employees to go for the training required and the employees who have done the training, based on the performance, they are given promotions and salary hike.
 - Allocates Gradation Number to Employee Grade such as Grade-A, Grade-B, Grade-C and Grade-D are allotted to the employees, which decides the job criteria and the allowances and deductions to be made as and when required.

It Demarcate the employee based on the grade. This helps in the salary calculation, bonus and incentives declarations, etc.

Database Structure

1. Attendance Table

Field Name	Type	Size
Antedate	Date	(8)
Emp Id	Varchar2	(10)
InTime	TimeStamp	(3)
OutTime	TimeStamp	(3)

2. Grade Detail Table

Field Name	Type	Field Size/Format
Grade_ID	Number	(2)
Grade_Name	Varchar2	(2)
HRA	Number	(10,2)
DA	Number	(10,2)
TA	Number	(10,2)
PF	Number	(10,2)
ESI	Number	(10,2)
LoanAmount	Number	(10,2)
PLEAVE	Number	(3)
SLEAVE	Number	(3)
CLEAVE	Number	(3)

3. EmpDetail Table

Field Name	Type	Size
Emp_Id	Varchar2	(10)
Emp_Name	Varchar2	(30)
Emp_FatherName	Varchar2	(30)
Emp_Address	Varchar2	(50)
Phone	Varchar2	(30)
Mobile	Varchar2	(30)
DOB	Date	(8)
Gender	Varchar2	(10)
BloodGroup	Varchar2	(10)
BasicSalary	Number	(10,2)
MaritalStatus	Varchar2	(15)
Nationality	Varchar2	(15)
ShiftID	Number	(3)
GradeID	Number	(2)
DOJ	Date	(8)
ID_Mark	Varchar2	(50)
Designation	Varchar2	(50)
Department	Varchar2	(50)
Status	Varchar2	(10)

4. HolidayDetail Table

Field Name	Type	Size
HolidayDate	Date	(8)
Description	Varchar2	(20)

5. LeaveDetail Table

Field	Type	Size
Emp_ID	Varchar2	(10)
LeaveType	Varchar2	(20)
LeaveDate	Date	(8)
Description	Varchar2	(50)

6. Salary Table

Field	Type	Size
Salary_ID	Number	(3)
SalaryMonth	Date	(8)

7. ShiftDetails Table

Field	Type	Size
Shift_ID	Number	(3)
Shift_Name	Varchar2	(5)
InTime	TimeStamp	(3)
OutTime	TimeStamp	(3)

8. SalaryDetails Table

Field	Type	Size
Salary_ID	Number	(10,2)
Emp_ID	Varchar2	(10)
WorkingDay	Number	(2)
WorkedDay	Number	(2)
BasicSalary	Number	(10,2)
HRA	Number	(10,2)
DA	Number	(10,2)
TA	Number	(10,2)
PF	Number	(10,2)
ESI	Number	(10,2)
LoanAmount	Number	(10,2)

Modules expected to be developed

Employee Master

This module has the facility of adding, editing, searching of records of the employees. The module helps to maintain the employee records and store it in the employee table.

Shift Master

This module is the shift master and it is used to handle master records of the shift. New shifts can be declared and even timing can be altered as per requirement. All such tasks will be done through this module.

Grade Master

Grade master module is used for grade handling. If any grade is added in the company, it is added in the system through this module. The facilities such as HRA, DA, PF, etc can be modified for all the employees from this module.

Daily Attendance

This module is used for attendance monitoring. This module should keep track of the employees coming in and going out of the office/company's premises. An employee may go out and come in the organization, hence his/her presence should be monitored with proper time

Leave Entry

This module is used for monitoring leave entry of the employee.

Holiday Entry

Holiday entry module is also required to be there in the system. This will help in intimation purpose and to manage the calendar of the company.

Monthly Salary

This is used to generate the monthly salary of the employee as per the grade of the employee which includes HRA, DA, TA, PF, etc. being specified.

Reports to be generated for the proposed system

- 'Employee List
- It will display the employee list report which is used for the reference purpose as and when required by the HR department.

- Grade List
- This report will display the grade list of the employees.

- Holiday List
- It will display the holidays falling in the month and even the holidays declared by the company/organization. It can be printed and displayed for the notification purpose.

- Shift List
- The work being done in the company is on shift basis. This report displays the shift details which will be beneficial for the organization.

- Employee-wise leave list
- This report once generated will be used to make a check on the employees as to who has taken how many leaves, whose leaves are to be sanctioned and so on.

- Today's Absentees list
- This report of the proposed system will display the list of the employees not present in the company on the current day.

- Dependents detail list
- This report gives details of the dependents of the employee. In case of claims and settlement this report can be used for verification purpose.

System Security

The system to be developed is also studied on the security aspects. The operator of the system is authenticated and no trespasser can access the system. Even during the entry level the critical operation which can cause damage, such as deleting the records of the employee, editing data of the dependents, etc are allowed only for the administrator and the operator can be given only to enter the transaction level entries. Master handling is reserved for the administrator. This feature if implemented in the system can increase the security of the system.

Only validated entries from the liable operator of the system are stored by the system.

CODING

```
import javax.swing.*;
import java.sql.*;
import java.util.*;
import java.io.*;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class splash extends Frame implements ActionListener
{
    JLabel lblbkg;
    JLabel lblbkg1;

    JButton Btnnext;

    public splash()
    {
        Dimension screen =Toolkit.getDefaultToolkit().getScreenSize();
        setLocation((screen.width - 500)/2,((screen.height-350)/2));
        setVisible(true);
        setSize(420,340);
        setTitle("An Enterprise Solution...");
        setLayout(null);
        lblbkg = new JLabel(new Image Icon("images/pic3.jpg"));
        lblbkg1 = new JLabel("Developed By: Shivam Kumar");

        Bonnet = new Button("NEXT", new Image Icon("images/2rightarrow.png"));

        Btnnext.setBounds(100,260,120,40);
        lblbkg.setBounds(0,0,420,300);
```



```

lblbkg1.setBounds(120,100,300,300);
add(Btnnext);
add(lblbkg);
add(lblbkg1);
Btnnext.addActionListener(this);
repaint();
}

```

```

public static void main(String ar[]) throws IOException
{
splash obj=new splash();
}

```

```

public void actionPerformed(ActionEvent e)
{
if(e.getSource()==Btnnext)
{
this.dispose();
loginform obj=new loginform();
}
}

```

```

}import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
import java.io.*;
import java.sql.*;

```

```

public class payslip extends Frame implements ActionListener
{
JButton BtnNew=new JButton("New",new ImageIcon("images/new.gif"));
JButton BtnSearch=new JButton("Search",new ImageIcon("images/search.png"));
JButton BtnCancel=new JButton("Cancel");
JLabel LblPayMonth=new JLabel("Pay Slip For Month:");
JComboBox monthname =new JComboBox();

```

```
String mname[]={ "January", "February", "March",  
"April", "May", "June", "July", "August", "September", "october", "November", "December"};  
JLabel LblEmp_Name=new JLabel("Employee Name:");  
JLabel LblEmp_Code=new JLabel("Employee Code:");  
JLabel LblEmp_Type=new JLabel("Employee Type:");  
JLabel LblEmp_Desig=new JLabel("Employee Designation");  
JLabel LblBasic_Pay=new JLabel("Basic Salary:");  
JLabel LblAllowance=new JLabel("Allowances...");  
JLabel LblDeduction=new JLabel("Deductions...");  
JLabel LblDA=new JLabel("DA:");  
JLabel LblHRA=new JLabel("HRA:");  
JLabel LblWA=new JLabel("WA:");  
JLabel LblGPF=new JLabel("GPF");  
JLabel LblIT=new JLabel("IT");  
JLabel LblGIS=new JLabel("GIS");  
JLabel LblPF=new JLabel("PF");  
JLabel LblLIC=new JLabel("LIC");  
JLabel LblTot_Allowance=new JLabel("Total Allowances:");  
JLabel LblTot_Deduction=new JLabel("Total Deduction:");  
JLabel LblNet_Salary=new JLabel("Net Salary:");  
JTextField TxtEmp_Name=new JTextField();  
JTextField TxtEmp_Code=new JTextField();  
JComboBox CmbEmp_Type=new JComboBox();  
JTextField TxtEmp_Desig=new JTextField();  
JTextField TxtBasic_Pay=new JTextField();  
JTextField TxtDA=new JTextField();  
JTextField TxtHRA=new JTextField();  
JTextField TxtWA=new JTextField();  
JTextField TxtGPF=new JTextField();  
JTextField TxtIT=new JTextField();  
JTextField TxtGIS=new JTextField();  
JTextField TxtPF=new JTextField();  
JTextField TxtLIC=new JTextField();  
JTextField TxtTot_Allowance=new JTextField();
```

```

JTextField TxtTot_Deduction=new JTextField();
JTextField TxtNet_Salary=new JTextField();
JTextField TxtSalary_Month=new JTextField();
JTextField TxtDate=new JTextField();
JTextField TxtCasual=new JTextField();
JTextField TxtSick=new JTextField();
JTextField TxtMedical=new JTextField();
JTextField TxtAnnual=new JTextField();
public payslip()
{
setLayout(null);
Dimension screen=Toolkit.getDefaultToolkit().getScreenSize();
setTitle("Employee Details");
setSize(screen.width,screen.height);
setVisible(true);
////////////////////////////////////
BtnNew.setBounds(220,640,100,40);
BtnSearch.setBounds(450,640,100,40);
BtnCancel.setBounds(670,640,100,40);
LblPayMonth.setBounds(100,100,150,30);
monthname.setBounds(270,100,150,30);
JLabel LblCasual=new JLabel("Casual Leave");
JLabel LblSick=new JLabel("Sick Leave");
JLabel LblMedical=new JLabel("Medical Leave");
JLabel LblAnnual=new JLabel("Annual Leave");
LblEmp_Type.setBounds(100,150,150,30);
LblEmp_Name.setBounds(100,200,150,30);
LblEmp_Code.setBounds(490,150,150,30);
LblEmp_Desi.setBounds(490,210,150,30);
LblBasic_Pay.setBounds(100,250,150,30);
LblAllowance.setBounds(100,300,150,30);
LblDeduction.setBounds(370,300,150,30);
LblIDA.setBounds(100,340,150,30);
LblHRA.setBounds(100,380,150,30);

```

LblWA.setBounds(100,420,150,30);
LblGPF.setBounds(370,340,150,30);
LblIT.setBounds(370,380,150,30);
LblGIS.setBounds(370,420,150,30);
LblPF.setBounds(370,460,150,30);
LblLIC.setBounds(370,500,150,30);
LblTot_Allowance.setBounds(100,540,100,30);
LblTot_Deduction.setBounds(370,540,100,30);
LblNet_Salary.setBounds(100,580,150,30);
CmbEmp_Type.setBounds(270,150,150,30);
TxtEmp_Name.setBounds(270,200,150,30);
TxtEmp_Code.setBounds(660,150,150,30);
TxtEmp_Desig.setBounds(660,210,150,30);
TxtBasic_Pay.setBounds(270,250,150,30);
TxtDA.setBounds(180,340,150,30);
TxtHRA.setBounds(180,380,150,30);
TxtWA.setBounds(180,420,150,30);
TxtGPF.setBounds(470,340,150,30);
TxtIT.setBounds(470,380,150,30);
TxtGIS.setBounds(470,420,150,30);
TxtPF.setBounds(470,460,150,30);
TxtLIC.setBounds(470,500,150,30);
TxtTot_Allowance.setBounds(210,540,150,30);
TxtTot_Deduction.setBounds(470,540,150,30);
TxtNet_Salary.setBounds(210,580,150,30);
LblCasual.setBounds(700,340,100,30);
LblSick.setBounds(700,380,100,30);
LblMedical.setBounds(700,420,100,30);
LblAnnual.setBounds(700,460,100,30);
TxtCasual.setBounds(820,340,100,30);
TxtSick.setBounds(820,380,100,30);
TxtMedical.setBounds(820,420,100,30);
TxtAnnual.setBounds(820,460,100,30);
////////////////////////////////////

```
for(int i=0;i<12;i++)
monthname.addItem(mname[i]);
add(BtnNew);
add(BtnSearch);
add(BtnCancel);
add(LblPayMonth);
add(monthname);
add(LblEmp_Type);
add(LblEmp_Name);
add(LblEmp_Code);
add(LblEmp_Desi);
add(LblBasic_Pay);
add(LblAllowance);
add(LblDeduction);
add(LblDA);
add(LblHRA);
add(LblWA);
add(LblGPF);
add(LblIT);
add(LblGIS);
add(LblPF);
add(LblLIC);
add(LblTot_Allowance);
add(LblTot_Deduction);
add(LblNet_Salary);
add(TxtEmp_Name);
add( TxtEmp_Code);
add( TxtEmp_Desi);
add( TxtBasic_Pay);
add( TxtDA);
add( TxtHRA);
add( TxtWA);
add( TxtGPF);
add( TxtIT);
```

```

add( TxtGIS);
add( TxtPF);
add( TxtLIC);
add(TxtTot_Allowance);
add( TxtTot_Deduction);
add( TxtNet_Salary);
CmbEmp_Type.addItem("Programmer");
CmbEmp_Type.addItem("Analyst");
CmbEmp_Type.addItem("Designer");
CmbEmp_Type.addItem("Team Leader");
add(CmbEmp_Type);
add(LblCasual);
add(LblSick);
add(LblMedical);
add(LblAnnual);
add(TxtCasual);
add(TxtSick);
add(TxtMedic);
add(TxtAnnual);
BtnNew.addActionListener(this);
BtnCancel.addActionListener(this);
BtnSearch.addActionListener(this);

repaint();
////////////////////////////////////
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==BtnNew)
{
TxtEmp_Name.setText("");
TxtEmp_Code.setText("");
TxtEmp_Desig.setText("");
TxtBasic_Pay.setText("");

