

PAYROLL GENERATING SYSTEM

A PROJECT REPORT SUBMITTED BY

TEAM -G2

N.G.R ABITHA (913121104004)

N.AFFRIN FOWMIYA (913121104005)

C.NANDHINI (913121104066)

S.SWETHA (913121104111)

in partial fulfilment for the award of the degree
of

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING**



**VELAMMAL COLLEGE OF ENGINEERING AND
TECHNOLOGY**

(AUTONOMOUS)

MADURAI-625009

DECEMBER-2022

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	1
1.	INTRODUCTION	
	1.1 BACKGROUND	2
	1.2 MOTIVATION	
	1.3 OBJECTIVES	
2.	ABOUT THE SYSTEM	3
	3.1 EXISTING SYSTEMS	
	3.2 NEED FOR NEW SYSTEM	
	3.3 PROPOSED SYSTEM	
3.	HARDWARE AND SOFTWARE REQUIREMENTS	3
4.	MODULE DESCRIPTION	4-9
	5.1 COMPANY	
	6.2 EMPLOYEE DETAILS	
	6.3 SALARY GENERATION	
	6.4 PAYROLL DETAILS	
5.	SAMPLE CODING	10-17
6.	APPENDICES	18-24
7.	CONCLUSION	25
8.	REFERENCE	26

ABSTRACT:

This System is meant to supply the power to line up all the tasks of employee payment. At first, the user has got to undergo login system to urge access, then the user can add, list, update, and take away the employee's record , This system deals with the financial aspects of employee's Salary, Deductions, allowances, Net pay.

The user can view the account of each and every employee's payment is displayed which includes: Name with deduction, overtime, bonus and Net pay. This system makes easier to the user for managing payroll system as it is not time-consuming. This project is not difficult to operate and understood by the users

The "Payroll Management System" is designed to automate the existing manual system using computerized equipment and cutting-edge computer software, meeting client's needs so that their valuable data and information can be stored for a longer period with easy access and manipulation. The necessary software is readily available and simple to use. This software allows users to keep track of and see computerized records without having to make duplicate entries. The project explains how to handle user data for optimal efficiency and better customer service.

INTRODUCTION

BACKGROUND:

The proposed project “Employee Database and Payroll Management System” has been developed to overcome the problems faced in the practicing of manual system. This software is built to eliminate and in some cases reduce the hardships faced by the existing system. Moreover this system is designed for particular need of the company to carry out its operation in smooth and effective manner.

MOTIVATION:

After studying the data base connectivity in java and GUI makes us to generate report automatically by using salary database .In this report just we want to give information of employee ID and it will generate the salary of the employee.

OBJECTIVE:

The objective of this document is to describe the functionality and specifications of the design of a web application for Managing Employees and their payroll. The expected audiences of this document are the developers and the admin of the web application. Now with the help of this system the admin has the information on his finger tips and can easily prepare a good record based on their requirements. Finally, we can say that this system will not only automate the process but save the valuable time of the manager or the admin, which can be well utilized buy his institute. This will be an additional advantage and management of power based on their free time from his normal duty.

ABOUT THE SYSTEM

PROPOSED SYSTEM:

While generating salary for the employee instead of calculating the amount reduced or amount incremented, this proposed system provides an easy method to generate the report by just getting the employee. This system uses a database for getting employee details and uses GUI for attractive application of GUI.

NEED FOR NEW SYSTEM

This system deals with the financial aspects of employee's Salary, Deductions, allowances, Net pay. The user can view the account of each and every employee's and update their payments, and the user can also manage deductions, modify overtime and salary rate.

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENT

- ➔ System : Pentium i3 Processor(minimum)
- ➔ Hard Disk : 500 GB.
- ➔ Monitor : 15'' LED
- ➔ Input Devices : Keyboard, Mouse System : Pentium i3 Processor(minimum)
- ➔ Ram : 2 GB

SOFTWARE REQUIREMENT

- ➔ Operating system : Windows 10.
- ➔ Coding Language : JAVA(SERVLET)
- ➔ Tool : Netbeans 8.2
- ➔ Database : JAVA DERBY
- ➔ Jar folder: itextpdf

MODULE DESCRIPTION

MODULES USED IN THE PROJECT ARE:

- ➔ **COMPANY**
- ➔ **EMPLOYEE_DETAILS**
- ➔ **SALARY GENERATION**
- ➔ **PAYROLL_DETAILS**

COMPANY MODULE :

*This module shows the portal for SAAN company's Employee details for Payroll.

*This module allows the user to view the details of SAAN company employees

*It contains a show button

*Whenever the user click the Show button it opens the new window to display the details of SAAN company employees.

➔



EMPLOYEE_DETAILS:

->In this particular module, the click button enables the user to view the complete details of SAAN company employees.

->The table contains 5 columns named as :

*Emp_ID

*Emp_name

*Leave_days

*Parttime_days

*Salary

->This module gets the details of the employees from the connected database.

