

COMPUTER SCIENCE SEM V

# Pocket Muster

Project Report

---

By Ms. **Janvi Koli**

Roll No 51

For PARIAL FULFILLMENT of Degree of

## BACHELORS OF COMPUTER SCIENCE

UNIVERSITY OF MUMBAI

Academic Year

2020-2021

UNDER THE GUIDANCE OF

**Prof. Nizam Sayyed**

S.K. Somaia Degree College of Arts, Science and Commerce

VIDYAVIHAR(EAST)

MUMBAI-400 077

---

---

## CERTIFICATE

---

This is to certify that the project titled “POCKET MUSTER” has been done by Ms Janvi Koli student of Third Year Computer Science as per the project requirement for Sem V for the academic year 2020-2021 under the guidance of Prof. Nizam Sayyed.

SK Somaiya College of Arts, Science and Commerce

University of Mumbai

Signature of Project Guide: \_\_\_\_\_

Date:

---

Signature of Examiner

Date:

---

Signature of Department Head

Date:

---

## INDEX

1.	Introduction	4
	Objective	4
	Feature	4
2.	Analysis	5
	History of Payroll	5
	Cost Benefit Analysis	5
	Contribution of Payroll	6
	Drawbacks in existing software	7
	Advantages of Payroll	7
	Feasibility Study	8
	Requirement Specification	10
3.	Design Phase	11
	ER Diagram	11
	Flowchart	12
	Class Diagram	13
	Sequence Diagram	14
4.	Coding Phase with respective screenshots and database	15
5.	Testing Phase	66
6.	Future Scope	67
7.	References	68

---

## Introduction

---

Payroll system is the heart of any Human Resource System of an organization. The solution has to take care of the calculation of salary as per rules of the company and various deductions to be done from the salary. It has to generate pay-slip, mark attendance. Managing thousands of odd papers, pay slips, payroll reports and salary details and so on can be tiresome. If we have a payroll processing system which will generate our pay slips and payroll reports within seconds would be very helpful and would save our time.

---

## Objective Of Payroll Management System

---

The main objective of the payroll management system is to provide an easy way out to automate all the functionalities. It manages the details of the employees, calculates their salary, marks attendance. It will manage all the details of the employees and payments. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of this project is to reduce the manual work.

---

## Features of payroll management system is as follows

---

- The data is only accessible to the administrator
- To improve the efficiency.
- Stores all the details of the employees
- The administrator can easily change or update an employees data
- Manages the information of salary
- Automatically calculates the salary based on the number of working days of the employee
- List employees attendance
- Every employee has a special employee ID
- With employee ID you can fetch the employees data as well a salary
- Prints salary slip
- Quickly finds out information of an employee details
- Provides easy and faster access information.
- Administrator needs to only have basic information about the database.

---

## History of Payroll Management System

---

Primarily, payroll can trace its roots back to bookkeeping in most industrialised nations. In the first instance bookkeeping was a way for merchants to keep track of their sales and outgoings, but over time as trade and business has developed on a large scale it has developed into something much more complex.

These days companies need to keep track of their employees' pay as well as their own buying and selling, and with tax and legislation ever changing it the need for accurate and efficient payroll solutions has become paramount. These days companies no longer need to manage their own payroll as there are companies that specialise in payroll management who can take care of it for you.

In today's workplace any business which employs staff needs a fluent and effective payroll system. It enables you to comply with legal obligations and to ensure that you are deducting the right amount of income tax and national insurance contributions.

It's important for companies to keep on top of this as failure to comply can land businesses in trouble – and end up costing them a lot of money, and the larger your company is the more difficult it will be to manage your payroll.

---

## Cost Benefit Analysis

---

- Role Based Screen Access
- The roles can be any of the following:
- *Administrator Role* - is for provisioning of Register/Update/Remove/Transfer Contract.
- *Data Entry role* – for entering of basic form data
- *Supervisory role* – for review/approval/rejection/modification of data
- *Decision making role* – viewing of report and decision making

---

## Contribution

---

Payroll Software is programmed to enable HR professionals for managing the data and information of the company's employees on the regular basis. The easy information regarding the employees is including contact details, investment details, attendance, salary data and much more.

### Accuracy:

Your payroll system must be able to accommodate your employees and their various working hours. Once a good system is in place, you will have an accurate record of the working hours of your employees. An automated system minimizes the possibility of human error, as the only real factor is whether employees remember to clock in and out.

### Record-Keeping:

Another feature of a payroll system is that it can be used to keep detailed and accurate records. These records can be stored in a main database on site, online or in an outside record-keeping facility. Record-keeping can help you monitor trends such as how much overtime you're paying and how many employees participate in your company retirement plans

---

## Drawbacks of the Existing Project

---

- Payroll software can be expensive
- You may need to hire staff to process your payroll
- You must constantly backup your Payroll Data
- Access to Payroll may be limited
- Your payroll software may lead to withloading errors
- You are still responsible for handling your payroll taxes

---

## Advantages Of Payroll System

---

Many businesses choose to use payroll software over manual processing, as it can help them to:

- work out payroll calculations and deductions quicker
- generate accurate payslips
- calculate bonuses, expenses, holiday pay, etc with minimum effort
- automate certain tasks, such as year-end reporting
- reduce the burden of compliance
- remove the need to understand complex tax legislation
- store data such as payslips and annual reports in a secure, easily accessible system

---

## **Feasibility Study**

To analyze whether the software will meet organizational requirements Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed.

To determine whether the software can be implemented using the current technology and within the specified budget and schedule A feasibility study is a preliminary investigation of a proposed system to decide whether the system can run smoothly with the organization.

To determine whether the software can be integrated with other existing software.benefits that are expected and to decide will the organization go for it.

### **Three Types of feasibility study**

#### ***1. Technical Feasibility.***

Ascertainment that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required

#### ***2. Operational Feasibility.***

Analyzes whether users will adapt to a new software.Determines whether the organization is satisfied by the alternative solutions proposed by the software development team

#### ***3. Economical Feasibility***

Cost of hardware, software, development team, and training. Feasible only if tangible and intangible benefits outweigh the cost. The cost for proposed online shopping system is outweighing the cost and efforts involved

---

## Flowchart

---

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

## Class Diagram

---

A class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

## E-R Diagram

---

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.

## Sequence Diagram

---

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a

---

---

group of objects works together. These diagrams are used to understand requirements for a new system or to document an existing process.

## Requirement Specification

---

For the development of this project both hardware and software were required. The hard required is any machine compatible for running the softwares and resources required.

### Software Requirements

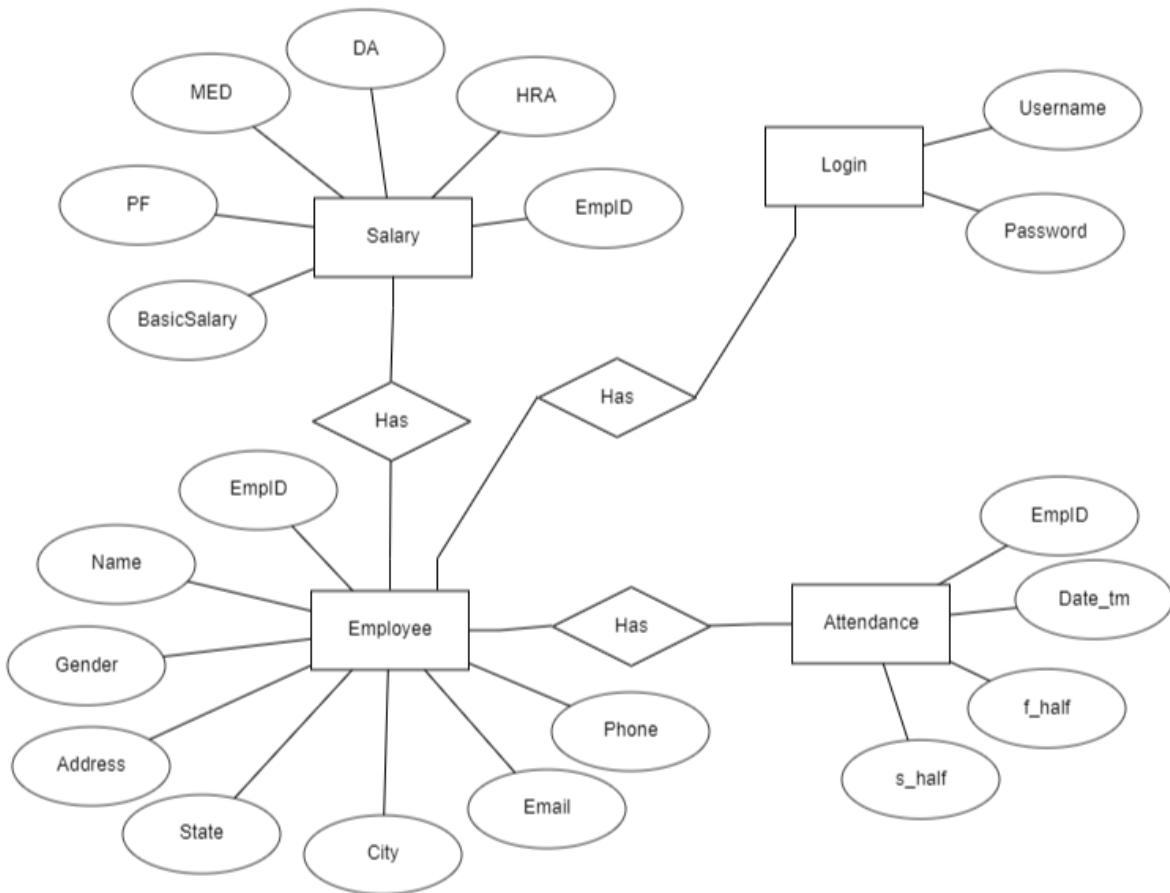
- Operating System              Windows 10
- Front End                      Java, NetBeans
- Back End                      MySQL