```

```

TxtDA.setText("");
TxtHRA.setText("");
TxtWA.setText("");
TxtGPF.setText("");
TxtIT.setText("");
TxtGIS.setText("");
TxtPF.setText("");
TxtLIC.setText("");
TxtTot_Allowance.setText("");
TxtTot_Deduction.setText("");
TxtNet_Salary.setText("");
TxtSick.setText("");
TxtMedical.setText("");
TxtAnnual.setText("");
TxtCasual.setText("");
}
if(e.getSource()==BtnSearch)
{
try
{
int flg=0;
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String tempcode=JOptionPane.showInputDialog(null,"Enter Employee Code:");
ResultSet res=st.executeQuery("Select * from employeetbl where
Emp_Code='"+tempcode+"'");
while(res.next())
{
TxtEmp_Code.setText(tempcode);
CmbEmp_Type.setSelectedItem(res.getString(2));
TxtEmp_Name.setText(res.getString(3));
TxtEmp_Desi.setText(res.getString(4));
TxtBasic_Pay.setText(res.getString(5));

```

```

flg=1;
}
if(flg==0)
OptionPane.showMessageDialog(null,"No such Record present");
else
{
ResultSet res1=st.executeQuery("Select * from remunerationtbl where
Emp_Type='"+CmbEmp_Type.getSelectedItemAt()+"'");
while(res1.next())
{
double tbasic=Double.parseDouble(TxtBasic_Pay.getText());
double thra=Double.parseDouble(res1.getString(2))*tbasic/100;
double tda=Double.parseDouble(res1.getString(3))*tbasic/100;
double twa=Double.parseDouble(res1.getString(4))*tbasic/100;
double tgpf=Double.parseDouble(res1.getString(5))*tbasic/100;
double tit=Double.parseDouble(res1.getString(6))*tbasic/100;
double tgis=Double.parseDouble(res1.getString(7))*tbasic/100;
double tpf=Double.parseDouble(res1.getString(8))*tbasic/100;
double tlic=Double.parseDouble(res1.getString(9))*tbasic/100;
TxtHRA.setText(""+thra);
TxtDA.setText(""+tda);
TxtWA.setText(""+twa);
TxtGPF.setText(""+tgpf);
TxtIT.setText(""+tit);
TxtGIS.setText(""+tgis);
TxtPF.setText(""+tpf);
TxtLIC.setText(""+tlic);
TxtTot_Allowance.setText(""+(thra+tda+twa));
TxtTot_Deduction.setText(""+(tgpf+tit+tgis+tpf+tlic));
TxtNet_Salary.setText(""+(tbasic+(thra+tda+twa)-(tgpf+tit+tgis+tpf+tlic)));
}
ResultSet res2=st.executeQuery("Select sum(Noofdays) from leavetbl where
Emp_Code='"+TxtEmp_Code.getText()+"' and Leave_Code='Casual Leave'");
while(res2.next())

```



```

{
TxtCasual.setText(res2.getString(1));
}
res2=st.executeQuery("Select sum(Noofdays) from leavetbl where
Emp_Code='"+TxtEmp_Code.getText()+"' and Leave_Code='Sick Leave'");
while(res2.next())
{
TxtSick.setText(res2.getString(1));
}
res2=st.executeQuery("Select sum(Noofdays) from leavetbl where
Emp_Code='"+TxtEmp_Code.getText()+"' and Leave_Code='Medical Leave'");
while(res2.next())
{
TxtMedical.setText(res2.getString(1));
}
res2=st.executeQuery("Select sum(Noofdays) from leavetbl where
Emp_Code='"+TxtEmp_Code.getText()+"' and Leave_Code='Annual Leave'");
while(res2.next())
{
TxtAnnual.setText(res2.getString(1));
}
}
}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==BtnCancel)
{
this.dispose();
}
}
public static void main(String argv[])throws IOException

```

```
{
payslip obj=new payslip();
}
```

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.awt.*;
import javax.swing.*;

public class menudemo extends JFrame implements ActionListener
{
JMenuBar mbar = new JMenuBar();
JMenu mnumast=new JMenu("Master");
JMenu mnutrans = new JMenu("Transaction");
JMenu mnurep= new JMenu("Report");
JMenu mnuexit = new JMenu("Exit");
JMenuItem meemp=new JMenuItem("Employee",new
ImageIcon("images/employee.png"));
JMenuItem merem=new JMenuItem("Remuneration",new ImageIcon("images/r.png"));
JMenuItem meatt=new JMenuItem("Attendance",new
ImageIcon("images/attendance1.png"));
JMenuItem meleave=new JMenuItem("Leave",new ImageIcon("images/leave.png"));
JMenuItem meholidays=new JMenuItem("Holidays",new
ImageIcon("images/holidays.png"));
JMenuItem mepay=new JMenuItem("PaySlip",new ImageIcon("images/payslip.png"));
JMenuItem meabsent=new JMenuItem("Absent Record");

JMenuItem meexit=new JMenuItem("exit");
JLabel lblbk;

public menudemo()
{
Dimension screen=Toolkit.getDefaultToolkit().getScreenSize();
setLayout(new BorderLayout());
```

```

setSize(screen.width,screen.height);
mnumast.add(meemp);
mnumast.add(merem);
setBackground(new Color(255,0,0));
lblbk=new JLabel(new ImageIcon("images/pic4.jpg"));
mnutrans.add(meatt);
mnutrans.add(meleave);
mnutrans.add(meholidays);

mnurep.add(mepay);
mnurep.add(meabsent);

mnuexit.add(meexit);

mbar.add(mnumast);
mbar.add(mnutrans);
mbar.add(mnurep);
mbar.add(mnuexit);

meemp.addActionListener(this);
merem.addActionListener(this);
meatt.addActionListener(this);
meleave.addActionListener(this);
meholidays.addActionListener(this);
mepay.addActionListener(this);
meabsent.addActionListener(this);
meexit.addActionListener(this);

setJMenuBar(mbar);
setVisible(true);
add(lblbk);
}

public void actionPerformed(ActionEvent e)

```

```

{
if(e.getSource()==meexit)
{
System.exit(0);
}
if(e.getSource()==meemp)
{
empdata obj=new empdata();
}
if(e.getSource()==merem)
{
empSetting obj=new empSetting();
}
if(e.getSource()==meatt)
{
Attendance obj=new Attendance();
}
if(e.getSource()==meleave)
{
leave obj=new leave();
}
if(e.getSource()==meholidays)
{
holidays obj=new holidays();
}
if(e.getSource()==mepay)
{
payslip obj=new payslip();
}
if(e.getSource()==meabsent)
{
absent obj=new absent();
}
}
}

```

```

public static void main(String arg[]) throws IOException
{
menudemo obj=new menudemo();
}import javax.swing.*;
import java.sql.*;
import java.util.*;
import java.io.*;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class loginform extends Frame implements ActionListener
{
// JLabel lblbkg;
JLabel lblusr=new JLabel("user name");
JLabel lblpwd=new JLabel("password");
JTextField txtusr=new JTextField(15);
JPasswordField txtpwd=new JPasswordField(15);
JButton Btnlogin=new JButton("login");
JButton Btnexit=new JButton("exit");
JLabel limg;
public loginform()
{
Dimension screen =Toolkit.getDefaultToolkit().getScreenSize();
setLocation((screen.width - 800)/2,((screen.height-350)/2));
setVisible(true);
setSize(620,300);
setTitle("loginform");
setLayout(null);
limg=new JLabel(new ImageIcon("D:/EmployeeManageSystem/images/loo111.jpg"));
// lblbkg = new JLabel(new ImageIcon("images/background2.jpg"));
Btnlogin = new JButton("Login", new ImageIcon("images/key.gif"));
Btnexit = new JButton("Exit", new ImageIcon("images/Keys.gif"));
lblusr.setBounds(150,80,150,40);
lblpwd.setBounds(150,130,150,40);

```

```

txtusr.setBounds(300,80,180,40);
txtpwd.setBounds(300,130,180,40);
Btnlogin.setBounds(300,220,120,40);
Btnexit.setBounds(440,220,120,40);
// lblbkg.setBounds(0,0,300,300);
limg.setBounds(-5,5,620,this.getHeight());
lblusr.setForeground(new Color(255,255,255));
lblpwd.setForeground(new Color(255,255,255));
add(lblusr);
add(lblpwd);
add(txtusr);
add(txtpwd);
add(Btnlogin);
add(Btnexit);
//add(lblbkg);
add(limg);
Btnlogin.addActionListener(this);
Btnexit.addActionListener(this);
repaint();
}

```

```

public static void main(String ar[]) throws IOException
{
loginform obj=new loginform();
}

```

```

public void actionPerformed(ActionEvent e)
{
if(e.getSource()==Btnlogin)
{
try
{
int flg=0;
Connection con;

```

```

con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
ResultSet res;
Statement st=con.createStatement();
res=st.executeQuery("select * from logintbl");
String userstr=txtusr.getText();
String paswstr=txtpwd.getText();

while(res.next())
{
if((userstr.equals(res.getString(1))) && (paswstr.equals(res.getString(2))))
{
flg=1;
this.dispose();
menudemo obj=new menudemo();
}
}
if(flg==0)
{
JOptionPane.showMessageDialog(null,"Invalid Login");
txtusr.setText("");
txtpwd.setText("");
}
}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==Btnexit)
{
this.dispose();
}
}
}

```

```

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

public class leave extends Frame implements ActionListener
{
    JButton BtnNew=new JButton("New",new ImageIcon("images/new.gif"));
    JButton BtnSave=new JButton("Save",new ImageIcon("images/save.gif"));
    JButton BtnSearch=new JButton("Search",new ImageIcon("images/search.png"));
    JButton BtnEdit=new JButton("Edit",new ImageIcon("images/edit.png"));
    JButton BtnCancel=new JButton("Cancel");
    JLabel LblEmp_Name=new JLabel("Employee Name:");
    JLabel LblEmp_Code=new JLabel("Employee Code:");
    JLabel LblEmp_Type=new JLabel("Employee Type:");
    JLabel LblEmp_Desig=new JLabel("Employee Designation");
    JLabel LblLeave_Code=new JLabel("Leave Code:");
    JLabel LblStart_Date=new JLabel("Start Date:");
    JLabel LblEnd_Date=new JLabel("End Date:");
    JLabel LblNo_Days=new JLabel("Number Of Days");
    JLabel LblReference_No=new JLabel("Reference Number");
    JLabel LblReason=new JLabel("Reason of leave:");
    JTextField TxtEmp_Name=new JTextField();
    JTextField TxtEmp_Code=new JTextField();
    JTextField TxtEmp_Type=new JTextField();
    JTextField TxtEmp_Desig=new JTextField();
    JComboBox CmbLeave_Code=new JComboBox();
    JTextField TxtStart_Date=new JTextField();
    JTextField TxtEnd_Date=new JTextField();
    JTextField TxtNo_Days=new JTextField();
    JTextField TxtReference_No=new JTextField();
    JTextField TxtReason=new JTextField();
    public leave()

```