->This module also contains a button for payment of salaries

->Whenever the user clicks the pay button it opens the window for pay button.

→



The screenshot shows a software interface for employee details. At the top center is a button labeled "CLICK". Below this button is a table with five columns: "EMP_ID", "EMP_NAME", "LEAVE_DAYS", "PARTTIME_DAYS", and "SALARY". The table has several empty rows for data entry. At the bottom center is another button labeled "PAY EMPLOYEE".

EMP_ID	EMP_NAME	LEAVE_DAYS	PARTTIME_DAYS	SALARY

SALARY_ GENERATION:

->This module gets the employee ID from the user

->After receiving the Employee ID from the user,it checks the employee ID column of the database.

->It displays the name of the employee whose ID is given by the user.

->This module also gives two information about the salary generation of employee

*salary for one day : 500

*salary for

parttime:250

SALARY

CALCULATION:

->Reduction amount=No.of leave days*500

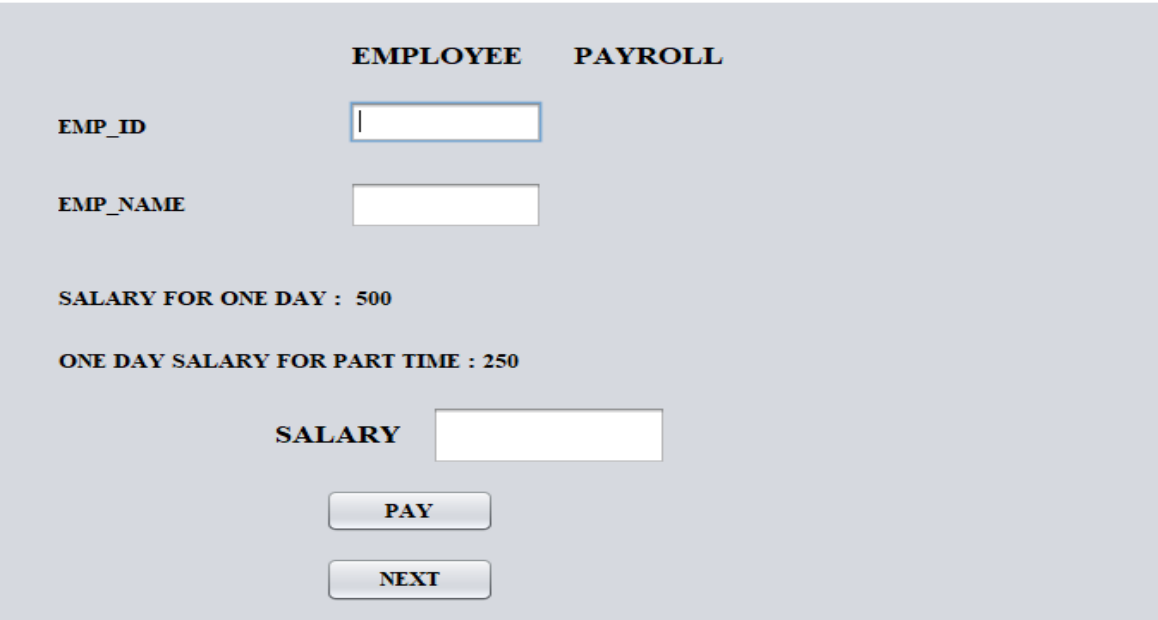
->Increment amount=No.of parttime days*250

->Now the salary is calculated by subtracting the reduction amount from the salary and adding the increment amount.

->By clicking the pay button ,it enables the user to view the salary and name of the employee whose ID is given by the user.

->It also contains the next button which shows the details of the salary generation.

→



The screenshot shows a window titled "EMPLOYEE PAYROLL" with a light blue background. It contains the following elements:

- Two input fields: "EMP_ID" and "EMP_NAME".
- Two lines of static text: "SALARY FOR ONE DAY : 500" and "ONE DAY SALARY FOR PART TIME : 250".
- A label "SALARY" followed by an input field.
- Two buttons: "PAY" and "NEXT".

PAYROLL_DETAILS:

->This module gets the employee ID from the user to display the details of the generated salary