### Hardware Requirements

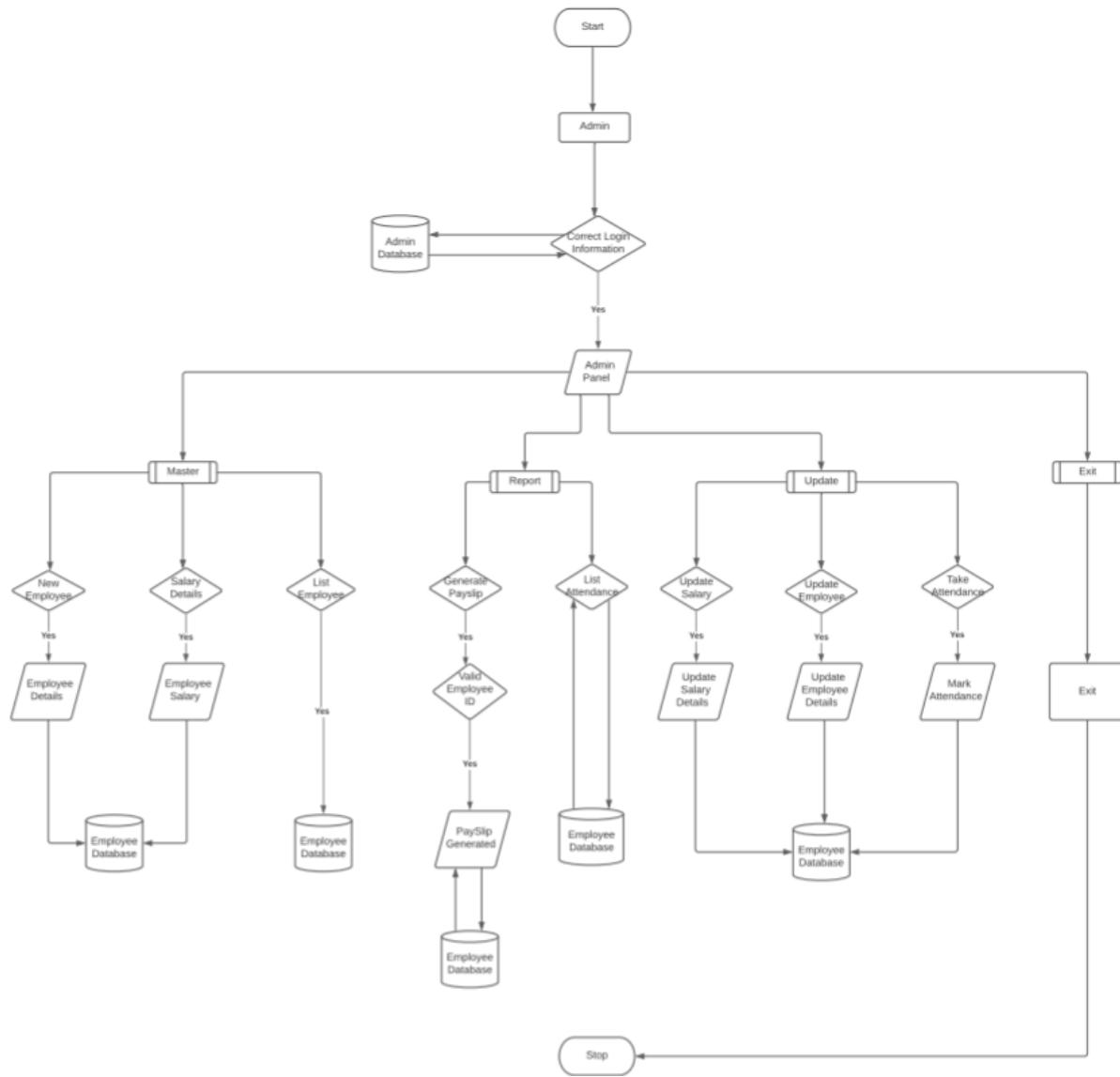
- Desktop PC or a Laptop
- Printer
- Operating System – Windows 10
- Intel® CoreTM i3-6006U CPU @ 2.00GHz
- 4.00 GB RAM
- 64-bit operating system, x64 based processor
- 1024 x 768 monitor resolution
- Keyboard and Mouse