```

{
setLayout(null);
Dimension screen=Toolkit.getDefaultToolkit().getScreenSize();
setTitle("Employee Leave Application Form...");
setSize(screen.width,screen.height);
setVisible(true);
////////////////////////////////////
BtnNew.setBounds(220,640,100,40);
BtnSave.setBounds(340,640,100,40);
BtnSearch.setBounds(450,640,100,40);
BtnEdit.setBounds(560,640,100,40);
BtnCancel.setBounds(670,640,100,40);
LblEmp_Code.setBounds(100,100,150,30);
LblEmp_Name.setBounds(100,150,150,30);
LblEmp_Type.setBounds(490,100,150,30);
LblEmp_Desi.setBounds(490,160,150,30);
LblLeave_Code.setBounds(100,200,150,30);
LblStart_Date.setBounds(100,250,150,30);
LblEnd_Date.setBounds(100,300,150,30);
LblNo_Days.setBounds(100,350,150,30);
LblReason.setBounds(100,390,150,30);
LblReference_No.setBounds(100,430,150,30);
TxtEmp_Type.setBounds(660,100,150,30);
TxtEmp_Name.setBounds(270,150,150,30);
TxtEmp_Code.setBounds(270,100,150,30);
TxtEmp_Desi.setBounds(660,160,150,30);
CmbLeave_Code.setBounds(270,200,150,30);
TxtStart_Date.setBounds(270,250,150,30);
TxtEnd_Date.setBounds(270,300,150,30);
TxtNo_Days.setBounds(270,350,150,30);
TxtReference_No.setBounds(270,430,150,30);
TxtReason.setBounds(270,390,150,30);
////////////////////////////////////
add(BtnNew);

```

```
add(BtnSave);
add(BtnSearch);
add(BtnEdit);
add(BtnCancel);
add(LblEmp_Type);
add(LblEmp_Name);
add(LblEmp_Code);
add(LblLeave_Code);
add(LblStart_Date);
add(LblEnd_Date);
add(LblNo_Days);
add(LblEmp_Desi);
add(LblReference_No);
add(LblReason);
add(TxtEmp_Name);
add(TxtEmp_Code);
add(TxtEmp_Desi);
add(TxtReason);
CmbLeave_Code.addItem("Casual Leave");
CmbLeave_Code.addItem("Sick Leave");
CmbLeave_Code.addItem("Medical Leave");
CmbLeave_Code.addItem("Annual Leave");
add(CmbLeave_Code);
add(TxtStart_Date);
add(TxtEnd_Date);
add(TxtNo_Days);
add(TxtReference_No);
add(TxtEmp_Type);
BtnNew.addActionListener(this);
BtnEdit.addActionListener(this);
BtnCancel.addActionListener(this);
BtnSave.addActionListener(this);
BtnSearch.addActionListener(this);
repaint();
```

```

////////////////////////////////////
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==BtnNew)
{
TxtEmp_Type.setText("");
TxtEmp_Name.setText("");
TxtEmp_Code.setText("");
TxtEmp_Desig.setText("");
TxtStart_Date.setText("");
TxtEnd_Date.setText("");
TxtNo_Days.setText("");
TxtReference_No.setText("");
TxtReason.setText("");
try
{
int flg=0;
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
ResultSet res=st.executeQuery("Select Reference_no from keytbl");
while(res.next())
{
TxtReference_No.setText(""+res.getString(1));
}
}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==BtnSave)
{

```

```

try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="insert into
leavetbl(Emp_Code,Emp_Name,Start_Date,End_Date,Noofdays,Reason,Leave_Code,Ref
erence_No)values(?,?,?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
ps.setString(1,TxtEmp_Code.getText());
ps.setString(2,TxtEmp_Name.getText());
ps.setString(3,TxtStart_Date.getText());
ps.setString(4,TxtEnd_Date.getText());
ps.setString(5,TxtNo_Days.getText());
ps.setString(6,TxtReason.getText());
ps.setString(7,""+CmbLeave_Code.getSelectedItem());
ps.setString(8,TxtReference_No.getText());
ps.executeUpdate();
JOptionPane.showMessageDialog(null,"Saved Successfully");
str="update keytbl set Reference_no=Reference_no+1";
st.executeUpdate(str);

}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==BtnSearch)
{
try
{
int flg=0;
Connection con;

```

```

con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String tempcode=JOptionPane.showInputDialog(null,"Enter Employee Code:");
ResultSet res=st.executeQuery("Select * from employeetbl where
Emp_Code='"+tempcode+"'");
while(res.next())
{
TxtEmp_Code.setText(res.getString(1));
TxtEmp_Type.setText(res.getString(2));
TxtEmp_Name.setText(res.getString(3));
TxtEmp_Desig.setText(res.getString(4));
flg=1;
}
if(flg==0)
JOptionPane.showMessageDialog(null,"No such Record present");
}

catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==BtnEdit)
{
}
if(e.getSource()==BtnCancel)
{
this.dispose();
}
}

public static void main(String args[])throws IOException
{
leave obj=new leave();
}

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.io.*;
import java.util.*;
import java.text.DateFormat;
import java.util.Date;
import java.util.*;
import javax.swing.table.DefaultTableModel;
class holidays extends JFrame implements ActionListener
{
String tempreg;
int tslno;
JTable tableholiday;
JScrollPane sp=new JScrollPane();
String [] col={"Sl.No.", "Holiday Name", "day", "month", "year"};
Object[][] data={{ tslno,tempreg }};
DefaultTableModel model=new DefaultTableModel(col,0);
JLabel lblholiday=new JLabel("Holiday Description:");
JTextField txtholiday=new JTextField();
JLabel lblmonth=new JLabel("Month");
JLabel lblday=new JLabel("Day");
JLabel lblyear=new JLabel("Year");
JButton b=new JButton("Add");
JButton b1=new JButton("Submit");
JButton b2=new JButton("Cancel");
JComboBox monthname =new JComboBox();
String mname[]={ "January", "February", "March",
"April", "May", "June", "July", "August", "September", "october", "November", "December"};
JComboBox dayname=new JComboBox();
JComboBox year=new JComboBox();
public holidays()
{

```

```

int i;
Dimension screen=Toolkit.getDefaultToolkit().getScreenSize();
setSize(screen.width,screen.height);
setLayout(null);
setVisible(true);
setTitle("Monthly Holidays Entry...");
lblholiday.setBounds(390,50,150,40);
txtholiday.setBounds(560,50,150,40);
lblday.setBounds(50,50,150,40);
dayname.setBounds(220,50,150,40);
lblmonth.setBounds(50,110,150,40);
monthname.setBounds(220,110,150,40);
lblyear.setBounds(50,170,150,40);
year.setBounds(220,170,150,40);
b.setBounds(560,110,100,40);
b1.setBounds(560,160,100,40);
b2.setBounds(560,210,100,40);
////////////////////Holiday Table////////////////////
tableholiday=new JTable(data,col);
sp=new JScrollPane(tableholiday);
tableholiday.setModel(model);
sp.setBounds(100,250,300,400);
tableholiday.setBounds(100,250,300,400);

tableholiday.setFillViewportHeight(true);
////////////////////////////////////

for(i=2000;i<2020;i++)
year.addItem(i);
for(i=0;i<12;i++)
monthname.addItem(mname[i]);
for(i=1;i<=31;i++)
dayname.addItem(""+i);

```

```

add(lblmonth);add(lblday);add(lblyear);add(b);add(b1);add(b2);add(monthname);add(day
name);add(year);
add(lblholiday);add(txtholiday);
add(sp);
Calendar cal=Calendar.getInstance();
monthname.setSelectedItem(mname[cal.get(Calendar.MONTH)]);
year.setSelectedItem(cal.get(Calendar.YEAR));
b.addActionListener(this);
b1.addActionListener(this);
b2.addActionListener(this);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
repaint();
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b)
{
Vector v=new Vector();
v.add(""+(tableholiday.getRowCount()+1));
v.add(txtholiday.getText());
v.add(dayname.getSelectedItemAt());
v.add(monthname.getSelectedItemAt());
v.add(year.getSelectedItemAt());
model.addRow(v);
}
if(e.getSource()==b2)
{
this.dispose();
}
if(e.getSource()==b1)
{
try
{
Connection con;

```



```

con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="insert into Holidaytbl(HolidayName,dday,mmonth,yyear)values(?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
for(int i=0;i<tableholiday.getRowCount();i++)
{
String tempcode=(String)tableholiday.getValueAt(i,1);
ps.setString(1,tempcode);
ps.setInt(2,Integer.parseInt(""+(String)tableholiday.getValueAt(i,2)));
ps.setString(3,(String)tableholiday.getValueAt(i,3));
ps.setInt(4,Integer.parseInt(""+tableholiday.getValueAt(i,4)));
ps.addBatch();
}
ps.executeBatch();
JOptionPane.showMessageDialog(null,"Holidays of month saved");
}
catch(Exception ee)
{
System.out.println(ee);
}
}
}
public static void main(String[] args)
{
holidays obj=new holidays();
}
}
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
import java.io.*;
import java.sql.*;
public class empSetting extends Frame implements ActionListener,FocusListener

```

```

{
JButton BtnNew=new JButton("New",new ImageIcon("images/new.gif"));
JButton BtnSave=new JButton("Save",new ImageIcon("images/save.gif"));
JButton BtnSearch=new JButton("Search",new ImageIcon("images/search.png"));
JButton BtnEdit=new JButton("Edit",new ImageIcon("images/edit.png"));
JButton BtnCancel=new JButton("Cancel");
JLabel LblEmp_Type=new JLabel("Employee Type:");
JLabel LblAllowance=new JLabel("Allowances...");
JLabel LblDeduction=new JLabel("Deductions...");
JLabel LblDA=new JLabel("DA:");
JLabel LblHRA=new JLabel("HRA:");
JLabel LblWA=new JLabel("WA:");
JLabel LblGPF=new JLabel("GPF");
JLabel LblIT=new JLabel("IT");
JLabel LblGIS=new JLabel("GIS");
JLabel LblPF=new JLabel("PF");
JLabel LblLIC=new JLabel("LIC");
JLabel LblTot_Allowance=new JLabel("Total Allowances:");
JLabel LblTot_Deduction=new JLabel("Total Deduction:");
JLabel LblCasual=new JLabel("Casual Leave");
JLabel LblSick=new JLabel("Sick Leave");
JLabel LblMedical=new JLabel("Medical Leave");
JLabel LblAnnual=new JLabel("Annual Leave");

JComboBox CmbEmp_Type=new JComboBox();
JTextField TxtDA=new JTextField();
JTextField TxtHRA=new JTextField();
JTextField TxtWA=new JTextField();
JTextField TxtGPF=new JTextField();
JTextField TxtIT=new JTextField();
JTextField TxtGIS=new JTextField();
JTextField TxtPF=new JTextField();
JTextField TxtLIC=new JTextField();
JTextField TxtTot_Allowance=new JTextField();

```

```

JTextField TxtTot_Deduction=new JTextField();
JTextField TxtSalary_Month=new JTextField();
JTextField TxtDate=new JTextField();
JTextField TxtCasual=new JTextField();
JTextField TxtSick=new JTextField();
JTextField TxtMedical=new JTextField();
JTextField TxtAnnual=new JTextField();

public empSetting()
{
setLayout(null);
Dimension screen=Toolkit.getDefaultToolkit().getScreenSize();
setTitle("Previlages Details");
setSize(screen.width,screen.height);
setVisible(true);
////////////////////////////////////
BtnNew.setBounds(200,490,100,30);
BtnSave.setBounds(310,490,100,30);
BtnSearch.setBounds(420,490,100,30);
BtnEdit.setBounds(530,490,100,30);
BtnCancel.setBounds(640,490,100,30);
LblEmp_Type.setBounds(100,100,150,30);
LblAllowance.setBounds(100,160,150,30);
LblDeduction.setBounds(430,160,150,30);
LblDA.setBounds(100,210,100,30);
LblHRA.setBounds(100,250,100,30);
LblWA.setBounds(100,290,100,30);
LblGPF.setBounds(430,210,100,30);
LblIT.setBounds(430,250,100,30);
LblGIS.setBounds(430,290,100,30);
LblPF.setBounds(430,330,100,30);
LblLIC.setBounds(430,370,100,30);
LblTot_Allowance.setBounds(100,410,150,30);
LblTot_Deduction.setBounds(430,410,150,30);

```

```
LblCasual.setBounds(700,210,100,30);
LblSick.setBounds(700,250,100,30);
LblMedical.setBounds(700,290,100,30);
LblAnnual.setBounds(700,330,100,30);
LblAllowance.setForeground(new Color(0,0,255));
LblDeduction.setForeground(new Color(255,0,0));
CmbEmp_Type.setBounds(270,100,150,30);
TxtDA.setBounds(220,210,150,30);
TxtHRA.setBounds(220,250,150,30);
TxtWA.setBounds(220,290,150,30);
TxtGPF.setBounds(530,210,150,30);
TxtIT.setBounds(530,250,150,30);
TxtGIS.setBounds(530,290,150,30);
TxtPF.setBounds(530,330,150,30);
TxtLIC.setBounds(530,370,150,30);
TxtTot_Allowance.setBounds(220,410,150,30);
TxtTot_Deduction.setBounds(530,410,150,30);
TxtCasual.setBounds(820,210,100,30);
TxtSick.setBounds(820,250,100,30);
TxtMedical.setBounds(820,290,100,30);
TxtAnnual.setBounds(820,330,100,30);
```