->Whenever the user clicks the show details button it displays

*emp_ID

*emp_name

*No.of leave_days

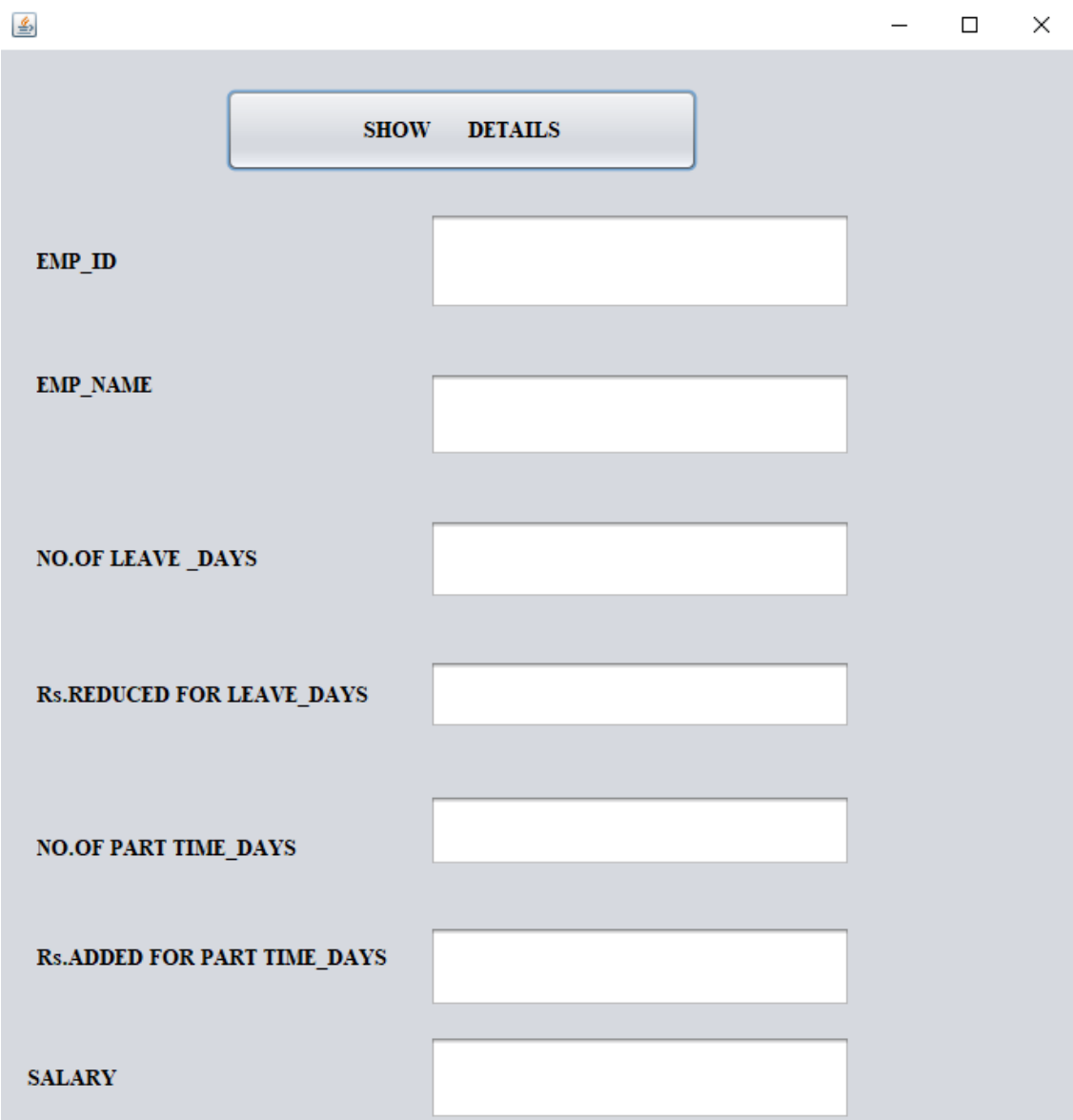
*Amount reduced for leave_days

*No.of parttime_days

*Incremented amount for parttime_days

*salary

→



EMP_ID	<input type="text"/>
EMP_NAME	<input type="text"/>
NO.OF LEAVE_DAYS	<input type="text"/>
Rs.REDUCED FOR LEAVE_DAYS	<input type="text"/>
NO.OF PART TIME_DAYS	<input type="text"/>
Rs.ADDED FOR PART TIME_DAYS	<input type="text"/>
SALARY	<input type="text"/>

CODE:

COMPANY:

```
public class main extends javax.swing.JFrame {

    public main() {
        initComponents();
        jButton1.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
        NOI18N
        jButton1.setText("SHOW");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling code here:
            main1 ob=new main1();
            ob.show();
        }
        public static void main(String args[]) {
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
                    new main().setVisible(true);
                }
            });
        }
    }
}
```

EMPLOYEE_DETAILS:

```
public class main1 extends javax.swing.JFrame {
    public main1() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
    Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jScrollPane2 = new javax.swing.JScrollPane();
        EMP = new javax.swing.JTable();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
        NOI18N
        jButton1.setText("CLICK");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```

public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton1ActionPerformed(evt);
}
});
jButton2.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jButton2.setText("PAY EMPLOYEE");
jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
}
});
EMP.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
{null, null, null, null, null},
{null, null, null, null, null},
{null, null, null, null, null},
{null, null, null, null, null},
{null, null, null, null, null}
},
new String [] {
"EMP_ID", "EMP_NAME", "LEAVE_DAYS",
"PARTTIME_DAYS", "SALARY"
}
));
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:
FINAL o=new FINAL();
o.show();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
Connection c;
try{
DefaultTableModel model=(DefaultTableModel) EMP.getModel();
model.setRowCount(0);
c=DriverManager.getConnection("jdbc:derby://localhost:1527/salary");
Statement st=c.createStatement();
ResultSet rs=st.executeQuery("select * from payroll");
while(rs.next())
{
String id=String.valueOf(rs.getInt("empid"));
String n=rs.getString("empname");
String l=String.valueOf(rs.getInt("leave_days"));
String p=String.valueOf(rs.getInt("parttime_days"));
String s=String.valueOf(rs.getInt("salary"));
String data[]={id,n,l,p,s};
model.addRow(data);
}
}
}