---

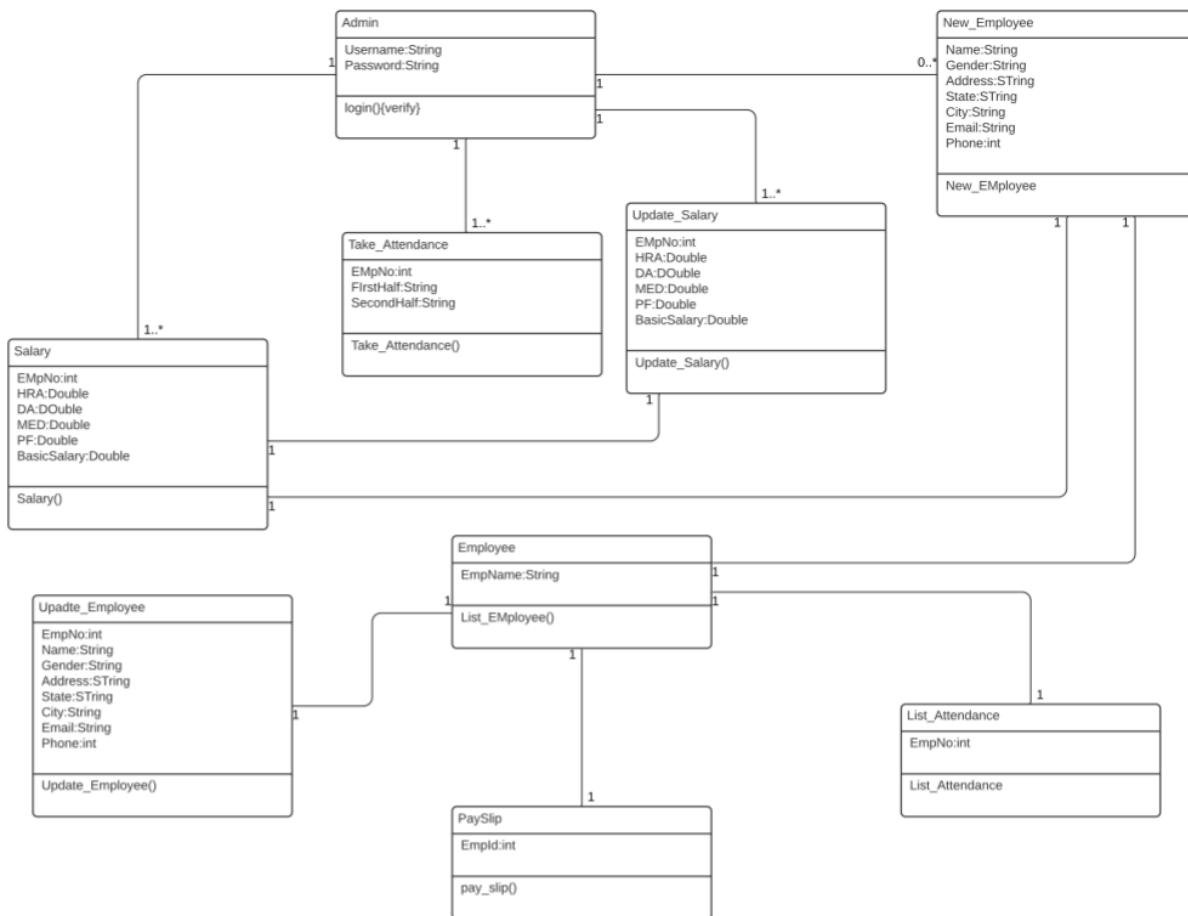
## ER Diagram



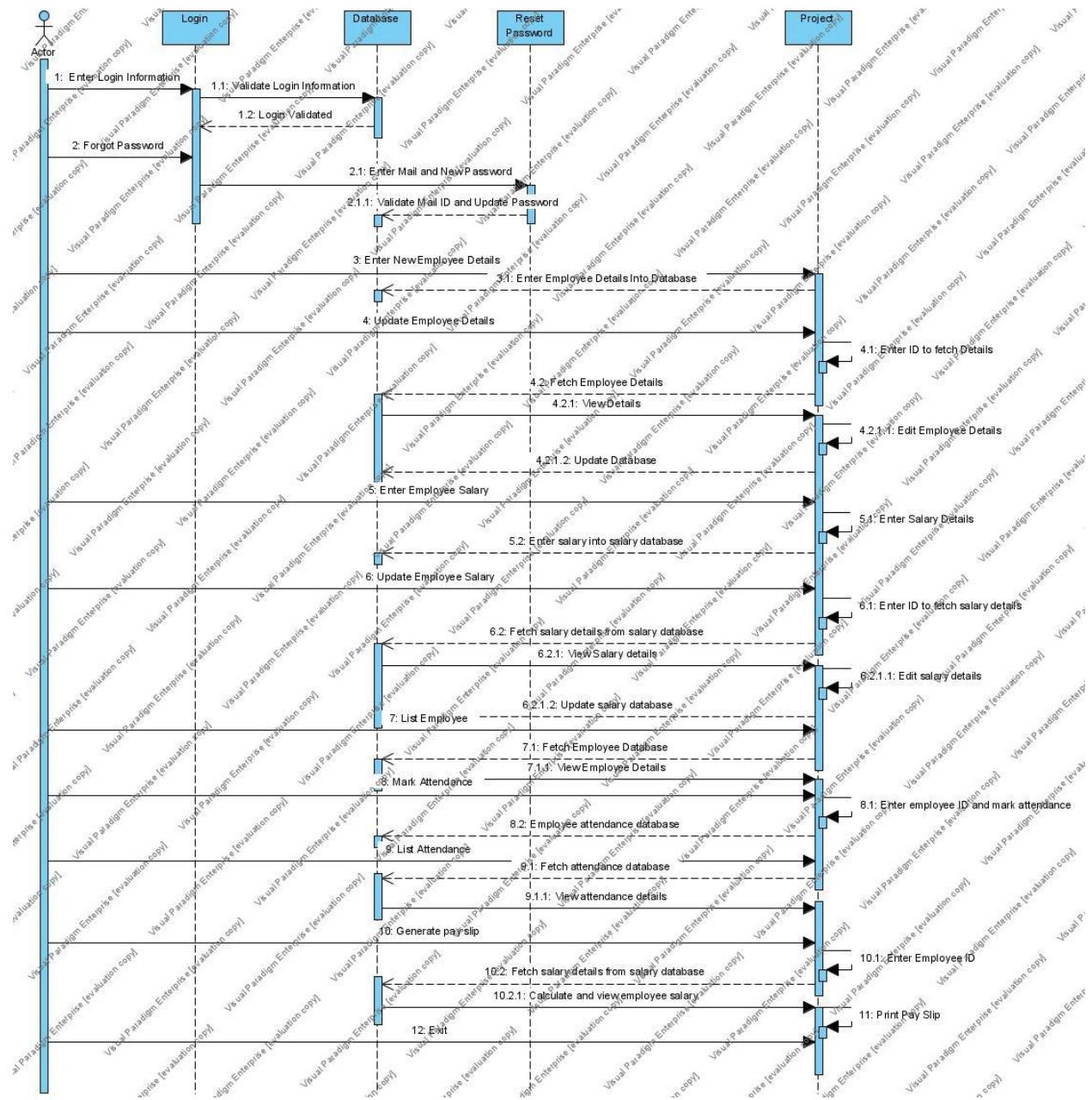
## FlowChart



## Class Diagram



## Sequence Diagram



---

## **CODE**

### **Connection**

```
package pocketmuster;
import java.sql.*;

public class conn {
    public Connection conn;
    public Statement s;
    public PreparedStatement pstmt;
    public conn() {
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            String connectionurl="jdbc:mysql://localhost:3306/pocketmuster?
user=root&password=9029059791";
            conn=DriverManager.getConnection(connectionurl);
            System.out.println("Connected");
            s = conn.createStatement();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

---

### **Login**

```
package pocketmuster;

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

public class login extends JFrame implements ActionListener{
JLabel l1,l2;
JTextField t1;
JPasswordField t2;
JButton b1,b2,b3;

login(){
super("Login Page");
setLayout(new BorderLayout());
t2 = new JPasswordField(10);
t1 = new JTextField(10);
JLabel l = new JLabel(new
ImageIcon(ClassLoader.getSystemResource("icon\\defaultpic.png")));
b1 = new JButton("Submit", new ImageIcon
(ClassLoader.getSystemResource("icon\\login.png")));
b2 = new JButton("Cancel", new ImageIcon
(ClassLoader.getSystemResource("icon\\Cancel.png")));
b3 = new JButton("Forgot password");

b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);

JPanel p1,p2,p3,p4;
p1=new JPanel();
p2=new JPanel();
```

---

```
p3=new JPanel();
p4=new JPanel();

add(l,BorderLayout.WEST);

p2.add(new JLabel("User Name   "));
p2.add(t1);
p2.add(new JLabel("Password "));
p2.add(t2);
add(p2,BorderLayout.CENTER);

p4.add(b1);
p4.add(b2);
p4.add(b3);

add(p4,BorderLayout.SOUTH);
setSize(400,250);
setLocation(600,400);
setVisible(true);
}

@Override
public void actionPerformed(ActionEvent ae) {
JButton b=(JButton)ae.getSource();
try
{
conn c1=new conn();
String u=t1.getText();
String v= t2.getText();
String q="select * from login where username='"+u+"' ";
ResultSet rs=c1.s.executeQuery(q); // query execute
if(rs.next()){
new project().setVisible(true);
setVisible(false);
}
else if(b==b3){
new password().setVisible(true);
setVisible(false);
}
else{
JOptionPane.showMessageDialog(null,"Invalid login");
setVisible(false);
}
}
```

---

```
}

catch(Exception e) {
e.printStackTrace();
}

}

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



---

### **Forgot password**

```
package pocketmuster;

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

public class password extends JFrame implements ActionListener{
JLabel l1,l2,l3;
JButton b1,b2;
JTextField t1;
JPasswordField t2,t3;
password()
{
super("Password");
setLayout(new BorderLayout());

t1 = new JTextField(20);
t2 = new JPasswordField(20);
t3 = new JPasswordField(20);

b1 = new JButton("Submit");
b2 = new JButton("Cancel");

b1.addActionListener(this);
b2.addActionListener(this);

JPanel p1,p2;

p1 = new JPanel();
p2 = new JPanel();

p1.add(new JLabel("Enter Your Email "));
p1.add(t1);
p1.add(new JLabel("Enter New Password "));
p1.add(t2);
p1.add(new JLabel("Re-Enter Password "));
```

---

```
p1.add(t3);

p2.add(b1);
p2.add(b2);

add(p1,BorderLayout.CENTER);
add(p2,BorderLayout.SOUTH);

setSize(400,250);
setLocation(600,400);
setVisible(true);
}

@Override
public void actionPerformed(ActionEvent ae) {
JButton b=(JButton)ae.getSource();
try{
conn c1=new conn();
String m=t1.getText();
String query="Select email from login";
ResultSet rs=c1.s.executeQuery(query);

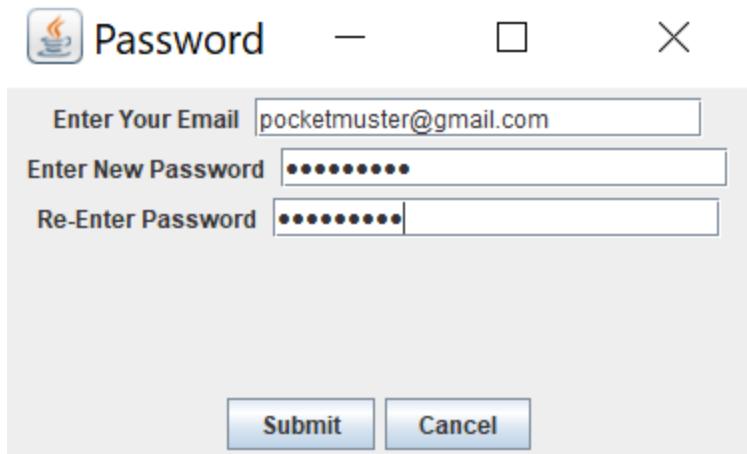
if(rs.next()){
System.out.println("Email matches");
String u=t2.getText();
String v=t3.getText();

if(u.equals(v))
{
System.out.println("Password matches");
String q="Update login set password='"+u+"' where username='admin'";
c1.s.executeUpdate(q);
System.out.println("Updated");
 JOptionPane.showMessageDialog(null,"Password Updated");
new login().setVisible(true);
setVisible(false);
}
else
{
JOptionPane.showMessageDialog(null,"Password Does Not Match");
}
}
else{
```

```
JOptionPane.showMessageDialog(null,"Invalid Mail");
setVisible(false);
}
}

catch(Exception e) {
}
}

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



---

### New employee

```
package pocketmuster;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class New_Employee extends JFrame implements ActionListener{
JLabel l1,l2,l3,l4,l5,l6,l7;
JTextField t1,t2,t3,t4,t5,t6,t7;
JButton b1,b2;
Choice c1;
New_Employee(){
super("New Employee");
setSize(600,650);
setLocation(600,200);
getContentPane().setBackground(Color.BLACK);

JPanel p1= new JPanel();
p1.setBackground(Color.BLACK);
p1.setLayout(new GridLayout(8,2,10,40));

l1 = new JLabel("Name");
l1.setForeground(Color.WHITE);

t1 = new JTextField(15);
t1.setBackground(Color.BLACK);
t1.setForeground(Color.WHITE);

p1.add(l1);
p1.add(t1);

c1 = new Choice();
c1.add("Male");
c1.add("Female");
c1.add("Others");
c1.setBackground(Color.BLACK);
c1.setForeground(Color.WHITE);

l2 = new JLabel("Gender");
```

---

```
12.setForeground(Color.WHITE);

p1.add(l2);
p1.add(c1);
l3 = new JLabel("Address");
l3.setForeground(Color.WHITE);

t3 = new JTextField(15);
t3.setBackground(Color.BLACK);
t3.setForeground(Color.WHITE);
p1.add(l3);
p1.add(t3);
l4 = new JLabel("State");
l4.setForeground(Color.WHITE);

t4 = new JTextField(15);
t4.setBackground(Color.BLACK);
t4.setForeground(Color.WHITE);

p1.add(l4);
p1.add(t4);
l5 = new JLabel("City");
l5.setForeground(Color.WHITE);

t5 = new JTextField(15);
t5.setBackground(Color.BLACK);
t5.setForeground(Color.WHITE);
p1.add(l5);
p1.add(t5);

l6 = new JLabel("Email");
l6.setForeground(Color.WHITE);
t6 = new JTextField(15);
t6.setBackground(Color.BLACK);
t6.setForeground(Color.WHITE);

p1.add(l6);
p1.add(t6);
l7 = new JLabel("Phone");
l7.setForeground(Color.WHITE);

t7= new JTextField(15);
t7.setBackground(Color.BLACK);
```

---

---

```
t7.setForeground(Color.WHITE);

p1.add(l7);
p1.add(t7);
b1 =new JButton("Submit");
b2 = new JButton("Cancel");
p1.add(b1);
p1.add(b2);

setLayout(new BorderLayout());
add(new JLabel(new ImageIcon
(ClassLoader.getSystemResource("icons/new_employee.png"))),"West");
add(p1,"Center");

b1.addActionListener(this);
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
b2.addActionListener(this);
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);
}

@Override
public void actionPerformed(ActionEvent ae) {

String n = t1.getText();
String g = c1.getSelectedItem();
String a = t3.getText();
String s = t4.getText();
String c = t5.getText();
String e = t6.getText();
String p = t7.getText();
String qry = "insert into employee
values(null,'" +n+"','"+g+"','"+a+"','"+s+"','"+c+"','"+e+"','"+p+"' )"
;
try{
conn c1 = new conn();
c1.s.executeUpdate(qry);
JOptionPane.showMessageDialog(null,"Employee Created");
this.setVisible(false);
}
(Exception ee){
ee.printStackTrace();
}
```

---