```
////////////////////////////////////
```

```
add(BtnNew);
add(BtnSave);
add(BtnSearch);
add(BtnEdit);
add(BtnCancel);
add(LblEmp_Type);
add(LblAllowance);
add(LblDeduction);
add(LblDA);
add(LblHRA);
add(LblWA);
```

```
add(LblGPF);
add(LblIT);
add(LblGIS);
add(LblPF);
add(LblLIC);
add(LblTot_Allowance);
add(LblTot_Deduction);
add(LblCasual);
add(LblSick);
add(LblMedical);
add(LblAnnual);

CmbEmp_Type.addItem("Programmer");
CmbEmp_Type.addItem("Analyst");
CmbEmp_Type.addItem("Designer");
CmbEmp_Type.addItem("Team Leader");
add(CmbEmp_Type);

add( TxtDA);
add( TxtHRA);
add( TxtWA);
add( TxtGPF);
add( TxtIT);
add( TxtGIS);
add( TxtPF);
add( TxtLIC);
add(TxtTot_Allowance);
add( TxtTot_Deduction);
add(TxtCasual);
add(TxtSick);
add(TxtMedical);
add(TxtAnnual);
BtnSearch.addActionListener(this);
BtnEdit.addActionListener(this);
```

```

BtnCancel.addActionListener(this);
BtnSave.addActionListener(this);
TxtTot_Allowance.addFocusListener(this);
TxtTot_Deduction.addFocusListener(this);
repaint();
////////////////////////////////////
}
public void focusGained(FocusEvent fe)
{
double tadd,tded;
tadd=Double.parseDouble(TxtDA.getText()+Double.parseDouble(TxtHRA.getText()+D
ouble.parseDouble(TxtWA.getText()));
tded=Double.parseDouble(TxtGPF.getText()+Double.parseDouble(TxtIT.getText()+Do
uble.parseDouble(TxtGIS.getText()+Double.parseDouble(TxtPF.getText()+Double.pars
eDouble(TxtLIC.getText()));
TxtTot_Allowance.setText(""+tadd);
TxtTot_Deduction.setText(""+tded);
}
public void focusLost(FocusEvent fe)
{
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==BtnSearch)
{
try
{
int flg=0;
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String tempcode=JOptionPane.showInputDialog(null,"Enter Employee Type:");
ResultSet res=st.executeQuery("Select * from remunerationtbl where
Emp_Type='"+tempcode+"'");

```

```

while(res.next())
{
CmbEmp_Type.setSelectedItem(tempcode);
TxtHRA.setText(res.getString(2));
TxtDA.setText(res.getString(3));
TxtWA.setText(res.getString(4));
TxtGPF.setText(res.getString(5));
TxtIT.setText(res.getString(6));
TxtGIS.setText(res.getString(7));
TxtPF.setText(res.getString(8));
TxtLIC.setText(res.getString(9));
TxtCasual.setText(res.getString(10));
TxtSick.setText(res.getString(11));
TxtMedical.setText(res.getString(12));
TxtAnnual.setText(res.getString(13));
flg=1;
}
if(flg==0)
JOptionPane.showMessageDialog(null,"No such Record present");
}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==BtnCancel)
{
this.dispose();
}
if(e.getSource()==BtnEdit)
{
try
{

```

```

Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
ResultSet res;
Statement st=con.createStatement();
res=st.executeQuery("select * from remunerationtbl");
String Str="update remunerationtbl set
hra="+Integer.parseInt(TxtHRA.getText())+",da="+Integer.parseInt(TxtDA.getText())+",
wa="+Integer.parseInt(TxtWA.getText())+",gpf="+Integer.parseInt(TxtGPF.getText())+",i
t="+Integer.parseInt(TxtIT.getText())+",gis="+Integer.parseInt(TxtGIS.getText())+",pf="
+Integer.parseInt(TxtPF.getText())+",lic="+Integer.parseInt(TxtLIC.getText())+",casual="
+Integer.parseInt(TxtCasual.getText())+",sick="+Integer.parseInt(TxtSick.getText())+",m
edical="+Integer.parseInt(TxtMedical.getText())+",annual="+Integer.parseInt(TxtAnnual.
getText())+" where Emp_Type='"+CmbEmp_Type.getSelectedItem()+"'";
st.executeUpdate(Str);
JOptionPane.showMessageDialog(null,"updated successfully");
}
catch(Exception ee1)
{
System.out.println(ee1);
}
}
if(e.getSource()==BtnSave)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="insert into
remunerationtbl(Emp_Type,Hra,Da,Wa,Gpf,It,Gis,Pf,Lic,casual,sick,medical,annual)value
s(?,?,?,?,?,?,?,?,?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
ps.setString(1,""+CmbEmp_Type.getSelectedItem());
ps.setString(2,""+TxtHRA.getText());

```



```

ps.setString(3,TxtDA.getText());
ps.setString(4,TxtWA.getText());
ps.setString(5,TxtGPF.getText());
ps.setString(6,TxtIT.getText());
ps.setString(7,TxtGIS.getText());
ps.setString(8,TxtPF.getText());
ps.setString(9,TxtLIC.getText());
ps.setString(10,TxtCasual.getText());
ps.setString(11,TxtSick.getText());
ps.setString(12,TxtMedical.getText());
ps.setString(13,TxtAnnual.getText());
ps.executeUpdate();
JOptionPane.showMessageDialog(null,"Saved Successfully");
}
catch(Exception ee)
{
System.out.println(ee);
}
}
}

public static void main(String argv[])throws IOException
{
empSetting obj=new empSetting();
}
}

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

public class empdata extends Frame implements ActionListener
{
JButton BtnNew=new JButton("New",new ImageIcon("images/new.gif"));

```

```
JButton BtnSave=new JButton("Save",new ImageIcon("images/save.gif"));
JButton BtnSearch=new JButton("Search",new ImageIcon("images/search.png"));
JButton BtnEdit=new JButton("Edit",new ImageIcon("images/edit.png"));
JButton BtnCancel=new JButton("Cancel");
JLabel LblEmp_Name=new JLabel("Employee Name:");
JLabel LblEmp_Code=new JLabel("Employee Code:");
JLabel LblEmp_Type=new JLabel("Employee Type:");
JLabel LblEmp_Desi=new JLabel("Employee Designation");
JLabel LblBasic_Pay=new JLabel("Basic Salary:");
JLabel LblAllowance=new JLabel("Allowances...");
JLabel LblDeduction=new JLabel("Deductions...");
JLabel LblDA=new JLabel("DA:");
JLabel LblHRA=new JLabel("HRA:");
JLabel LblWA=new JLabel("WA:");
JLabel LblGPF=new JLabel("GPF");
JLabel LblIT=new JLabel("IT");
JLabel LblGIS=new JLabel("GIS");
JLabel LblPF=new JLabel("PF");
JLabel LblLIC=new JLabel("LIC");
JLabel LblTot_Allowance=new JLabel("Total Allowances:");
JLabel LblTot_Deduction=new JLabel("Total Deduction:");
JLabel LblNet_Salary=new JLabel("Net Salary:");
JTextField TxtEmp_Name=new JTextField();
JTextField TxtEmp_Code=new JTextField();
JComboBox CmbEmp_Type=new JComboBox();
JTextField TxtEmp_Desi=new JTextField();
JTextField TxtBasic_Pay=new JTextField();
JTextField TxtDA=new JTextField();
JTextField TxtHRA=new JTextField();
JTextField TxtWA=new JTextField();
JTextField TxtGPF=new JTextField();
JTextField TxtIT=new JTextField();
JTextField TxtGIS=new JTextField();
JTextField TxtPF=new JTextField();
```

```

JTextField TxtLIC=new JTextField();
JTextField TxtTot_Allowance=new JTextField();
JTextField TxtTot_Deduction=new JTextField();
JTextField TxtNet_Salary=new JTextField();
JTextField TxtSalary_Month=new JTextField();
JTextField TxtDate=new JTextField();
public empdata()
{
setLayout(null);
Dimension screen=Toolkit.getDefaultToolkit().getScreenSize();
setTitle("Employee Details");
setSize(screen.width,screen.height);
setVisible(true);
////////////////////////////////////
BtnNew.setBounds(220,640,100,40);
BtnSave.setBounds(340,640,100,40);
BtnSearch.setBounds(450,640,100,40);
BtnEdit.setBounds(560,640,100,40);
BtnCancel.setBounds(670,640,100,40);
LblEmp_Type.setBounds(100,100,150,30);
LblEmp_Name.setBounds(100,150,150,30);
LblEmp_Code.setBounds(490,100,150,30);
LblEmp_Desi.setBounds(490,160,150,30);
LblBasic_Pay.setBounds(100,200,150,30);
LblAllowance.setBounds(100,280,150,30);
LblDeduction.setBounds(370,280,150,30);
LblDA.setBounds(100,340,150,30);
LblHRA.setBounds(100,380,150,30);
LblWA.setBounds(100,420,150,30);
LblGPF.setBounds(370,340,150,30);
LblIT.setBounds(370,380,150,30);
LblGIS.setBounds(370,420,150,30);
LblPF.setBounds(370,460,150,30);
LblLIC.setBounds(370,500,150,30);

```

```

LblTot_Allowance.setBounds(100,540,100,30);
LblTot_Deduction.setBounds(370,540,100,30);
LblNet_Salary.setBounds(100,580,150,30);
CmbEmp_Type.setBounds(270,100,150,30);
TxtEmp_Name.setBounds(270,150,150,30);
TxtEmp_Code.setBounds(660,100,150,30);
TxtEmp_Desig.setBounds(660,160,150,30);
TxtBasic_Pay.setBounds(270,200,150,30);
TxtDA.setBounds(180,340,150,30);
TxtHRA.setBounds(180,380,150,30);
TxtWA.setBounds(180,420,150,30);
TxtGPF.setBounds(470,340,150,30);
TxtIT.setBounds(470,380,150,30);
TxtGIS.setBounds(470,420,150,30);
TxtPF.setBounds(470,460,150,30);
TxtLIC.setBounds(470,500,150,30);
TxtTot_Allowance.setBounds(210,540,150,30);
TxtTot_Deduction.setBounds(470,540,150,30);
TxtNet_Salary.setBounds(210,580,150,30);
/////////////////////////////////lock textbox////////////////////////////////
TxtDA.setEditable(false);
TxtHRA.setEditable(false);
TxtWA.setEditable(false);
TxtGPF.setEditable(false);
TxtIT.setEditable(false);
TxtGIS.setEditable(false);
TxtPF.setEditable(false);
TxtLIC.setEditable(false);
TxtTot_Allowance.setEditable(false);
TxtTot_Deduction.setEditable(false);
TxtNet_Salary.setEditable(false);
/////////////////////////////////
add(BtnNew);
add(BtnSave);

```

```
add(BtnSearch);
add(BtnEdit);
add(BtnCancel);
add(LblEmp_Type);
add(LblEmp_Name);
add(LblEmp_Code);
add(LblEmp_Desi);
add(LblBasic_Pay);
add(LblAllowance);
add(LblDeduction);
add(LblDA);
add(LblHRA);
add(LblWA);
add(LblGPF);
add(LblIT);
add(LblGIS);
add(LblPF);
add(LblLIC);
add(LblTot_Allowance);
add(LblTot_Deduction);
add(LblNet_Salary);
add(TxtEmp_Name);
add( TxtEmp_Code);
add( TxtEmp_Desi);
add( TxtBasic_Pay);
add( TxtDA);
add( TxtHRA);
add( TxtWA);
add( TxtGPF);
add( TxtIT);
add( TxtGIS);
add( TxtPF);
add( TxtLIC);
add(TxtTot_Allowance);
```

```

add( TxtTot_Deduction);
add( TxtNet_Salary);
//TxtSalary_Month
//TxtDate
CmbEmp_Type.addItem("Programmer");
CmbEmp_Type.addItem("Analyst");
CmbEmp_Type.addItem("Designer");
CmbEmp_Type.addItem("Team Leader");
add(CmbEmp_Type);
BtnNew.addActionListener(this);
BtnEdit.addActionListener(this);
BtnCancel.addActionListener(this);
BtnSave.addActionListener(this);
BtnSearch.addActionListener(this);
////////////////////////////////////
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==BtnNew)
{
try
{
int flg=0;
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
ResultSet res=st.executeQuery("Select Emp_Code from keytbl");
while(res.next())
{
TxtEmp_Code.setText("E/"+res.getString(1));
}
}
catch(Exception ee)
{

```