```

```

    }
    }catch(SQLException e){
    System.out.println(e);
    }
    }
    public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
    new main1().setVisible(true);
    }
    });
    }

```

SALARY_GENERATION:

```

public class FINAL extends javax.swing.JFrame {
    public FINAL() {
    initComponents();
    }
    @SuppressWarnings(&quot;unchecked&quot;);
    // &lt;editor-fold defaultstate=&quot;collapsed&quot; desc=&quot;Generated
    Code&quot;&gt;
    private void initComponents() {
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    jTextField2 = new javax.swing.JTextField();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jTextField3 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    jLabel1.setFont(new java.awt.Font(&quot;Times New Roman&quot;, 1, 18)); //
    NOI18N
    jLabel1.setText(&quot;EMPLOYEE PAYROLL&quot;);
    jLabel2.setFont(new java.awt.Font(&quot;Times New Roman&quot;, 1, 14)); //
    NOI18N
    jLabel2.setText(&quot;EMP_ID&quot;);
    jLabel3.setFont(new java.awt.Font(&quot;Times New Roman&quot;, 1, 14)); //
    NOI18N
    jLabel3.setText(&quot;EMP_NAME&quot;);
    jTextField2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
    jTextField2ActionPerformed(evt);
    }
    }

```

```

});
jLabel4.setFont(new java.awt.Font(&quot;Times New Roman&quot;, 1, 14)); //
NOI18N
jLabel4.setText(&quot;SALARY FOR ONE DAY : 500&quot;);

jLabel5.setFont(new java.awt.Font(&quot;Times New Roman&quot;, 1, 14)); //
NOI18N
jLabel5.setText(&quot;ONE DAY SALARY FOR PART TIME : 250&quot;);
jLabel6.setFont(new java.awt.Font(&quot;Times New Roman&quot;, 1, 18)); //
NOI18N
jLabel6.setText(&quot;SALARY&quot;);
jButton1.setFont(new java.awt.Font(&quot;Times New Roman&quot;, 1, 14)); //
NOI18N
jButton1.setText(&quot;PAY&quot;);
jButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton1ActionPerformed(evt);
}
});
jButton2.setFont(new java.awt.Font(&quot;Times New Roman&quot;, 1, 14)); //
NOI18N
jButton2.setText(&quot;NEXT&quot;);
jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
}
});
pack();
} // &lt;/editor-fold&gt;
private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
String s1=jTextField1.getText();
int a=Integer.parseInt(s1);
int l = 0,p=0,s=0,leave,parttime,i=0,n;
String q;
try{
Connection c
=DriverManager.getConnection(&quot;jdbc:derby://localhost:1527/salary&quot;);
Statement st=c.createStatement();
ResultSet rs=st.executeQuery(&quot;select
empid,leave_days,parttime_days,salary,empname from payroll&quot;);
while(rs.next()){

i=rs.getInt(1);
l=rs.getInt(2);
p=rs.getInt(3);

```

```

s=rs.getInt(4);
q=rs.getString(5);
if(i==a)
{
leave=l*500;
parttime=p*250;
s=s+parttime;
s=s-leave;
String r=String.valueOf(s);
jTextField3.setText(r);
jTextField2.setText(q);
System.out.println("&quot;the detail of employee paid is &quot;);
System.out.println("&quot;NAME : &quot;+q);
System.out.println("&quot;EMPLOYEE_ID : &quot;+i);
System.out.println("&quot;SALARY : &quot;+r);
//System.out.println(r);
}
}
}
catch(Exception ex){
System.out.println(ex);
}
}

```

PAYROLL_DETAILS:

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class user1 extends javax.swing.JFrame {
public user1() {
 initComponents();
}
@SuppressWarnings("&quot;unchecked&quot;")
// &lt;editor-fold defaultstate=&quot;collapsed&quot; desc=&quot;Generated
Code&quot;&gt;
private void initComponents() {

jPanel1 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jTextField2 = new javax.swing.JTextField();
jTextField4 = new javax.swing.JTextField();
jTextField5 = new javax.swing.JTextField();

```