```
}

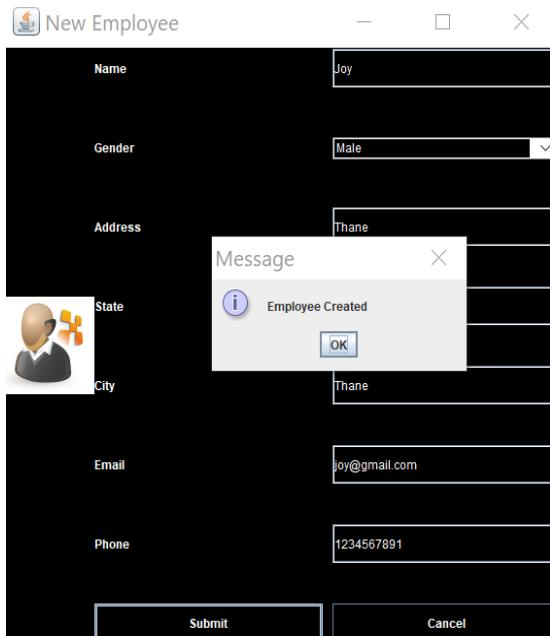
}

public static void main(String s[]){
new New_Employee().setVisible(true);
}
}
```

New Employee

---

Name	Joy
Gender	Male
Address	Thane
State	Maharashtra
City	Thane
Email	joy@gmail.com
Phone	1234567891
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>	



### Update employee

```
package pocketmuster;

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

public class Update_employee extends JFrame implements
ActionListener, ItemListener{
JLabel l1,l2,l3,l4,l5,l6,l7,emp;
JTextField t1,t2,t3,t4,t5,t6,t7;
JButton b1,b2;
Choice c1,c2;

Update_employee() {

super("Update Employee");

setLayout(null);
getContentPane().setBackground(Color.WHITE);

c2 = new Choice();
c2.setForeground(Color.WHITE);
```

---

```
c2.setBackground(Color.BLACK);
c2.setBounds(160,40,200,20);

try{
conn c = new conn();
ResultSet rs = c.s.executeQuery("select * from employee");

while(rs.next()){
c2.add(rs.getString("id"));
}
} catch(Exception e) { }

emp = new JLabel("Select Empno");
emp.setForeground(Color.WHITE);
emp.setBackground(Color.BLACK);
emp.setBounds(40,40,100,20);
add(emp);
add(c2);

l1 = new JLabel("Name : ");
l1.setForeground(Color.WHITE);
l1.setBackground(Color.BLACK);

t1 = new JTextField(15);
t1.setForeground(Color.WHITE);
t1.setBackground(Color.BLACK);

l1.setBounds(40,80,100,20);
t1.setBounds(160,80,200,20);
add(l1);
add(t1);

c1 = new Choice();
c1.setForeground(Color.WHITE);
c1.setBackground(Color.BLACK);

c1.add("Male");
c1.add("Female");

l2 = new JLabel("Gender : ");
l2.setForeground(Color.WHITE);
l2.setBackground(Color.BLACK);
```

---

---

```
l2.setBounds(40,120,100,20);
c1.setBounds(160,120,200,20);
add(l2);
add(c1);

l3 = new JLabel("Address : ");
l3.setForeground(Color.WHITE);
l3.setBackground(Color.BLACK);

t3 = new JTextField(15);
t3.setForeground(Color.WHITE);
t3.setBackground(Color.BLACK);

l3.setBounds(40,160,100,20);
t3.setBounds(160,160,200,20);
add(l3);
add(t3);

l4 = new JLabel("State : ");
l4.setForeground(Color.WHITE);
l4.setBackground(Color.BLACK);
t4 = new JTextField(15);
t4.setForeground(Color.WHITE);
t4.setBackground(Color.BLACK);

l4.setBounds(40,200,100,20);
t4.setBounds(160,200,200,20);
add(l4);
add(t4);

l5 = new JLabel("City : ");
l5.setForeground(Color.WHITE);
l5.setBackground(Color.BLACK);

t5 = new JTextField(15);
t5.setForeground(Color.WHITE);
t5.setBackground(Color.BLACK);

l5.setBounds(40,240,100,20);
t5.setBounds(160,240,200,20);
add(l5);
add(t5);
```

---

---

```
l6 = new JLabel("Email : ");
l6.setForeground(Color.WHITE);
l6.setBackground(Color.BLACK);

t6 = new JTextField(15);
t6.setForeground(Color.WHITE);
t6.setBackground(Color.BLACK);

l6.setBounds(40,280,100,20);
t6.setBounds(160,280,200,20);
add(l6);
add(t6);

l7 = new JLabel("Phone : ");
l7.setForeground(Color.WHITE);
l7.setBackground(Color.BLACK);

t7= new JTextField(15);
t7.setForeground(Color.WHITE);
t7.setBackground(Color.BLACK);

l7.setBounds(40,320,100,20);
t7.setBounds(160,320,200,20);
add(l7);
add(t7);

b1 =new JButton("Update");
b2 = new JButton("Delete");

b1.setBounds(40,400,150,30);
b2.setBounds(200,400,150,30);
add(b1);
add(b2);

b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);

b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);

b1.addActionListener(this);
b2.addActionListener(this);
```

---

---

```
c2.addItemListener(this);

setVisible(true);
setSize(400,550);
 setLocation(600,200);

getContentPane().setBackground(Color.BLACK);
}

public void actionPerformed(ActionEvent ae) {
if(ae.getSource()==b1){String n = t1.getText();
String g = c1.getSelectedItem();
String a = t3.getText();
String s = t4.getText();
String c = t5.getText();String e = t6.getText();
String p = t7.getText();
String qry = "update employee set
name='"+n+"',gender='"+g+"',address='"+a+"',state='"+s+"',city='"+c+"'
',email='"+e+"',phone='"+p+"' where id="+c2.getSelectedItem();

try{
conn c1 = new conn();
c1.s.executeUpdate(qry);
JOptionPane.showMessageDialog(null,"Employee Updated");
}
catch(Exception ee){
ee.printStackTrace();
}
}

if(ae.getSource()==b2) {
try{
conn c1 = new conn();
c1.s.executeUpdate("delete from employee where
id="+c2.getSelectedItem());
JOptionPane.showMessageDialog(null,"Employee Deleted");
this.setVisible(false);
}catch(Exception ee){
ee.printStackTrace();
}
}
}
```

---

```
public void itemStateChanged(ItemEvent ie){  
try{  
conn c1 = new conn();  
ResultSet rs = c1.s.executeQuery("select * from employee where  
id='"+c2.getSelectedItem());  
  
if(rs.next()){  
t1.setText(rs.getString("name"));  
t3.setText(rs.getString("address"));  
t4.setText(rs.getString("state"));  
t5.setText(rs.getString("city"));  
t6.setText(rs.getString("email"));  
t7.setText(rs.getString("phone"));  
}  
}  
  
catch(Exception ee){  
ee.printStackTrace();  
}  
}  
  
public static void main(String[] args){  
Update_employee update_employee = new Update_employee();  
}  
}
```



Update...



Select Empno	<input type="text" value="5"/>
Name :	<input type="text" value="ABC"/>
Gender :	<input type="text" value="Male"/>
Address :	<input type="text" value="Thane"/>
State :	<input type="text" value="Maharashtra"/>
City :	<input type="text" value="Thane"/>
Email :	<input type="text" value="abc@gmail.com"/>
Phone :	<input type="text" value="1234567891"/>

**Update**

**Delete**

Update... — □ ×

Select Empno :

Name :

Gender :

Address :

State :

City :

Email :

Phone :

---

### Salary

```
package pocketmuster;

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

public class Salary extends JFrame implements ActionListener{
    JLabel l1,l2,l3,l4,l5,l6,l7,l8;
    JTextField t1,t3,t4,t5,t6,t7;
    JButton b1,b2;
    Choice c2;

    Salary(){
        super("Set Salary");

        setLayout(new GridLayout(8,2,20,20));
        c2 = new Choice();
        c2.setBackground(Color.BLACK);
        c2.setForeground(Color.WHITE);

        try{
            conn c = new conn();
            ResultSet rs = c.s.executeQuery("select * from
employee");

            while(rs.next()){
                c2.add(rs.getString("id"));
            }
        }catch(Exception e){ }

        l8=new JLabel("Select Empno");
        l8.setForeground(Color.WHITE);
        add(l8);
        add(c2);

        l1 = new JLabel("HRA");
        l1.setBackground(Color.BLACK);
```

---

```
l1.setForeground(Color.WHITE);

t1 = new JTextField(15);
t1.setBackground(Color.BLACK);
t1.setForeground(Color.WHITE);

add(l1);
add(t1);

l3 = new JLabel("DA");
l3.setBackground(Color.BLACK);
l3.setForeground(Color.WHITE);

t3 = new JTextField(15);
t3.setBackground(Color.BLACK);
t3.setForeground(Color.WHITE);

add(l3);
add(t3);

l4 = new JLabel("MED");
l4.setBackground(Color.BLACK);
l4.setForeground(Color.WHITE);

t4 = new JTextField(15);
t4.setBackground(Color.BLACK);
t4.setForeground(Color.WHITE);

add(l4);
add(t4);

l5 = new JLabel("PF");
l5.setBackground(Color.BLACK);
l5.setForeground(Color.WHITE);

t5 = new JTextField(15);
t5.setBackground(Color.BLACK);
t5.setForeground(Color.WHITE);

add(l5);
add(t5);
```

---

---

```
l6 = new JLabel("Basic Salary");
l6.setBackground(Color.BLACK);
l6.setForeground(Color.WHITE);

t6 = new JTextField(15);
t6.setBackground(Color.BLACK);
t6.setForeground(Color.WHITE);
add(l6);
add(t6);

b1 =new JButton("Submit");
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);

b2 = new JButton("Cancel");
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);

add(b1);
add(b2);

b1.addActionListener(this);
b2.addActionListener(this);

setSize(450,550);
setLocation(500,200);
setVisible(true);

getContentPane().setBackground(Color.BLACK);