```

System.out.println(ee);
}
}
if(e.getSource()==BtnSave)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
ResultSet res=st.executeQuery("Select * from employeetbl");
String str="insert into
employeetbl(Emp_Code,Emp_Type,Emp_Name,Emp_Desi,Emp_Basic)values(?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
ps.setString(1,TxtEmp_Code.getText());
ps.setString(2,""+CmbEmp_Type.getSelectedItem());
ps.setString(3,TxtEmp_Name.getText());
ps.setString(4,TxtEmp_Desi.getText());
ps.setString(5,TxtBasic_Pay.getText());
ps.executeUpdate();
str="Update keytbl set Emp_Code=Emp_Code+1";
st.executeUpdate(str);
JOptionPane.showMessageDialog(null,"Saved Successfully");

}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==BtnSearch)
{
try
{

```

```

int flg=0;
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String tempcode=JOptionPane.showInputDialog(null,"Enter Employee Code:");
ResultSet res=st.executeQuery("Select * from employeetbl where
Emp_Code='"+tempcode+"'");
while(res.next())
{
TxtEmp_Code.setText(tempcode);
CmbEmp_Type.setSelectedItem(res.getString(2));
TxtEmp_Name.setText(res.getString(3));
TxtEmp_Desi.setText(res.getString(4));
TxtBasic_Pay.setText(res.getString(5));
flg=1;
}
if(flg==0)
JOptionPane.showMessageDialog(null,"No such Record present");
else
{
ResultSet res1=st.executeQuery("Select * from remunerationtbl where
Emp_Type='"+CmbEmp_Type.getSelectedItem()+"");
while(res1.next())
{
double tbasic=Double.parseDouble(TxtBasic_Pay.getText());
double thra=Double.parseDouble(res1.getString(2))*tbasic/100;
double tda=Double.parseDouble(res1.getString(3))*tbasic/100;
double twa=Double.parseDouble(res1.getString(4))*tbasic/100;
double tgpf=Double.parseDouble(res1.getString(5))*tbasic/100;
double tit=Double.parseDouble(res1.getString(6))*tbasic/100;
double tgis=Double.parseDouble(res1.getString(7))*tbasic/100;
double tpf=Double.parseDouble(res1.getString(8))*tbasic/100;
double tlic=Double.parseDouble(res1.getString(9))*tbasic/100;
TxtHRA.setText(""+thra);

```



```

TxtDA.setText(""+tda);
TxtWA.setText(""+twa);
TxtGPF.setText(""+tgpf);
TxtIT.setText(""+tit);
TxtGIS.setText(""+tgis);
TxtPF.setText(""+tpf);
TxtLIC.setText(""+tlic);
TxtTot_Allowance.setText(""+(thra+tda+twa));
TxtTot_Deduction.setText(""+(tgpf+tit+tgis+tpf+tlic));
TxtNet_Salary.setText(""+(tbasic+(thra+tda+twa)-(tgpf+tit+tgis+tpf+tlic)));
}
}
}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==BtnEdit)
{
try
{

Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
ResultSet res;
Statement st=con.createStatement();
res=st.executeQuery("select * from employeetbl");
String Str="update employeetbl set
Emp_Code='"+TxtEmp_Code.getText()+"',Emp_Type='"+CmbEmp_Type.getSelectedIte
m()+"',Emp_Name='"+TxtEmp_Name.getText()+"',Emp_Desi='"+TxtEmp_Desi.getText()
+'',Emp_Basic="+Double.parseDouble(TxtBasic_Pay.getText())+" where
Emp_Code='"+TxtEmp_Code.getText()+"'";
st.executeUpdate(Str);

```

```
JOptionPane.showMessageDialog(null,"updated successfully");
}
catch(Exception ee1)
{
System.out.println(ee1);
}
}
if(e.getSource()==BtnCancel)
{
this.dispose();
}
}
public static void main(String argv[])throws IOException
{
empdata obj=new empdata();
}
}
```

```

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

public class departform extends Frame implements ActionListener
{
JLabel lblDepartNo=new JLabel("Department Number:");
JLabel lblDepartName=new JLabel("Department Name:");
JTextField txtDepartNo=new JTextField();
JTextField txtDepartName=new JTextField();
JButton BtnNew=new JButton("New",new ImageIcon("images/new.gif"));
JButton BtnSave=new JButton("Save",new ImageIcon("images/save.gif"));
JButton BtnSearch=new JButton("Search",new ImageIcon("images/search.png"));
JButton BtnEdit=new JButton("Edit",new ImageIcon("images/edit.png"));
JButton BtnCancel=new JButton("Cancel");

public departform()
{
Dimension screen =Toolkit.getDefaultToolkit().getScreenSize();
setLocation((screen.width - 500)/2,((screen.height-350)/2));
setVisible(true);
setSize(420,300);
setTitle("Department Form");
setLayout(null);

lblDepartNo.setBounds(50,80,150,40);
lblDepartName.setBounds(50,130,150,40);
txtDepartNo.setBounds(200,80,180,40);
txtDepartName.setBounds(200,130,180,40);
BtnNew.setBounds(220,250,100,40);
BtnSave.setBounds(340,250,100,40);
BtnSearch.setBounds(450,250,100,40);

```

```

BtnEdit.setBounds(560,250,100,40);
BtnCancel.setBounds(670,250,100,40);

add(lblDepartNo);
add(lblDepartName);
add(txtDepartNo);
add(txtDepartName);
add(BtnNew);
add(BtnSave);
add(BtnSearch);
add(BtnEdit);
add(BtnCancel);

repaint();
}
public static void main(String ar[]) throws IOException
{
departform obj=new departform();
}
public void actionPerformed(ActionEvent e)
{
}
}

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

```

```

import javax.swing.table.DefaultTableModel;
class Attendance extends JFrame implements ActionListener,FocusListener
{
String tempreg,tempname;
int tslno;
JTable tableAtt;
JScrollPane sp=new JScrollPane();
String [] col={"Sl.No.", "Employee Registration", "Employee Name"};
Object[][] data={{ tslno,tempreg,tempname } };
DefaultTableModel model=new DefaultTableModel(col,0);
JLabel lblEmpReg=new JLabel("Employee Registration Id:");
JLabel lblEmpName=new JLabel("Employee Name:");
JTextField txtEmpReg=new JTextField();
JTextField txtEmpName=new JTextField();
JLabel lblmonth=new JLabel("Month");
JLabel lblday=new JLabel("Day");
JLabel lblyear=new JLabel("Year");
JButton b=new JButton("Present");
JButton b1=new JButton("Submit");
JButton b2=new JButton("Cancel");
JButton b3=new JButton("Check");

JComboBox monthname =new JComboBox();
String mname[]={ "January", "February", "March",
"April", "May", "June", "July", "August", "September", "october", "November", "December"};
JComboBox dayname=new JComboBox();
JComboBox year=new JComboBox();
//JFrame f=new JFrame();
public Attendance()
{
int i;
setLayout(null);
setVisible(true);
setTitle("Daily Attendance...");

```

```

setSize(1000,800);
lblEmpReg.setBounds(390,50,150,40);
lblEmpName.setBounds(390,110,150,40);
txtEmpReg.setBounds(560,50,150,40);
txtEmpName.setBounds(560,110,150,40);
lblday.setBounds(50,50,150,40);
dayname.setBounds(220,50,150,40);
lblmonth.setBounds(50,110,150,40);
monthname.setBounds(220,110,150,40);
lblyear.setBounds(50,170,150,40);
year.setBounds(220,170,150,40);
b.setBounds(560,170,120,40);
b1.setBounds(560,230,120,40);
b2.setBounds(560,290,120,40);
b3.setBounds(560,400,120,40);
////////////////////////////////////Attendance Table////////////////////////////////////
tableAtt=new JTable(data,col);
//tableAtt.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
sp=new JScrollPane(tableAtt);
tableAtt.setModel(model);
sp.setBounds(100,250,300,400);
tableAtt.setBounds(100,250,300,400);

tableAtt.setFillViewportHeight(true);
////////////////////////////////////

for(i=2015;i<2030;i++)
year.addItem(i);
for(i=0;i<12;i++)
monthname.addItem(mname[i]);
for(i=1;i<=31;i++)
dayname.addItem(""+i);
add(lblmonth);add(lblday);add(lblyear);
add(b);add(b1);add(b2);add(b3);

```

```

add(monthname);add(dayname);add(year);
add(lblEmpReg);add(lblEmpName);add(txtEmpReg);add(txtEmpName);
add(sp);
b.addActionListener(this);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
txtEmpReg.addFocusListener(this);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
repaint();
}
public void focusGained(FocusEvent fe)
{
}
public void focusLost(FocusEvent fe)
{
try
{
int flg=0;
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="select * from employeetbl where Emp_code="+txtEmpReg.getText()+"";
ResultSet res=st.executeQuery(str);
while(res.next())
{
flg=1;
txtEmpName.setText(""+res.getString(3));
}
if(flg==0)
txtEmpName.setText("");
}
catch(Exception ee)
{

```

```

}
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b)
{
int found=0,flg=0;
for(int i=0;i<model.getRowCount();i++)
{
if(txtEmpReg.getText().equalsIgnoreCase(""+tableAtt.getValueAt(i,1)))
{
found=1;
break;
}
}
////////////////////////////////////
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="select * from employeetbl where Emp_code="+txtEmpReg.getText()+"";
ResultSet res=st.executeQuery(str);
while(res.next())
{
flg=1;
}
}
catch(Exception ee)
{
}
////////////////////////////////////
if(found==0&&flg==1)
{

```



```

Vector v=new Vector();
v.add(""+(tableAtt.getRowCount()+1));
v.add(txtEmpReg.getText());
v.add(txtEmpName.getText());
model.addRow(v);
}
else
{
if(found==1)
OptionPane.showMessageDialog(null,"Already Present");
if(flg==0)
OptionPane.showMessageDialog(null,"No such Employee exists");

}
}
if(e.getSource()==b2)
{
this.dispose();
}
if(e.getSource()==b3)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="select * from Attendencetbl where dday='"+dayname.getSelectedItem()+" and
mmonth='"+monthname.getSelectedItem()+" and yyear='"+year.getSelectedItem()+"";
ResultSet res=st.executeQuery(str);
model.setRowCount(0);
while(res.next())
{
Vector v=new Vector();
v.add(""+(tableAtt.getRowCount()+1));

```

```

v.add(res.getString(1));
v.add(res.getString(5));
model.addRow(v);
}
}
catch(Exception ee)
{
}
}
if(e.getSource()==b1)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="insert into
Attendencetbl(Emp_Code,dday,mmonth,yyeear,Emp_Name)values(?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
for(int i=0;i<tableAtt.getRowCount();i++)
{
String tempcode=(String)tableAtt.getValueAt(i,1);
ps.setString(1,tempcode);
ps.setString(2,(""+dayname.getSelectedItemAt()));
ps.setString(3,(""+monthname.getSelectedItemAt()));
ps.setString(4,(""+year.getSelectedItemAt()));
ps.setString(5,txtEmpName.getText());
ps.addBatch();
}
ps.executeBatch();
JOptionPane.showMessageDialog(null,"Attendance Saved");
}
catch(Exception ee)
{

```

```

System.out.println(ee);
}
}
}
public static void main(String[] args)
{
Attendance obj=new Attendance();
}
}
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.io.*;
import java.util.*;
import java.text.DateFormat;
import java.util.Date;
import java.util.*;
import javax.swing.table.DefaultTableModel;
class Attendance extends JFrame implements ActionListener,FocusListener
{
String tempreg,tempname;
int tslno;
JTable tableAtt;
JScrollPane sp=new JScrollPane();
String [] col={"Sl.No.", "Employee Registration", "Employee Name"};
Object[][] data={{ tslno,tempreg,tempname }};
DefaultTableModel model=new DefaultTableModel(col,0);
JLabel lblEmpReg=new JLabel("Employee Registration Id:");
JLabel lblEmpName=new JLabel("Employee Name:");
JTextField txtEmpReg=new JTextField();
JTextField txtEmpName=new JTextField();
JLabel lblmonth=new JLabel("Month");
JLabel lblday=new JLabel("Day");

```

```

JLabel lblyear=new JLabel("Year");
JButton b=new JButton("Present");
JButton b1=new JButton("Submit");
JButton b2=new JButton("Cancel");
JButton b3=new JButton("Check");