```

jButton1 = new javax.swing.JButton();
jTextField6 = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
jTextField3 = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
jTextField7 = new javax.swing.JTextField();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jLabel1.setText("EMP_ID");
jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jLabel2.setText("EMP_NAME");
jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jLabel3.setText("NO.OF LEAVE _DAYS");
jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jLabel4.setText("NO.OF PART TIME_DAYS");
jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jLabel5.setText("SALARY");
jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jButton1.setText("SHOW DETAILS");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jLabel6.setText("Rs.REDUCED FOR LEAVE_DAYS");
jLabel7.setFont(new java.awt.Font("Times New Roman", 1, 14)); //
NOI18N
jLabel7.setText("Rs.ADDED FOR PART TIME_DAYS");
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    .addComponent(jTextField6)
    .addComponent(jTextField4)
    .addComponent(jTextField2)
    .addComponent(jTextField1)
    .addComponent(jTextField3)

```

```

.addComponent(jTextField7))
pack();
} // </editor-fold>
private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
String s1=jTextField1.getText();
int a=Integer.parseInt(s1);
int l = 0,p=0,s=0,leave,parttime,i=0,n;
String u;
try{
Connection c
=DriverManager.getConnection(""jdbc:derby://localhost:1527/salary"");
Statement st=c.createStatement();
ResultSet rs=st.executeQuery(""select
empname,leave_days,parttime_days,salary,empid from payroll"");
while(rs.next()){

i=rs.getInt(5);
l=rs.getInt(2);
p=rs.getInt(3);
s=rs.getInt(4);
u=rs.getString(1);
if(i==a)
{
leave=l*500;
parttime=p*250;
s=s+parttime;
s=s-leave;
String r=String.valueOf(s);
jTextField5.setText(r);
String q=String.valueOf(l);
jTextField6.setText(q);
String t=String.valueOf(p);
jTextField4.setText(t);
jTextField2.setText(u);
String j=String.valueOf(leave);
jTextField3.setText(j);
String k=String.valueOf(parttime);
jTextField7.setText(k);
}
}
}
catch(Exception ex){
System.out.println(ex);
}
}

```



```

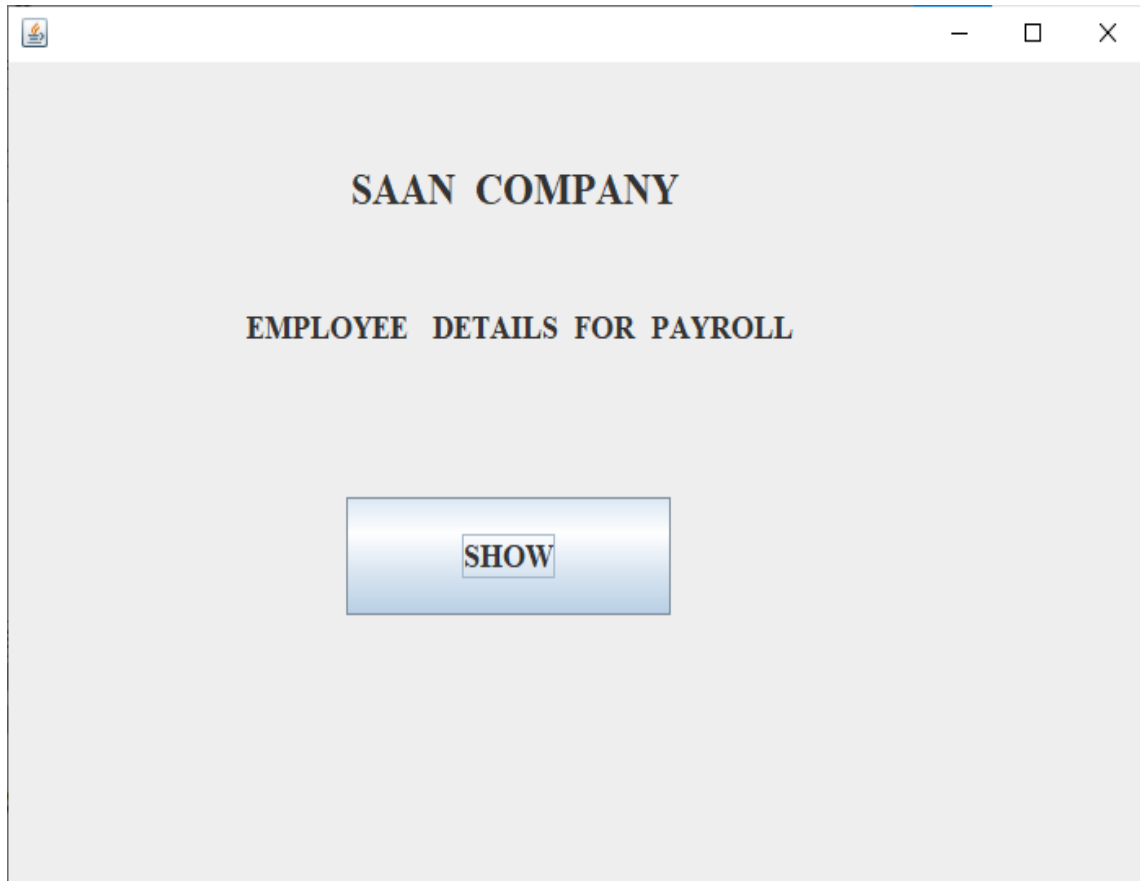
public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(user1.class.getName()).log(java.util.logging.Level
            .SEVERE, null,
            ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(user1.class.getName()).log(java.util.logging.Level
            .SEVERE, null,
            ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(user1.class.getName()).log(java.util.logging.Level
            .SEVERE, null,
            ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(user1.class.getName()).log(java.util.logging.Level
            .SEVERE, null,
            ex);
    }
}
//</editor-fold>
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new user1().setVisible(true);
    }
});
}