}

@Override
public void actionPerformed(ActionEvent ae) {

    Double hra = Double.parseDouble(t1.getText());
    Double id = Double.parseDouble(c2.getSelectedItem());
    Double da = Double.parseDouble(t3.getText());
    Double med= Double.parseDouble(t4.getText());
    Double pf = Double.parseDouble(t5.getText());
```

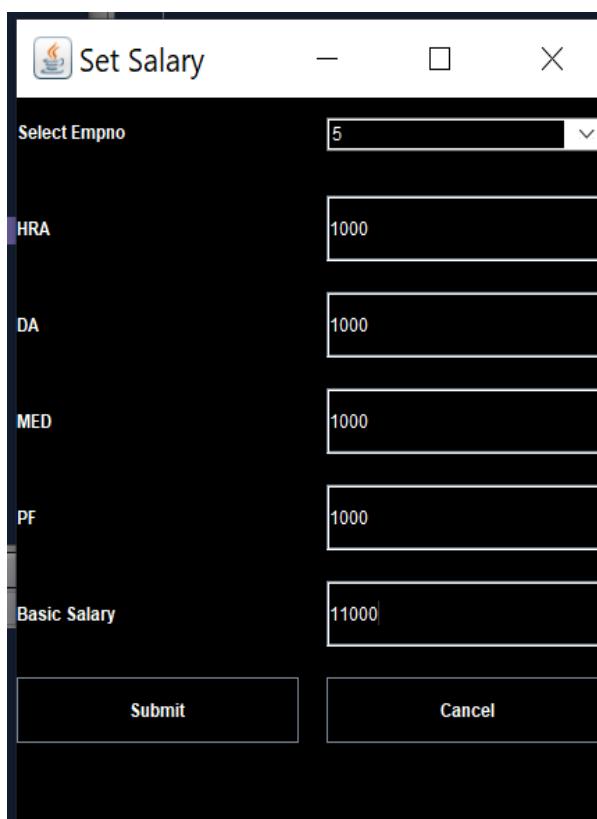
```

        Double basic = Double.parseDouble(t6.getText());
        String qry = "insert into salary values ('"+
id+"','"+hra+"','"+da+"','"+med+"','"+pf+"','"+basic+"')";

        try{
            conn c1 = new conn();
            c1.s.executeUpdate(qry);
            JOptionPane.showMessageDialog(null,"Salary updated");
            this.setVisible(false);
        }catch(Exception ee){
            ee.printStackTrace();
        }
    }

    public static void main(String[] args){
        new Salary().setVisible(true);
    }
}

```



Set Salary

Select Empno: 5

HRA: 1000

DA: 1000

MED:

PF:

Basic Salary: 11000

Submit Cancel

Message

Salary updated

OK

The screenshot shows a Java Swing application window titled "Set Salary". Inside the window, there are several input fields and buttons. At the bottom, there are "Submit" and "Cancel" buttons. A message dialog box titled "Message" is overlaid on the window, displaying the message "Salary updated" with an "OK" button.

---

### **Update salary**

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

public class Update_salary extends JFrame implements
ActionListener,ItemListener{
    JLabel l1,l2,l3,l4,l5,l6;
    JTextField t1,t2,t3,t4,t5,t6;
    JButton b1,b2;
    Choice c2;

    Update_salary(){
        setLayout(null);
        c2 = new Choice();
        try{
            conn c = new conn();
            ResultSet rs = c.s.executeQuery("select * from salary");
            while(rs.next()){
                c2.add(rs.getString("id"));
            }
        }catch(SQLException e){ }

        JLabel emp = new JLabel("Select Empno");
        emp.setBackground(Color.BLACK);
        emp.setForeground(Color.WHITE);

        emp.setBounds(20,20,100,20);
        add(emp);

        c2.setBounds(120,20,200,20);
        c2.setBackground(Color.BLACK);
        c2.setForeground(Color.WHITE);

        add(c2);
```

---

```
l1 = new JLabel("Hra");
l1.setBackground(Color.BLACK);
l1.setForeground(Color.WHITE);

t1 = new JTextField(15);
t1.setBackground(Color.BLACK);
t1.setForeground(Color.WHITE);

l1.setBounds(20,60,100,20);
t1.setBounds(120,60,200,20);
add(l1);
add(t1);

l2 = new JLabel("Da");
l2.setBackground(Color.BLACK);
l2.setForeground(Color.WHITE);

t2 = new JTextField(15);
t2.setBackground(Color.BLACK);
t2.setForeground(Color.WHITE);

l2.setBounds(20,100,100,20);
t2.setBounds(120,100,200,20);
add(l2);
add(t2);

l3 = new JLabel("Med");
l3.setBackground(Color.BLACK);
l3.setForeground(Color.WHITE);

t3 = new JTextField(15);
t3.setBackground(Color.BLACK);
t3.setForeground(Color.WHITE);

l3.setBounds(20,140,100,20);
t3.setBounds(120,140,200,20);
add(l3);
add(t3);

l4 = new JLabel("Pf");
l4.setBackground(Color.BLACK);
l4.setForeground(Color.WHITE);
```

---

---

```
t4 = new JTextField(15);
t4.setBackground(Color.BLACK);
t4.setForeground(Color.WHITE);

l4.setBounds(20,180,100,20);
t4.setBounds(120,180,200,20);
add(l4);
add(t4);

l5 = new JLabel("basic_salary");
l5.setBackground(Color.BLACK);
l5.setForeground(Color.WHITE);

t5 = new JTextField(15);
t5.setBackground(Color.BLACK);
t5.setForeground(Color.WHITE);

l5.setBounds(20,220,100,20);
t5.setBounds(120,220,200,20);
add(l5);
add(t5);

b1 =new JButton("Update");
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);

b2 = new JButton("Delete");
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);

b1.setBounds(40,280,100,20);
b2.setBounds(200,280,100,20);
add(b1);
add(b2);

b1.addActionListener(this);
b2.addActionListener(this);
c2.addItemListener(this);

getContentPane().setBackground(Color.BLACK) ;

setVisible(true);
```

---

---

```
        setSize(400,450);
        setLocation(600,200);
    }

    @Override
    public void actionPerformed(ActionEvent ae) {

        if(ae.getSource()==b1){
            Double hra = Double.parseDouble(t1.getText());
            Double id = Double.parseDouble(c2.getSelectedItem());
            Double da = Double.parseDouble(t2.getText());
            Double med = Double.parseDouble(t3.getText());
            Double pf = Double.parseDouble(t4.getText());
            Double basic = Double.parseDouble(t5.getText());

            String qry = "update salary set
hra="+hra+",da="+da+",med="+med+",pf="+pf+",basic_salary="+basic+
where id="+c2.getSelectedItem();

            try{
                conn c1 = new conn();
                c1.s.executeUpdate(qry);
                JOptionPane.showMessageDialog(null,"Salary Updated");
                this.setVisible(false);
            }catch(Exception ee){
                ee.printStackTrace();
            }
        }

        if(ae.getSource()==b2){

            try{
                conn c1 = new conn();
                c1.s.executeUpdate("delete from salary where
id="+c2.getSelectedItem());
                JOptionPane.showMessageDialog(null,"Salary Deleted");
                this.setVisible(false);
            }catch(Exception ee){
                ee.printStackTrace();
            }
        }
    }
}
```

---

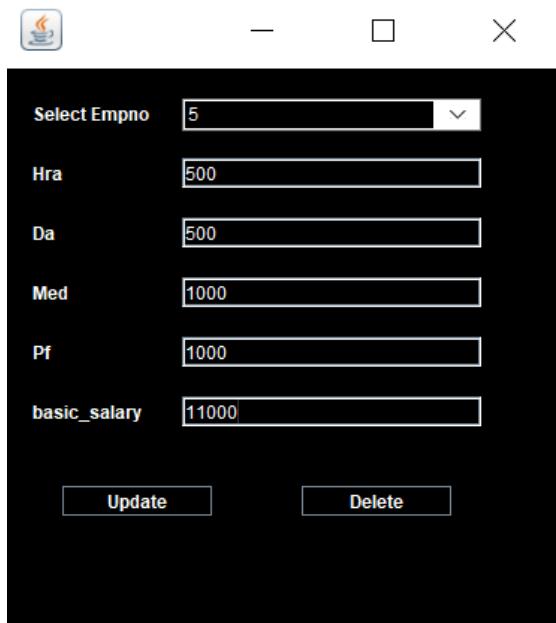
```
@Override
public void itemStateChanged(ItemEvent ie)
{
    try{
        conn c1 = new conn();
        ResultSet rs = c1.s.executeQuery("select * from salary
where id='"+c2.getSelectedItem());
        if(rs.next()){
            t1.setText(rs.getString("hra"));
            t2.setText(rs.getString("da"));
            t3.setText(rs.getString("med"));
            t4.setText(rs.getString("pf"));
            t5.setText(rs.getString("basic_salary"));

        }
    }catch(Exception ee){
        ee.printStackTrace();
    }
}

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

}
```

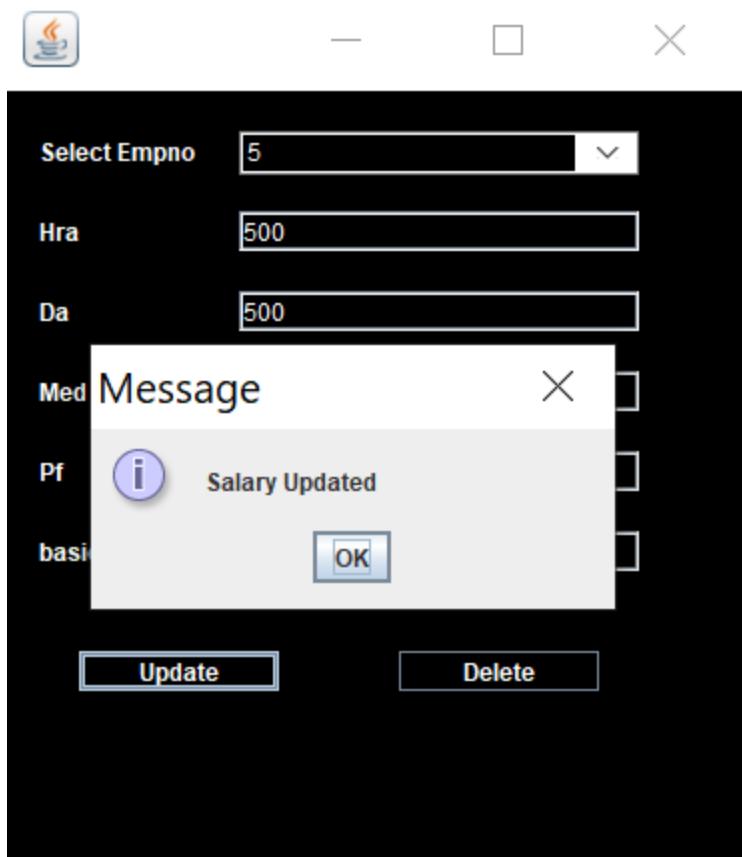
---



A screenshot of a Java Swing application window. The title bar has standard minimize, maximize, and close buttons. The main area contains a form with the following fields:

Select Empno	5
Hra	500
Da	500
Med	1000
Pf	1000
basic_salary	11000

At the bottom are two buttons: "Update" and "Delete".