JComboBox monthname =new JComboBox();
String mname[]={ "January", "February", "March",
"April", "May", "June", "July", "August", "September", "october", "November", "December" };
JComboBox dayname=new JComboBox();
JComboBox year=new JComboBox();
//JFrame f=new JFrame();
public Attendance()
{
int i;
setLayout(null);
setVisible(true);
setTitle("Daily Attendance...");
setSize(1000,800);
lblEmpReg.setBounds(390,50,150,40);
lblEmpName.setBounds(390,110,150,40);
txtEmpReg.setBounds(560,50,150,40);
txtEmpName.setBounds(560,110,150,40);
lblday.setBounds(50,50,150,40);
dayname.setBounds(220,50,150,40);
lblmonth.setBounds(50,110,150,40);
monthname.setBounds(220,110,150,40);
lblyear.setBounds(50,170,150,40);
year.setBounds(220,170,150,40);
b.setBounds(560,170,120,40);
b1.setBounds(560,230,120,40);
b2.setBounds(560,290,120,40);
b3.setBounds(560,400,120,40);
//////////Attendance Table//////////

```

```

tableAtt=new JTable(data,col);
//tableAtt.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
sp=new JScrollPane(tableAtt);
tableAtt.setModel(model);
sp.setBounds(100,250,300,400);
tableAtt.setBounds(100,250,300,400);

tableAtt.setFillViewportHeight(true);
////////////////////////////////////

for(i=2015;i<2030;i++)
year.addItem(i);
for(i=0;i<12;i++)
monthname.addItem(mname[i]);
for(i=1;i<=31;i++)
dayname.addItem(""+i);
add(lblmonth);add(lblday);add(lblyear);
add(b);add(b1);add(b2);add(b3);
add(monthname);add(dayname);add(year);
add(lblEmpReg);add(lblEmpName);add(txtEmpReg);add(txtEmpName);
add(sp);
b.addActionListener(this);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
txtEmpReg.addFocusListener(this);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
repaint();
}
public void focusGained(FocusEvent fe)
{
}
public void focusLost(FocusEvent fe)
{
}

```

```

try
{
int flg=0;
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="select * from employeetbl where Emp_code="+txtEmpReg.getText()+"";
ResultSet res=st.executeQuery(str);
while(res.next())
{
flg=1;
txtEmpName.setText(""+res.getString(3));
}
if(flg==0)
txtEmpName.setText("");
}
catch(Exception ee)
{
}
}

public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b)
{
int found=0,flg=0;
for(int i=0;i<model.getRowCount();i++)
{
if(txtEmpReg.getText().equalsIgnoreCase(""+tableAtt.getValueAt(i,1)))
{
found=1;
break;
}
}
}

////////////////////////////////////

```

```

try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="select * from employeetbl where Emp_code='"+txtEmpReg.getText()+"'";
ResultSet res=st.executeQuery(str);
while(res.next())
{
flg=1;
}
}
catch(Exception ee)
{
}
////////////////////////////////////
if(found==0&&flg==1)
{
Vector v=new Vector();
v.add(""+(tableAtt.getRowCount()+1));
v.add(txtEmpReg.getText());
v.add(txtEmpName.getText());
model.addRow(v);
}
else
{
if(found==1)
OptionPane.showMessageDialog(null,"Already Present");
if(flg==0)
OptionPane.showMessageDialog(null,"No such Employee exists");

}
}
if(e.getSource()==b2)

```

```

{
this.dispose();
}
if(e.getSource()==b3)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="select * from Attendencetbl where dday='"+dayname.getSelectedItemAt()+"' and
mmonth='"+monthname.getSelectedItemAt()+"' and yyear='"+year.getSelectedItemAt()+"'";
ResultSet res=st.executeQuery(str);
model.setRowCount(0);
while(res.next())
{
Vector v=new Vector();
v.add(""+(tableAtt.getRowCount()+1));
v.add(res.getString(1));
v.add(res.getString(5));
model.addRow(v);
}
}
catch(Exception ee)
{
}
}
if(e.getSource()==b1)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();

```



```

String str="insert into
Attendancetbl(Emp_Code,dday,mmonth,yyear,Emp_Name)values(?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
for(int i=0;i<tableAtt.getRowCount();i++)
{
String tempcode=(String)tableAtt.getValueAt(i,1);
ps.setString(1,tempcode);
ps.setString(2,(""+dayname.getSelectedItemAt()));
ps.setString(3,(""+monthname.getSelectedItemAt()));
ps.setString(4,(""+year.getSelectedItemAt()));
ps.setString(5,txtEmpName.getText());
ps.addBatch();
}
ps.executeBatch();
JOptionPane.showMessageDialog(null,"Attendance Saved");
}
catch(Exception ee)
{
System.out.println(ee);
}
}
}
public static void main(String[] args)
{
Attendance obj=new Attendance();
}

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.io.*;
import java.util.*;
import java.text.DateFormat;
import java.util.Date;
import java.util.*;
import javax.swing.table.DefaultTableModel;
class absent extends JFrame implements ActionListener
{
String tempreg,tslno,tempname;
JTable tableAtt;
JScrollPane sp=new JScrollPane();
String [] col={"Sl.No.", "Employee Code", "Employee Name"};
Object[][] data={{tslno,tempreg,tempname}};
DefaultTableModel model=new DefaultTableModel(col,0);
JLabel lblEmpReg=new JLabel("Day:");
JTextField txtday=new JTextField();
JLabel lblmonth=new JLabel("Month");

JLabel lblyear=new JLabel("Year");
JButton b=new JButton("Check Absentees");
JButton b1=new JButton("Submit");
JButton b2=new JButton("Cancel");
JComboBox monthname =new JComboBox();
String mname[]={ "January", "February", "March",
"April", "May", "June", "July", "August", "September", "october", "November", "December"};

JComboBox year=new JComboBox();

public absent()
{

```

```

int i;
setLayout(null);
setVisible(true);
setTitle("Absent Record...");
setSize(1000,800);
lblEmpReg.setBounds(50,50,150,40);
txtday.setBounds(220,50,150,40);
lblmonth.setBounds(50,110,150,40);
monthname.setBounds(220,110,150,40);
lblyear.setBounds(50,170,150,40);
year.setBounds(220,170,150,40);
b.setBounds(560,110,120,40);
b1.setBounds(560,160,120,40);
b2.setBounds(560,210,120,40);
////////////////////Attendance Table////////////////////
tableAtt=new JTable(data,col);
//tableAtt.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
sp=new JScrollPane(tableAtt);
tableAtt.setModel(model);
sp.setBounds(100,250,300,400);
tableAtt.setBounds(100,250,300,400);

tableAtt.setFillViewportHeight(true);
////////////////////

for(i=2015;i<2030;i++)
year.addItem(i);
for(i=0;i<12;i++)
monthname.addItem(mname[i]);
add(lblmonth);add(lblyear);
add(b);add(b1);add(b2);add(monthname);add(year);
add(lblEmpReg);add(txtday);
add(sp);
b.addActionListener(this);

```

```

b1.addActionListener(this);
b2.addActionListener(this);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
repaint();
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
ResultSet res=st.executeQuery("Select * from attendencetbl where
dday='"+txtday.getText()+"' and mmonth='"+monthname.getSelectedItem()+"' and
yyyear='"+year.getSelectedItem()+"'");
String str="";
while(res.next())
{
str=str+res.getString(1)+" ";
}
str=str+" ";
Connection con1;
con1=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st1=con1.createStatement();
ResultSet res1=st.executeQuery("Select * from employeetbl where emp_code not in
("+str+")");
model.setRowCount(0);
while(res1.next())
{
Vector v=new Vector();
v.add(""+(tableAtt.getRowCount()+1));
v.add(""+res1.getString(1));
}
}
}

```

```

v.add(""+res1.getString(3));
model.addRow(v);
}
}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==b2)
{
this.dispose();
}
if(e.getSource()==b1)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="insert into
absenttbl(Emp_Code,Emp_name,dday,mmonth,yyear)values(?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
for(int i=0;i<tableAtt.getRowCount();i++)
{
ps.setString(1,""+tableAtt.getValueAt(i,1));
ps.setString(2,""+tableAtt.getValueAt(i,2));
ps.setString(3,txtday.getText());
ps.setString(4,""+monthname.getSelectedItemAt());
ps.setString(5,""+year.getSelectedItemAt());
ps.executeUpdate();
}
JOptionPane.showMessageDialog(null,"Absentees Record Saved");
}

```

```
catch(Exception ee)
{
System.out.println(ee);
}
}
}
public static void main(String[] args)
{
absent obj=new absent();
}
}
```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.io.*;
import java.util.*;
import java.text.DateFormat;
import java.util.Date;
import java.util.*;
import javax.swing.table.DefaultTableModel;
class Attendance extends JFrame implements ActionListener,FocusListener
{
String tempreg,tempname;
int tslno;
JTable tableAtt;
JScrollPane sp=new JScrollPane();
String [] col={"Sl.No.", "Employee Registration", "Employee Name"};
Object[][] data={{ tslno,tempreg,tempname } };
DefaultTableModel model=new DefaultTableModel(col,0);
JLabel lblEmpReg=new JLabel("Employee Registration Id:");
JLabel lblEmpName=new JLabel("Employee Name:");
JTextField txtEmpReg=new JTextField();
JTextField txtEmpName=new JTextField();
JLabel lblmonth=new JLabel("Month");
JLabel lblday=new JLabel("Day");
JLabel lblyear=new JLabel("Year");
JButton b=new JButton("Present");
JButton b1=new JButton("Submit");
JButton b2=new JButton("Cancel");
JButton b3=new JButton("Check");

JComboBox monthname =new JComboBox();
String mname[]={ "January", "February", "March",
"April", "May", "June", "July", "August", "September", "october", "November", "December"};

```

```

JComboBox dayname=new JComboBox();
JComboBox year=new JComboBox();
//JFrame f=new JFrame();
public Attendance()
{
int i;
setLayout(null);
setVisible(true);
setTitle("Daily Attendance...");
setSize(1000,800);
lblEmpReg.setBounds(390,50,150,40);
lblEmpName.setBounds(390,110,150,40);
txtEmpReg.setBounds(560,50,150,40);
txtEmpName.setBounds(560,110,150,40);
lblday.setBounds(50,50,150,40);
dayname.setBounds(220,50,150,40);
lblmonth.setBounds(50,110,150,40);
monthname.setBounds(220,110,150,40);
lblyear.setBounds(50,170,150,40);
year.setBounds(220,170,150,40);
b.setBounds(560,170,120,40);
b1.setBounds(560,230,120,40);
b2.setBounds(560,290,120,40);
b3.setBounds(560,400,120,40);
//////////Attendance Table//////////
tableAtt=new JTable(data,col);
//tableAtt.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
sp=new JScrollPane(tableAtt);
tableAtt.setModel(model);
sp.setBounds(100,250,300,400);
tableAtt.setBounds(100,250,300,400);

tableAtt.setFillViewportHeight(true);
//////////

```



```

for(i=2015;i<2030;i++)
year.addItem(i);
for(i=0;i<12;i++)
monthname.addItem(mname[i]);
for(i=1;i<=31;i++)
dayname.addItem(""+i);
add(lblmonth);add(lblday);add(lblyear);
add(b);add(b1);add(b2);add(b3);
add(monthname);add(dayname);add(year);
add(lblEmpReg);add(lblEmpName);add(txtEmpReg);add(txtEmpName);
add(sp);
b.addActionListener(this);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
txtEmpReg.addFocusListener(this);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
repaint();
}
public void focusGained(FocusEvent fe)
{
}
public void focusLost(FocusEvent fe)
{
try
{
int flg=0;
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="select * from employeetbl where Emp_code="+txtEmpReg.getText()+"";
ResultSet res=st.executeQuery(str);
while(res.next())

```

```

{
    flg=1;
    txtEmpName.setText(""+res.getString(3));
}
if(flg==0)
    txtEmpName.setText("");
}
catch(Exception ee)
{
}
}
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==b)
    {
        int found=0,flg=0;
        for(int i=0;i<model.getRowCount();i++)
        {
            if(txtEmpReg.getText().equalsIgnoreCase(""+tableAtt.getValueAt(i,1)))
            {
                found=1;
                break;
            }
        }
        //////////////////////////////////////
        try
        {
            Connection con;
            con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
            Statement st=con.createStatement();
            String str="select * from employeetbl where Emp_code="+txtEmpReg.getText()+"";
            ResultSet res=st.executeQuery(str);
            while(res.next())
            {

```

```

flg=1;
}
}
catch(Exception ee)
{
}
////////////////////////////////////
if(found==0&&flg==1)
{
Vector v=new Vector();
v.add(""+(tableAtt.getRowCount()+1));
v.add(txtEmpReg.getText());
v.add(txtEmpName.getText());
model.addRow(v);
}
else
{
if(found==1)
JOptionPane.showMessageDialog(null,"Already Present");
if(flg==0)
JOptionPane.showMessageDialog(null,"No such Employee exists");

}
}
if(e.getSource()==b2)
{
this.dispose();
}
if(e.getSource()==b3)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");

```

```

Statement st=con.createStatement();
String str="select * from Attendencetbl where dday='"+dayname.getSelectedItemAt()+" and
mmonth='"+monthname.getSelectedItemAt()+" and yyear='"+year.getSelectedItemAt()+"";
ResultSet res=st.executeQuery(str);
model.setRowCount(0);
while(res.next())
{
Vector v=new Vector();
v.add(""+(tableAtt.getRowCount()+1));
v.add(res.getString(1));
v.add(res.getString(5));
model.addRow(v);
}
}
catch(Exception ee)
{
}
}
if(e.getSource()==b1)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="insert into
Attendencetbl(Emp_Code,dday,mmonth,yyear,Emp_Name)values(?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
for(int i=0;i<tableAtt.getRowCount();i++)
{
String tempcode=(String)tableAtt.getValueAt(i,1);
ps.setString(1,tempcode);
ps.setString(2,(""+dayname.getSelectedItemAt()));
ps.setString(3,(""+monthname.getSelectedItemAt()));

```

```

ps.setString(4,(""+year.getSelectedItem()));
ps.setString(5,txtEmpName.getText());
ps.addBatch();
}
ps.executeBatch();
JOptionPane.showMessageDialog(null,"Attendance Saved");
}
catch(Exception ee)
{
System.out.println(ee);
}
}
}
public static void main(String[] args)
{
Attendance obj=new Attendance();
}
}
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.io.*;
import java.util.*;
import java.text.DateFormat;
import java.util.Date;
import java.util.*;
import javax.swing.table.DefaultTableModel;
class absent extends JFrame implements ActionListener
{
String tempreg,tslno,tempname;
JTable tableAtt;
JScrollPane sp=new JScrollPane();
String [] col={"Sl.No.", "Employee Code", "Employee Name"};

```

```

Object[][] data={{ tslno,tempreg,tempname }};
DefaultTableModel model=new DefaultTableModel(col,0);
JLabel lblEmpReg=new JLabel("Day:");
JTextField txtday=new JTextField();
JLabel lblmonth=new JLabel("Month");

JLabel lblyear=new JLabel("Year");
JButton b=new JButton("Check Absentees");
JButton b1=new JButton("Submit");
JButton b2=new JButton("Cancel");
JComboBox monthname =new JComboBox();
String mname[]={"January","February", "March",
"April","May","June","July","August","September","october","November","December"};

JComboBox year=new JComboBox();

public absent()
{
int i;
setLayout(null);
setVisible(true);
setTitle("Absent Record...");
setSize(1000,800);
lblEmpReg.setBounds(50,50,150,40);
txtday.setBounds(220,50,150,40);
lblmonth.setBounds(50,110,150,40);
monthname.setBounds(220,110,150,40);
lblyear.setBounds(50,170,150,40);
year.setBounds(220,170,150,40);
b.setBounds(560,110,120,40);
b1.setBounds(560,160,120,40);
b2.setBounds(560,210,120,40);
//////////Attendance Table//////////
tableAtt=new JTable(data,col);

```

```

//tableAtt.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
sp=new JScrollPane(tableAtt);
tableAtt.setModel(model);
sp.setBounds(100,250,300,400);
tableAtt.setBounds(100,250,300,400);

tableAtt.setFillViewportHeight(true);
////////////////////////////////////

for(i=2015;i<2030;i++)
year.addItem(i);
for(i=0;i<12;i++)
monthname.addItem(mname[i]);
add(lblmonth);add(lbyear);
add(b);add(b1);add(b2);add(monthname);add(year);
add(lblEmpReg);add(txtday);
add(sp);
b.addActionListener(this);
b1.addActionListener(this);
b2.addActionListener(this);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
repaint();
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b)
{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();

```

```

ResultSet res=st.executeQuery("Select * from attendencetbl where
dday='"+txtday.getText()+"' and mmonth='"+monthname.getSelectedItems()+"' and
yyear='"+year.getSelectedItems()+"");
String str="";
while(res.next())
{
str=str+res.getString(1)+" ";
}
str=str+" ";
Connection con1;
con1=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st1=con1.createStatement();
ResultSet res1=st.executeQuery("Select * from employeetbl where emp_code not in
("+str+")");
model.setRowCount(0);
while(res1.next())
{
Vector v=new Vector();
v.add(""+(tableAtt.getRowCount()+1));
v.add(""+res1.getString(1));
v.add(""+res1.getString(3));
model.addRow(v);
}
}
catch(Exception ee)
{
System.out.println(ee);
}
}
if(e.getSource()==b2)
{
this.dispose();
}
if(e.getSource()==b1)

```



```

{
try
{
Connection con;
con=DriverManager.getConnection("jdbc:odbc:payrolldsn");
Statement st=con.createStatement();
String str="insert into
absenttbl(Emp_Code,Emp_name,dday,mmonth,yyear)values(?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(str);
for(int i=0;i<tableAtt.getRowCount();i++)
{
ps.setString(1,""+tableAtt.getValueAt(i,1));
ps.setString(2,""+tableAtt.getValueAt(i,2));
ps.setString(3,txtday.getText());
ps.setString(4,""+monthname.getSelectedItemAt());
ps.setString(5,""+year.getSelectedItemAt());
ps.executeUpdate();
}
JOptionPane.showMessageDialog(null,"Absentees Record Saved");
}
catch(Exception ee)
{
System.out.println(ee);
}
}
}

public static void main(String[] args)
{
absent obj=new absent();
}
}

```

SCREENSHOTS

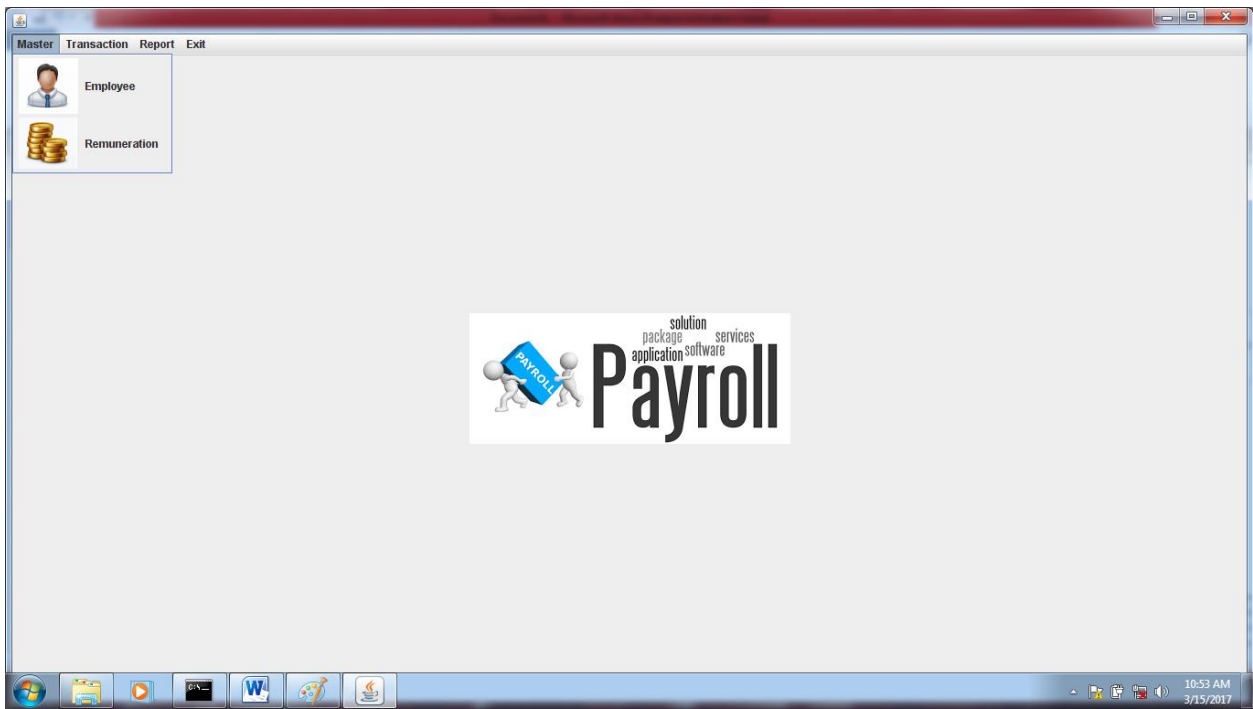
THIS IS OUR FIRST USER INTERFACE LOOKS GEATE



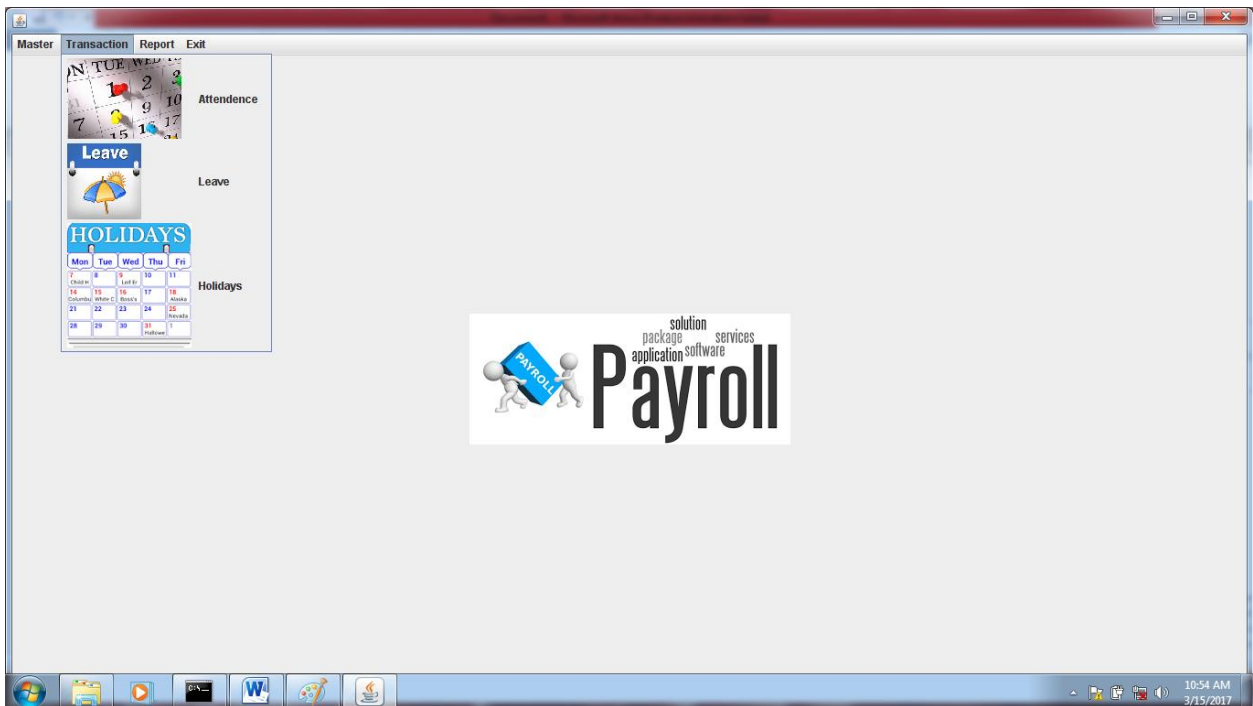
AND THIS IS THE USER ID AND PASSWORD CREDENTIAL INTERFACE YOU CAN ENTER YOUR ID AND PASSWORD ACCORDINGLY