A screenshot of the same Java Swing application window as above, but now with a message dialog box overlaid. The message dialog has a title "Message" and a message "Salary Updated" with an information icon. It also has an "OK" button.

---

### List employee

```
package pocketmuster;
import java.sql.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class List_Employee extends JFrame implements ActionListener{
    JTable j1;
    JButton b1;
    String h[]={ "Emp
id","Name","Gender","Address","State","City","Email id","Phone"};
    String d[][]=new String[20][8];
    int i=0,j=0;
    List_Employee(){
        super("View Employees");
        setSize(1000,400);
        setLocation(450,200);
        try{
            String q="select * from employee";
            conn c1=new conn();
            ResultSet rs=c1.s.executeQuery(q);
            while(rs.next()){
                // i = 0 j = 0
                d[i][j++]=rs.getString("id");
```

```
d[i][j++]=rs.getString("name");
d[i][j++]=rs.getString("gender");
d[i][j++]=rs.getString("address");
d[i][j++]=rs.getString("state");
d[i][j++]=rs.getString("city");
d[i][j++]=rs.getString("email");
d[i][j++]=rs.getString("phone");
i++;
j=0;
}
j1=new JTable(d,h);
}
catch(Exception e){}
b1=new JButton("Print");
add(b1,"South");
JScrollPane s1=new JScrollPane(j1);
add(s1);
b1.addActionListener(this);
}
public void actionPerformed(ActionEvent ae){
try{
j1.print();
}catch(Exception e){}
}
public static void main(String s[]){
new List_Employee().setVisible(true);
}
}
```

---

### **List attendance**

```
package pocketmuster;
import java.sql.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class ListAttendance extends JFrame implements ActionListener{
    JTable j1;
    JButton b1;
    String h[]={ "Emp id", "Date Time", "First Half", "Second Half"};
    String d[][]=new String[15][4];
    int i=0,j=0;
    ListAttendance() {
        super("View Employees Attendence");
        setSize(800,300);
        setLocation(450,150);
        try{
            String q="select * from attendence";
            conn c1=new conn();
            ResultSet rs=c1.s.executeQuery(q);
            while(rs.next()){
                d[i][j++]=rs.getString("id");
                d[i][j++]=rs.getString("date_tm");
                d[i][j++]=rs.getString("f_half");
                d[i][j++]=rs.getString("s_half");
            }
        }
    }
}
```

---

```
i++;
j=0;
j1=new JTable(d,h);
}catch(Exception e){}
b1=new JButton("Print");
add(b1,"South");
JScrollPane s1=new JScrollPane(j1);
add(s1);
b1.addActionListener(this);
public void actionPerformed(ActionEvent ae){
try{
j1.print();
}catch(Exception e){}
}
public static void main(String[] args){
new ListAttendance().setVisible(true);
}}
```

### **Take attendance**

```
package pocketmuster;

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

public class TakeAttendance extends JFrame implements ActionListener{
JLabel l1,l2,l3,l4,l5,l6,l7,l8;
JTextField t1,t2,t3,t4,t5,t6,t7;
JButton b1,b2;
Choice c2,fh,sh;

TakeAttendance () {

setLayout(new GridLayout(4,2,50,50));
c2 = new Choice();
try{
conn c = new conn();
ResultSet rs = c.s.executeQuery("select * from
employee");
while(rs.next()){
c2.add(rs.getString("id"));
}
}
```

```
}

} catch (Exception e) { }

l8=new JLabel("Select Empno");
l8.setForeground(Color.WHITE);
add(l8);

c2.setBackground(Color.BLACK);
c2.setForeground(Color.WHITE);

add(c2);

l1 = new JLabel("First Half");
l1.setForeground(Color.WHITE);

fh = new Choice();
fh.add("Present");
fh.add("Absent");
fh.add("Leave");
fh.setBackground(Color.BLACK);
fh.setForeground(Color.WHITE);

add(l1);
add(fh);

l2 = new JLabel("Second Half");
l2.setForeground(Color.WHITE);

sh = new Choice();
sh.add("Present");
sh.add("Absent");
sh.add("Leave");
sh.setBackground(Color.BLACK);
sh.setForeground(Color.WHITE);

add(l2);
add(sh);

b1 =new JButton("Submit");
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
```

---

```
b2 = new JButton("Cancel");
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);

add(b1);
add(b2);

b1.addActionListener(this);
b2.addActionListener(this);

getContentPane().setBackground(Color.BLACK);

setVisible(true);
setSize(400,450);
setLocation(600,200);

}

@Override
public void actionPerformed(ActionEvent ae) {

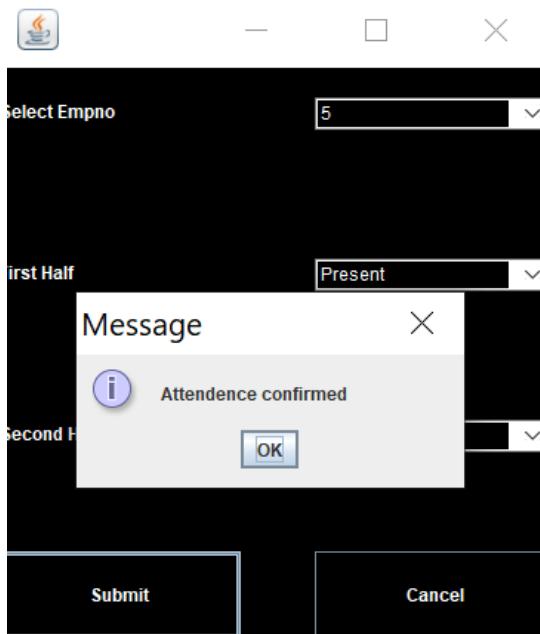
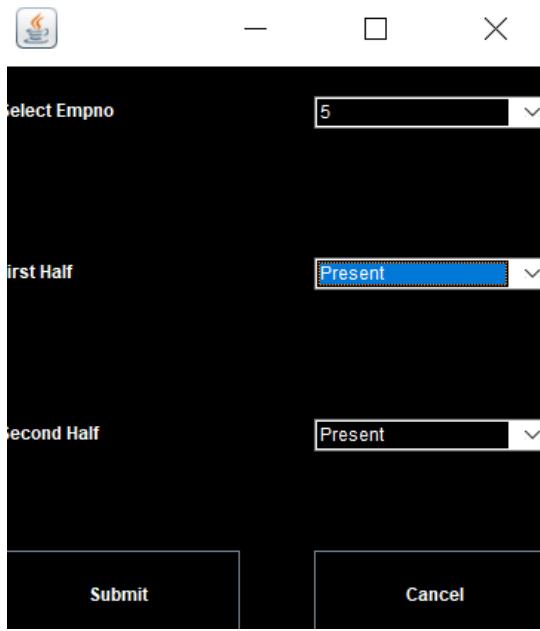
    String f = fh.getSelectedItem();
    String s = sh.getSelectedItem();
    String dt = new java.util.Date().toString();
    String id=c2.getSelectedItem();
    String qry = "insert into attendence values("+ id
+", '"+dt+"', '"+f+"', '"+s+"'')";

    try{
        conn c1 = new conn();
        c1.s.executeUpdate(qry);
        JOptionPane.showMessageDialog(null,"Attendence
confirmed");
        this.setVisible(false);
    }catch(Exception ee){
        ee.printStackTrace();
    }
}

public static void main(String s[]){
    new TakeAttendance();
}
```

---

```
}
```



---

### **Pay slip**

```
package pocketmuster;

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

public class pay_slip extends JFrame implements ActionListener{

    Choice c1;
    JTextArea t1;
    JButton b1;

    pay_slip(){

        setSize(800,700);
        setLocation(400,150);
        c1 = new Choice();
        try{
            conn c = new conn();
            ResultSet rs = c.s.executeQuery("select * from salary");
            while(rs.next()){
                c1.add(rs.getString("id"));
            }
        }catch(SQLException e) { }

        setLayout(new BorderLayout());

        JPanel p1 = new JPanel();
        p1.add(new JLabel("Select Id"));
        p1.add(c1);
        add(p1,"North");
        //c1.addItemListener(this);

        t1 = new JTextArea(30,80);
        JScrollPane jsp = new JScrollPane(t1);
```

---

```
Font f1 = new Font("arial",Font.BOLD,20);
t1.setFont(f1);

b1 = new JButton("Generate Pay Slip");

add(b1,"South");
add(jsp,"Center");
b1.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

try{
conn c = new conn();
System.out.println("Generating pay slip.....");

String empid=c1.getSelectedItem();

ResultSet rs = c.s.executeQuery("select * from employee
where id='"+empid';
rs.next();
String name = rs.getString("name");
rs.close();
/*
rs=c.s.executeQuery("Select *from attendence where
id='"+empid';
rs.next();
String n=rs.getString("f_half");
System.out.println(n);
*/
String n="Present";
String query="Select * from attendence where
id='"+empid+"' and f_half='Present'";
rs=c.s.executeQuery(query);
int count=0;
while(rs.next())
{
count++;
}
```

---

---

```
int nf_half=count;
rs.close();

rs=c.s.executeQuery("Select * from attendance where
id='"+empid+"' and s_half='Present'");
int count1=0;
while(rs.next())
{
count1++;
}

int ns_half=count1;
rs.close();

rs = c.s.executeQuery("select * from salary where
id='"+empid+"';
double gross=0.0;
double net=0.0;

java.util.Date d1 = new java.util.Date();
int month = d1.getMonth();
t1.setText(" ----- PAY SLIP FOR THE MONTH OF
"+month+" ,2019 -----");
t1.append("\n");

if(rs.next()){

t1.append("\n      Employee ID "+rs.getString("id"));
t1.append("\n      Employee Name "+name);

t1.append("\n-----");
t1.append("\n");

double hra = rs.getDouble("hra");
t1.append("\n                  HRA      : "+hra);
double da   = rs.getDouble("da");
t1.append("\n                  DA      : "+da);
double med  = rs.getDouble("med");
t1.append("\n                  MED     : "+med);
```

---

---

```

        double pf  = rs.getDouble("pf");
        t1.append("\n                  PF          : "+pf);
        double basic  = rs.getDouble("basic_salary");
        t1.append("\n                  Basic       : ");
        "+basic);

        double perday=basic/30;

        double f_half=perday/2;
        double s_half=perday/2;

        System.out.println(f_half);
        System.out.println(s_half);

        System.out.println(nf_half);
        System.out.println(ns_half);

        gross =
hra+da+med+pf+(nf_half*f_half)+(ns_half*s_half);
        net = gross - pf;

t1.append("\n-----");
        t1.append("\n");

        t1.append("\n                  GROSS SALARY :"+gross+"\n");
NET SALARY : "+net);
        t1.append("\n                  Tax      :    2.1% of gross "+(gross*2.1/100));
        t1.append("\n-----");
        t1.append("\n");
        t1.append("\n");
        t1.append("\n");
        t1.append("      (  Signature   )");

    }

} catch(SQLException ee) {
}

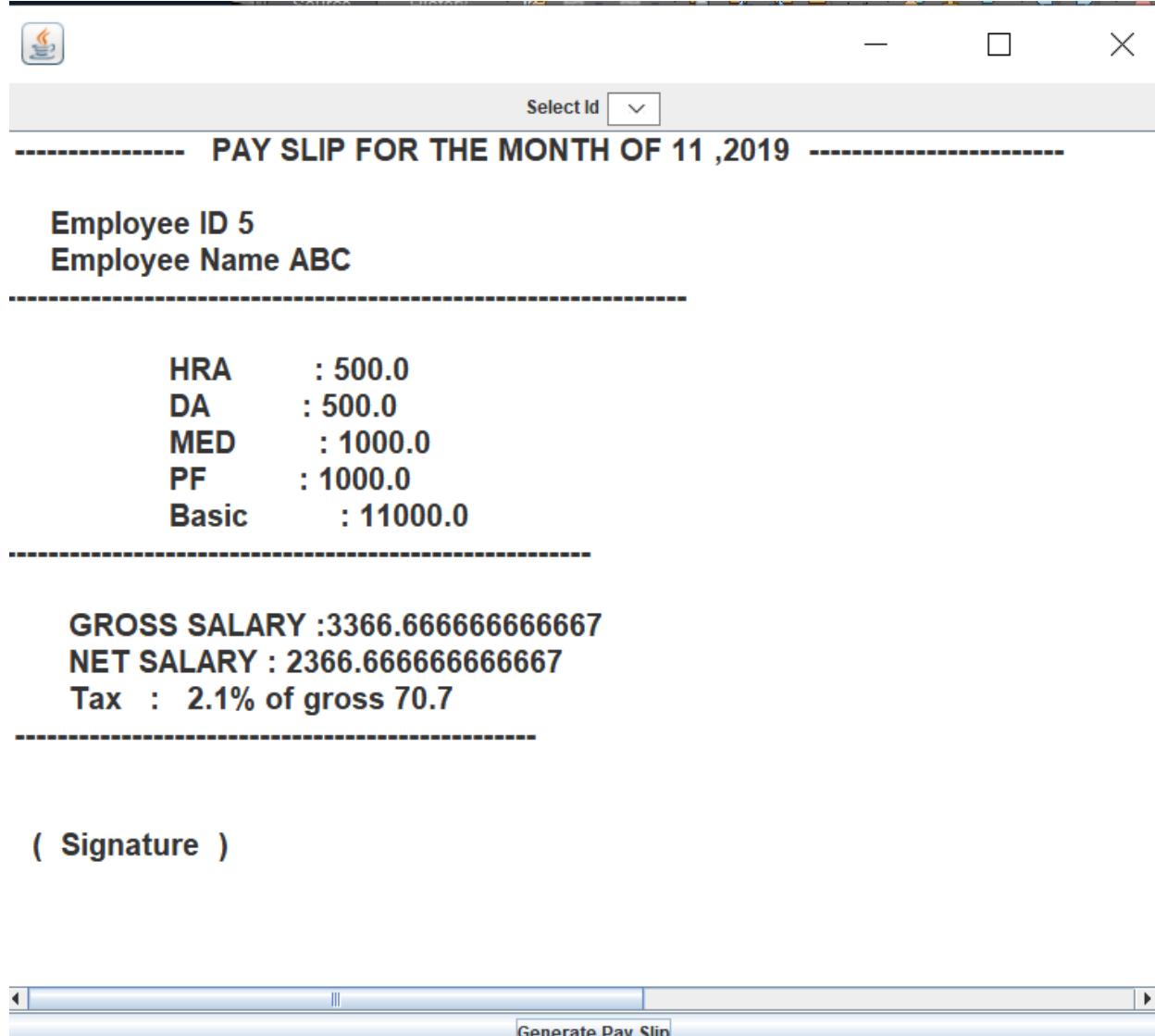
```

---

```
}

public static void main(String[] args){
    new pay_slip().setVisible(true);
}

}
```



---

## Project

```
package pocketmuster;

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

public class project extends JFrame implements ActionListener {

    project(){
        setSize(2000,1100);
        getContentPane().setBackground(Color.WHITE);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("icon/payroll.jpg"));
        Image i2 =
i1.getImage().getScaledInstance(1900,1000,Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l1 = new JLabel(i3);
        add(l1);

        JMenuBar mb = new JMenuBar();
        setJMenuBar(mb);
        JMenu m1 = new JMenu("Master");
        m1.setForeground(Color.blue);

        JMenuItem t1 = new JMenuItem("New Employee");

        t1.setForeground(Color.blue);
        t1.setFont(new Font("monospaced",Font.PLAIN,12));
        t1.setMnemonic('N');
        t1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N,
ActionEvent.CTRL_MASK));
        t1.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icons/New.png")));

        JMenuItem t3 = new JMenuItem("Salary");
        t3.setForeground(Color.blue);
```

---

```
t3.setFont(new Font("monospaced",Font.PLAIN,12));
t3.setMnemonic('S');
t3.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icon/schedreport.png")));
t3.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_S,
ActionEvent.CTRL_MASK));

JMenuItem t4 = new JMenuItem("List Employee");

t4.setForeground(Color.blue);
t4.setFont(new Font("monospaced",Font.PLAIN,12));
t4.setMnemonic('L');
t4.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icon/newinvoice.png")));
t4.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_L,
ActionEvent.CTRL_MASK));

m1.add(t1);
// m1.add(t2);
m1.add(t3);
m1.add(t4);
mb.add(m1);

t1.addActionListener(this);
//t2.addActionListener(this);
t3.addActionListener(this);
t4.addActionListener(this);

JMenu edit =new JMenu("Update");
edit.setForeground(Color.RED);

mb.add(edit);
JMenuItem s1 = new JMenuItem("Update Salary");
s1.setForeground(Color.blue);
s1.setFont(new Font("monospaced",Font.PLAIN,12));
s1.setMnemonic('U');
s1.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icons/EditOpen.png")));
```

---

---

```
s1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_U,
ActionEvent.CTRL_MASK));

edit.add(s1);

JMenuItem s2 = new JMenuItem("Update Employee");

s2.setForeground(Color.blue);
s2.setFont(new Font("monospaced",Font.PLAIN,12));
s2.setMnemonic('p');
s2.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icon/empreport.png")));
s2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_P,
ActionEvent.CTRL_MASK));

edit.add(s2);

JMenuItem s3 = new JMenuItem("Take Attendence");

s3.setForeground(Color.blue);
s3.setFont(new Font("monospaced",Font.PLAIN,12));
s3.setMnemonic('T');
s3.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icon/EXPENSE.PNG")));
s3.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_T,
ActionEvent.CTRL_MASK));

edit.add(s3);

s1.addActionListener(this);
s2.addActionListener(this);
s3.addActionListener(this);
JMenu rep =new JMenu("Reports");
rep.setForeground(Color.blue);

mb.add(rep);
JMenuItem p1 = new JMenuItem("Generate PaySlip");

p1.setForeground(Color.blue);
p1.setFont(new Font("monospaced",Font.PLAIN,12));
p1.setMnemonic('P');
```