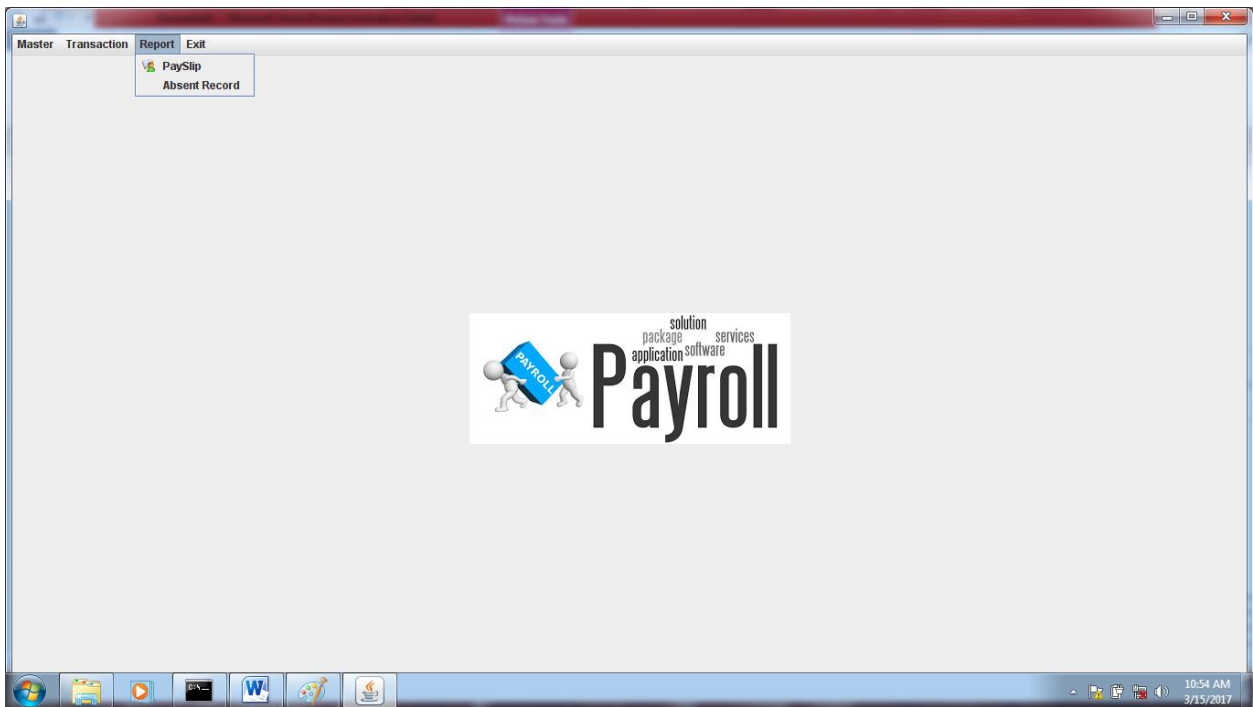
AFTER LOGGED IN THIS TY0PE OF DOMAINS ARE AVAILABLE IN THIS APP



THERE ARE THREE TYPES OF EMPLOYEE INFORMATION MENTIONED



IF YOU NEED TO CHECK SALRY STATUS THAN YOU CAN SEARCH HERE



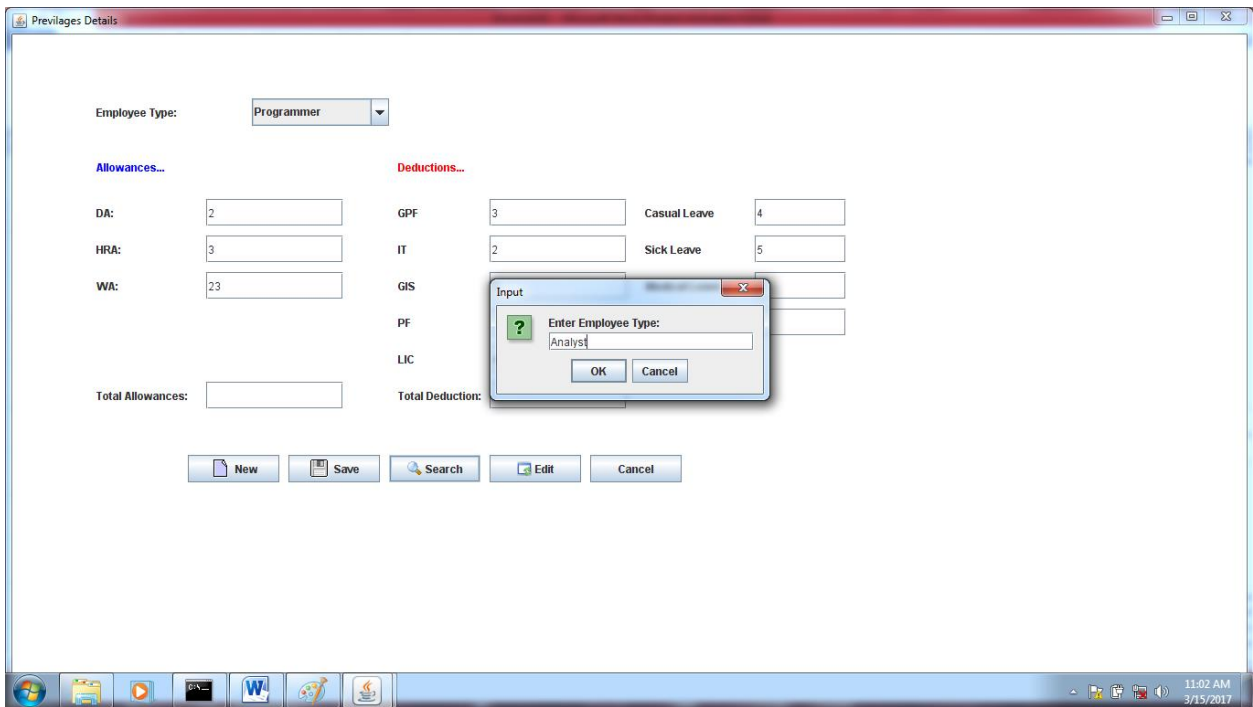
YOU WILL NEED TO ENTER THIS TYPE OF DETAILS AND THAN YOU WILL SEE RESULTS

Employee Type:	Programmer	Employee Code:	E001
Employee Name:	Rajan Singh	Employee Designation:	Senior
Basic Salary:	30000		

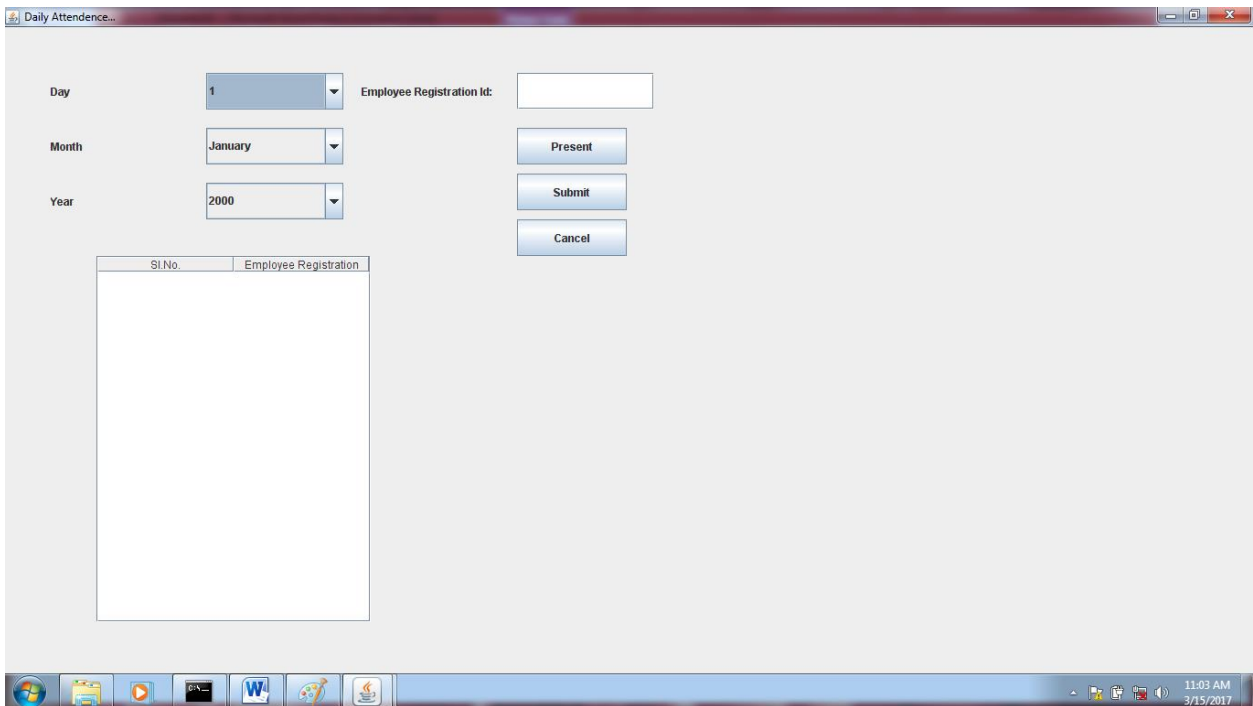
Allowances...		Deductions...	
DA:	600.0	GPF:	900.0
HRA:	900.0	IT:	600.0
WA:	6900.0	GIS:	600.0
		PF:	900.0
		LIC:	12000.0
Total Allowances:	8400.0	Total Deduction:	15000.0
Net Salary:	23400.0		

Buttons: New, Save, Search, Edit, Cancel

LEAVE DETAILS IF YOU WANT TO TAKE LEAVE THAN YOU HAVE TO FILL THIS



EMPLOYEE REGISTRATION PROCESS



EMPLOYEE HOLYDAY DETAILS

Monthly Holidays Entry...

Day: 1 Holiday Description:

Month: March

Year: 2017

Sl.No.	Holiday ...	day	month	year

11:03 AM 3/15/2017

EMPLOYEE ADDRESS DETAILS

Employee Leave Application Form...

Employee Code: E008 Employee Type: Analyst

Employee Name: Ramandeep Employee Designation: senior

Leave Code: Casual Leave

Start Date:

End Date:

Number Of Days:

Reason of leave:

Reference Number: 4

11:04 AM 3/15/2017

EMPLOYEE PROFILE DETAILS DESIGNATION AND ALL

EMPLOYEE PROFILE DETAILS DESIGNATION AND ALL

Employee Details

Pay Slip For Month:

Employee Type: Employee Code:

Employee Name: Employee Designation:

Basic Salary:

Allowances...		Deductions...			
DA:	<input type="text" value="800.0"/>	GPF	<input type="text" value="400.0"/>	Casual Leave	<input type="text"/>
HRA:	<input type="text" value="800.0"/>	IT	<input type="text" value="800.0"/>	Sick Leave	<input type="text"/>
WA:	<input type="text" value="1200.0"/>	GIS	<input type="text" value="800.0"/>	Medical Leave	<input type="text" value="2.0"/>
		PF	<input type="text" value="2400.0"/>	Annual Leave	<input type="text"/>
		LIC	<input type="text" value="1200.0"/>		
Total Allowances:	<input type="text" value="2800.0"/>	Total Deduction:	<input type="text" value="5600.0"/>		
Net Salary:	<input type="text" value="37200.0"/>				

11:05 AM 3/15/2017

Implementation and System Testing

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

SYSTEM TESTING

The goal of the system testing process was to determine all faults in our project .The program was subjected to set of test inputs and many explanations were made and based on these explanations it will be decided whether th program behaves as expected or not. Our Project went through two levels of testing

Unit testing

Integration testing

UNIT TESTING

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require

The procedures belonging to other units that the unit under test calls

Non local data structures that module accesses

A procedure to call the functions of the unit under test with appropriate parameters

Test for the admin module

Testing admin login form-This form is used for log in of administrator of the system. In this form we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.

Report Generation: admin can generate report from the main database.

INTEGRATION TESTING

In the Integration testing we test various combination of the project module by providing the input.

The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes

FUTURE SCOPE

collection will be obviously, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to college. It will also reduce the cost of collecting data about the employees. The management & collection procedure will go on smoothly with the help of this system.

The present project has been developed to meet the aspirations indicated in the modern age. An attempt has been made through this project to do all work fast & easily. It provides many facilities to accomplish the desired objectives. The facility Included in this project and the suggested activities have been organized to impart knowledge & develop skill & attitude **the other module.**

CONCLUSION

In this report, an information system's development has been presented. It was emphasized on the basic steps, consequently taken during the project's development course as a particular attention was turned to the basic operative functions performed upon the data into the database. The report's content comprises the whole task solution, starting from the programming environments have been selected, going through the database, the applications analyze and construction, and finishing with the code-implementation and test-samples, shown separately in Appendix chapters. As a future work, some additional stuff could be implemented and integrated into the application code making it much more reliable and flexible; especially what concerns a pay-roll module, for instance. Apparently, the role of such systems is basic and essential within each company that wants to keep a really good control and record concerning its personnel data, functionality and performance on all levels in its structure. Every organization, in nowadays, has the necessity of managing its staff on a really good level as the staff has definitely the greatest merit of building up a company as such as it is. The wellmanaged staff means giving the appropriate financial award-ness and all kind of benefits as such as they have been deserved. That's why the development of such systems is not just a programming business – a lot of people are ordinarily involved in such projects and one of the basic requirements is the reliability of the system, especially what concerns the storage of data and all of the operations that will be performed upon it.