---

---

```
    p1.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icon/payments.png")));
    p1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_P,
ActionEvent.CTRL_MASK));

    rep.add(p1);
    JMenuItem p2 = new JMenuItem("List Attendance");

    p2.setForeground(Color.blue);
    p2.setFont(new Font("monospaced", Font.PLAIN, 12));
    p2.setMnemonic('L');
    p2.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icon/empreport.png")));
    p2.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_L,
ActionEvent.CTRL_MASK));

    rep.add(p2);
    p1.addActionListener(this);
    p2.addActionListener(this);

    JMenu m8=new JMenu("Exit");
    m8.setForeground(Color.red);
    mb.add(m8);
    JMenuItem m8i1=new JMenuItem("Exit");
    m8.add(m8i1);
    m8i1.setForeground((Color.blue));
    m8i1.setFont(new Font("monospaced", Font.PLAIN, 14));
    m8i1.setMnemonic('X');
    m8i1.setIcon(new
ImageIcon(ClassLoader.getSystemResource("icon/exit.PNG")));
    m8i1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X,
ActionEvent.CTRL_MASK));
    m8i1.addActionListener(this);

}

@Override
public void actionPerformed(ActionEvent ae) {
    String msg= ae.getActionCommand();

    if(msg.equals("New Employee"))
        new New_Employee().setVisible(true);
    else if(msg.equals("List Employee"))
```

---

```

        new List_Employee().setVisible(true);
    else if(msg.equals("Update Employee"))
        new Update_employee().setVisible(true);
    else if(msg.equals("Salary"))
        new Salary().setVisible(true);
    else if(msg.equals("Update Salary"))
        new Update_salary().setVisible(true);
    else if(msg.equals("Take Attendance")){
        new TakeAttendance().setVisible(true);
    }
    else if(msg.equals("Exit"))
        System.exit(0);
    else if(msg.equals("Generate PaySlip"))
        new pay_slip().setVisible(true);
    else if(msg.equals("List Attendance"))
        new ListAttendance().setVisible(true);
}

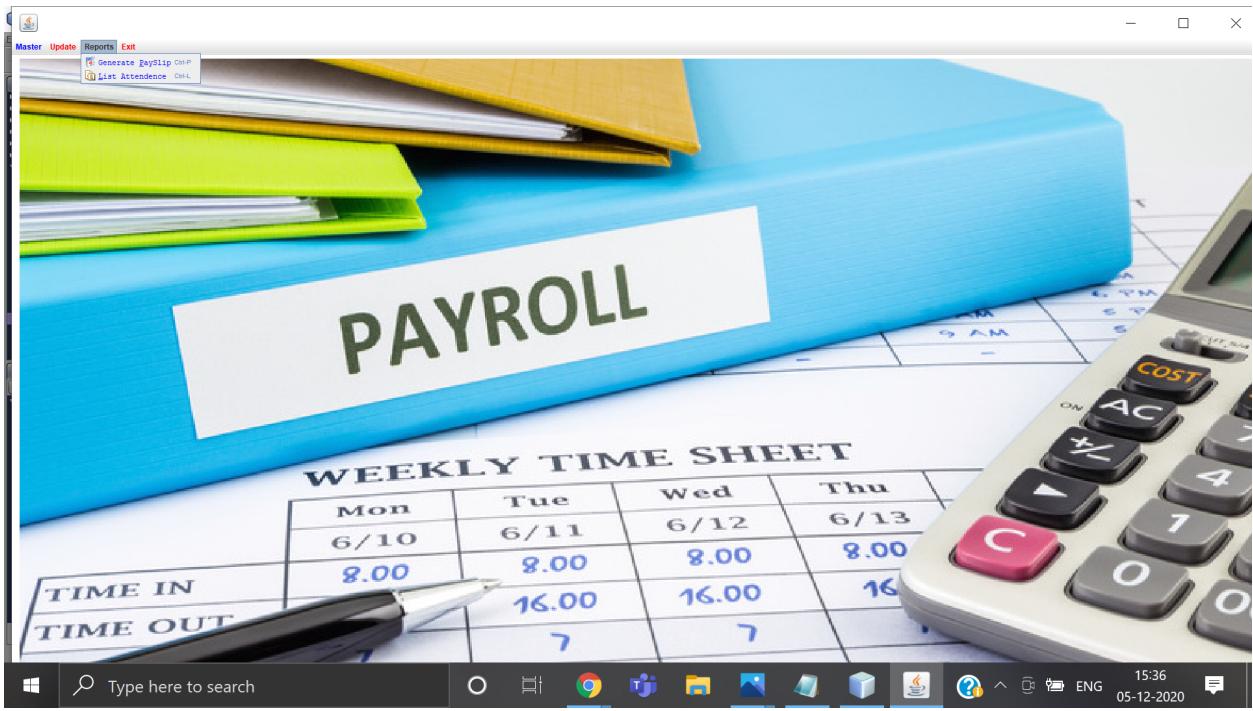
public static void main(String[] args){
    new project().setVisible(true);
}

}

```







---

## **Database PocketMuster**

### *Attendance Database*

```
CREATE TABLE attendence (
    int(10) DEFAULT NULL,
    date_tm varchar(30) DEFAULT NULL,
    f_half varchar(20) DEFAULT NULL,
    s_half varchar(30) DEFAULT NULL
);
```

### *Employee Database*

```
CREATE TABLE employee (
    id int(10) NOT NULL AUTO_INCREMENT,
    name varchar(30) DEFAULT NULL,
    gender varchar(30) DEFAULT NULL,
    address varchar(30) DEFAULT NULL,
    state varchar(30) DEFAULT NULL,
    city varchar(30) DEFAULT NULL,
    email varchar(30) DEFAULT NULL,
    phone varchar(30) DEFAULT NULL,
    PRIMARY KEY (`id`)
);
```

---

*Login Database*

```
CREATE TABLE login (
    username varchar(30) DEFAULT NULL,
    password varchar(30) DEFAULT NULL,
);
```

*Salary Database*

```
CREATE TABLE salary (
    id int(10) DEFAULT NULL,
    hra decimal(12,2) DEFAULT NULL,
    da decimal(12,2) DEFAULT NULL,
    med decimal(12,2) DEFAULT NULL,
    pf decimal(12,2) DEFAULT NULL,
    basic_salary decimal(12,2) DEFAULT NULL
);
```

```
MySQL 8.0 Command Line Client

mysql> use pocketmuster;
Database changed
mysql> Select *from login;
+-----+-----+-----+
| username | password | email
+-----+-----+-----+
| admin    | admin    | pocketmuster@gmail.com |
+-----+-----+-----+
1 row in set (1.98 sec)

mysql> Select *from employee;
+-----+-----+-----+-----+-----+-----+-----+
| id   | name  | gender | address | state  | city   | email
+-----+-----+-----+-----+-----+-----+-----+
| 5    | ABC   | Male   | Thane   | Maharashtra | Thane | abc@gmail.com | 1234567891
| 6    | Parth  | Male   | 101 Rose Apt | Maharashtra | Mulund | parth@yahoo.com | 2222222222
| 7    | Anvi   | Female | 304 sun apt | Maharashtra | Mulund | ak@gmail.com | 3541897265
| 8    | Sia    | Male   | 210 Rose apt | Maharashtra | Thane | sia@yahoo.com | 2547936814
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> Select *from salary;
+-----+-----+-----+-----+-----+
| id   | hra   | da    | med   | pf    | basic_salary |
+-----+-----+-----+-----+-----+
| 5    | 500.00 | 500.00 | 1000.00 | 1000.00 | 11000.00
| 6    | 400.00 | 600.00 | 700.00 | 200.00 | 18000.00
| 7    | 400.00 | 700.00 | 700.00 | 800.00 | 10000.00
| 8    | 700.00 | 500.00 | 500.00 | 400.00 | 12000.00
+-----+-----+-----+-----+-----+
4 rows in set (0.05 sec)

mysql> Select *from attendence;
```

```
MySQL 8.0 Command Line Client

+-----+-----+-----+-----+-----+-----+-----+
| id   | name  | gender | address | state  | city   | email
+-----+-----+-----+-----+-----+-----+-----+
| 5    | ABC   | Male   | Thane   | Maharashtra | Thane | abc@gmail.com | 1234567891
| 6    | Parth  | Male   | 101 Rose Apt | Maharashtra | Mulund | parth@yahoo.com | 2222222222
| 7    | Anvi   | Female | 304 sun apt | Maharashtra | Mulund | ak@gmail.com | 3541897265
| 8    | Sia    | Male   | 210 Rose apt | Maharashtra | Thane | sia@yahoo.com | 2547936814
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> Select *from salary;
+-----+-----+-----+-----+-----+
| id   | hra   | da    | med   | pf    | basic_salary |
+-----+-----+-----+-----+-----+
| 5    | 500.00 | 500.00 | 1000.00 | 1000.00 | 11000.00
| 6    | 400.00 | 600.00 | 700.00 | 200.00 | 18000.00
| 7    | 400.00 | 700.00 | 700.00 | 800.00 | 10000.00
| 8    | 700.00 | 500.00 | 500.00 | 400.00 | 12000.00
+-----+-----+-----+-----+-----+
4 rows in set (0.05 sec)

mysql> Select *from attendence;
+-----+-----+-----+-----+
| id   | date_tm          | f_half | s_half |
+-----+-----+-----+-----+
| 5    | Sat Dec 05 00:37:55 IST 2020 | Present | Present |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

---

## **Testing**

### Unit Testing

Unit testing is a testing technique in which modules are tested individually. Small individual units of source code are tested to determine whether it is fit to use or not. Different modules of games are put to test while the modules are being developed. Here modules refer to individual levels, players, scenes.

### Integration Testing

Integration testing is the technique in which individual components or modules are grouped together and tested. It occurs after testing. The input for the integrated testing, are the modules that have already been unit tested.

### System Testing

System testing is conducted on the entire system as a whole to check whether the system meets its requirements or not. 'Valar Morghulis' was installed on different systems and any errors or bugs that occurred were fixed.

### Acceptance Testing

User Acceptance is defined as a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This testing happens in the final phase of testing before moving the software application to the Market or Production environment.

---

## **Future Scope**

As small as well as large scale industries are coming more into action with it comes the effort to maintain employee details and generate their payslip. These industrial people have employees work on an hourly basis as per their project requirement.

Pocket Muster gives you a very easy way to calculate your employees salary based on the number of hours they have worked. It reduces paperwork as everything you won't have to write down the entry and exit time of an employee. Also saves you the hassle of sitting for hours to calculate an employee's salary.

---

## **REFERENCES**

[www.youtube.com](http://www.youtube.com)

[www.google.com](http://www.google.com)