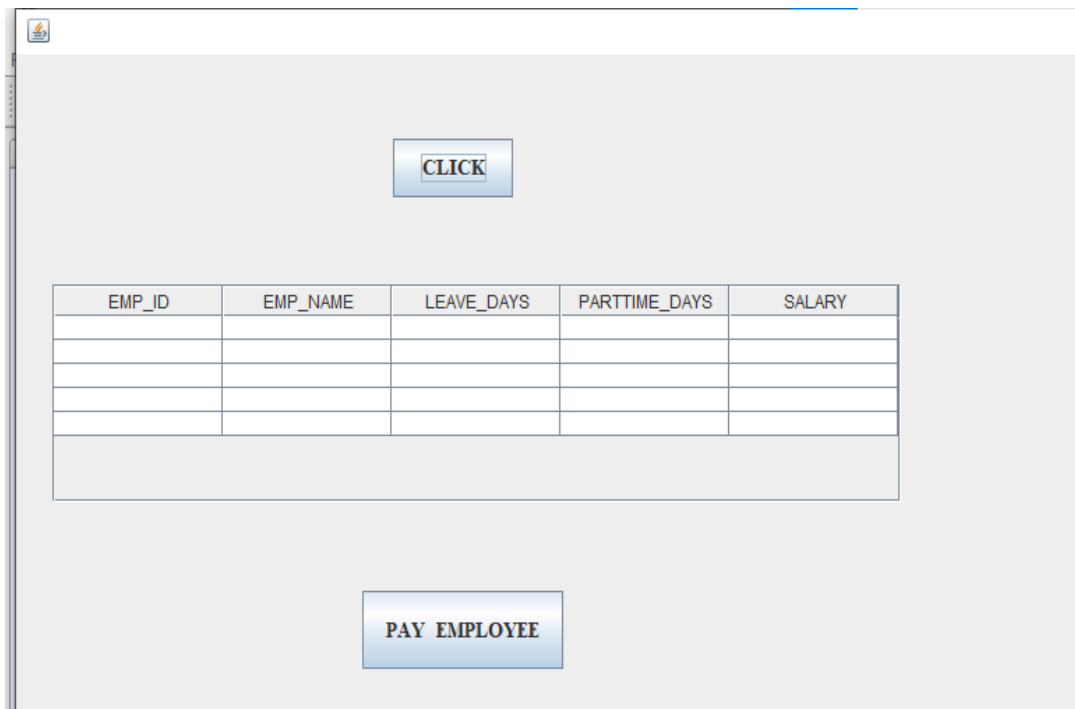
```

APPENDICES:

COMPANY PAGE:



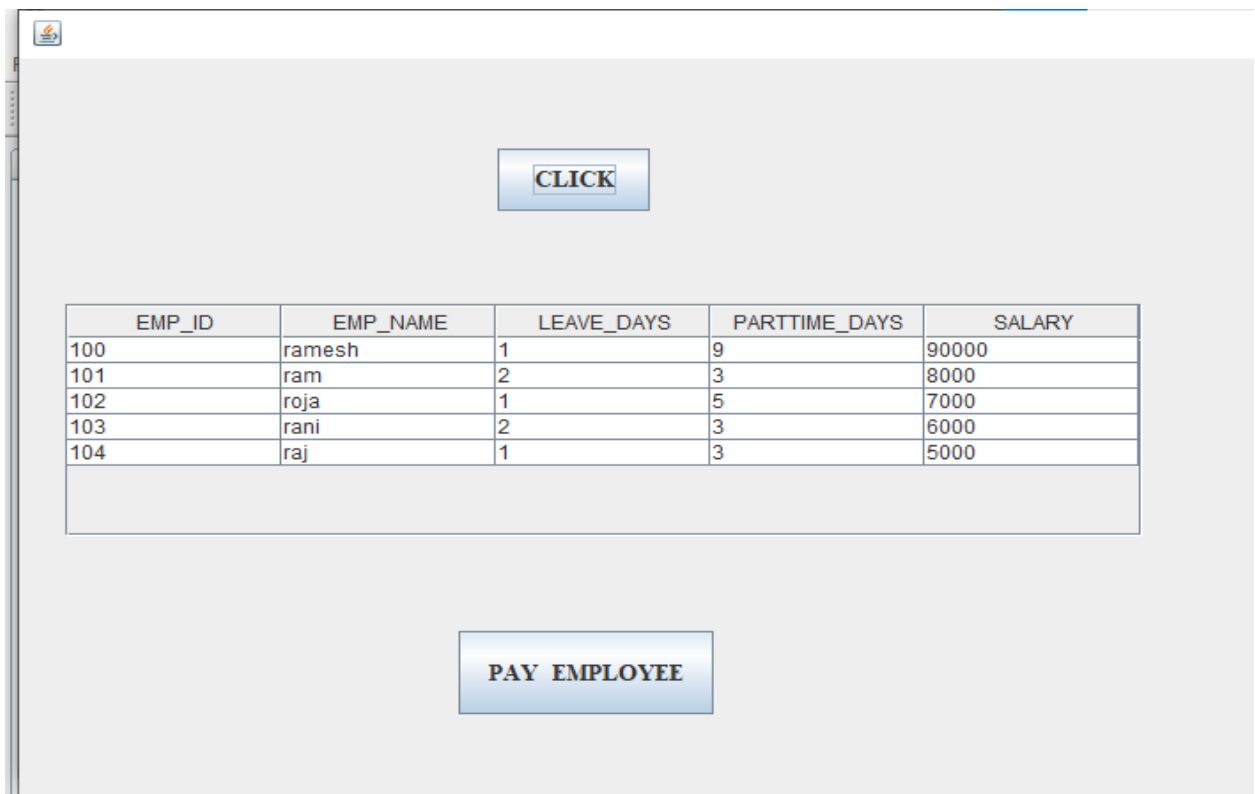
EMPLOYEE DETAILS PAGE:



A screenshot of a web application interface for employee details. It features a light gray background with a white table in the center. Above the table is a blue button with the text "CLICK". Below the table is another blue button with the text "PAY EMPLOYEE". The table has five columns: EMP_ID, EMP_NAME, LEAVE_DAYS, PARTTIME_DAYS, and SALARY. The table is currently empty, showing only header rows.

EMP_ID	EMP_NAME	LEAVE_DAYS	PARTTIME_DAYS	SALARY

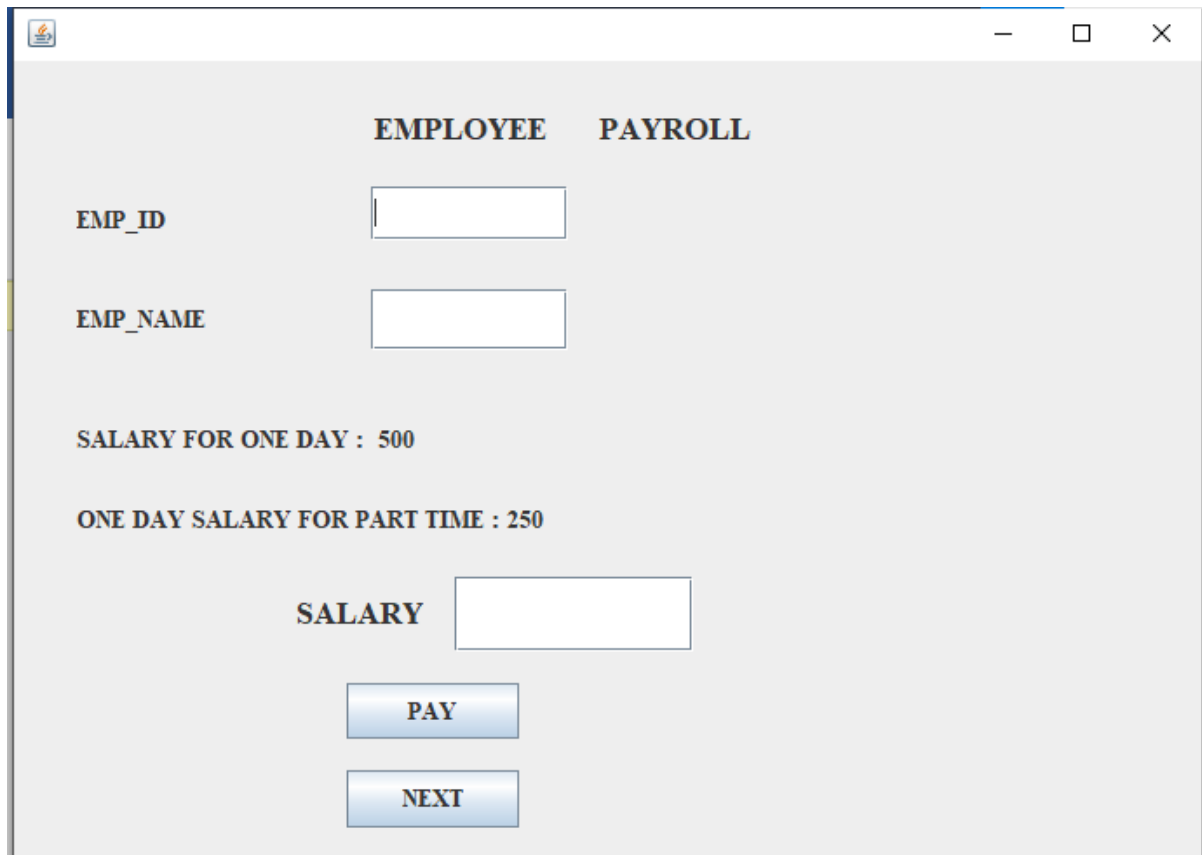
DETAILS OF EMPLOYEE PAGE:



A screenshot of a web application interface for employee details, showing populated data. It features a light gray background with a white table in the center. Above the table is a blue button with the text "CLICK". Below the table is another blue button with the text "PAY EMPLOYEE". The table has five columns: EMP_ID, EMP_NAME, LEAVE_DAYS, PARTTIME_DAYS, and SALARY. The table is populated with five rows of data.

EMP_ID	EMP_NAME	LEAVE_DAYS	PARTTIME_DAYS	SALARY
100	ramesh	1	9	90000
101	ram	2	3	8000
102	roja	1	5	7000
103	rani	2	3	6000
104	raj	1	3	5000

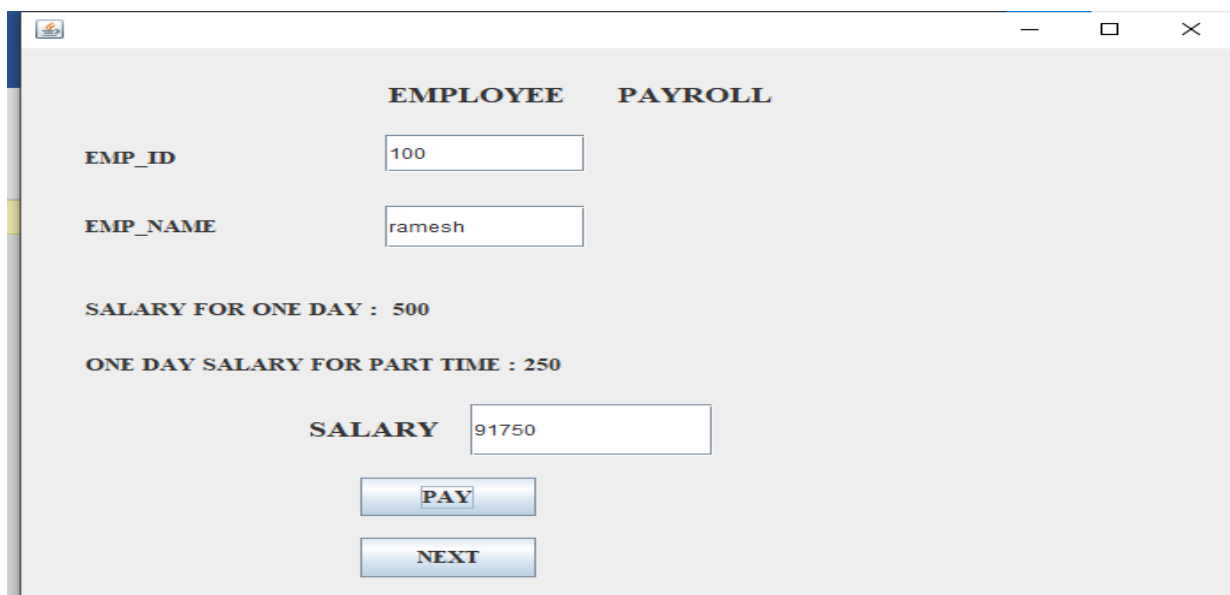
SALARY GENERATION PAGE:



A screenshot of a web application window titled "EMPLOYEE PAYROLL". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is light gray. At the top, the text "EMPLOYEE PAYROLL" is displayed in a bold, black, serif font. Below this, there are two input fields: "EMP_ID" and "EMP_NAME". The "EMP_ID" field is empty, and the "EMP_NAME" field is also empty. Below these fields, there are two lines of text: "SALARY FOR ONE DAY : 500" and "ONE DAY SALARY FOR PART TIME : 250". Below this text, there is a label "SALARY" followed by an empty input field. At the bottom of the form, there are two buttons: "PAY" and "NEXT".

EMPLOYEE PAYROLL	
EMP_ID	<input type="text"/>
EMP_NAME	<input type="text"/>
SALARY FOR ONE DAY : 500	
ONE DAY SALARY FOR PART TIME : 250	
SALARY	<input type="text"/>
<input type="button" value="PAY"/>	
<input type="button" value="NEXT"/>	

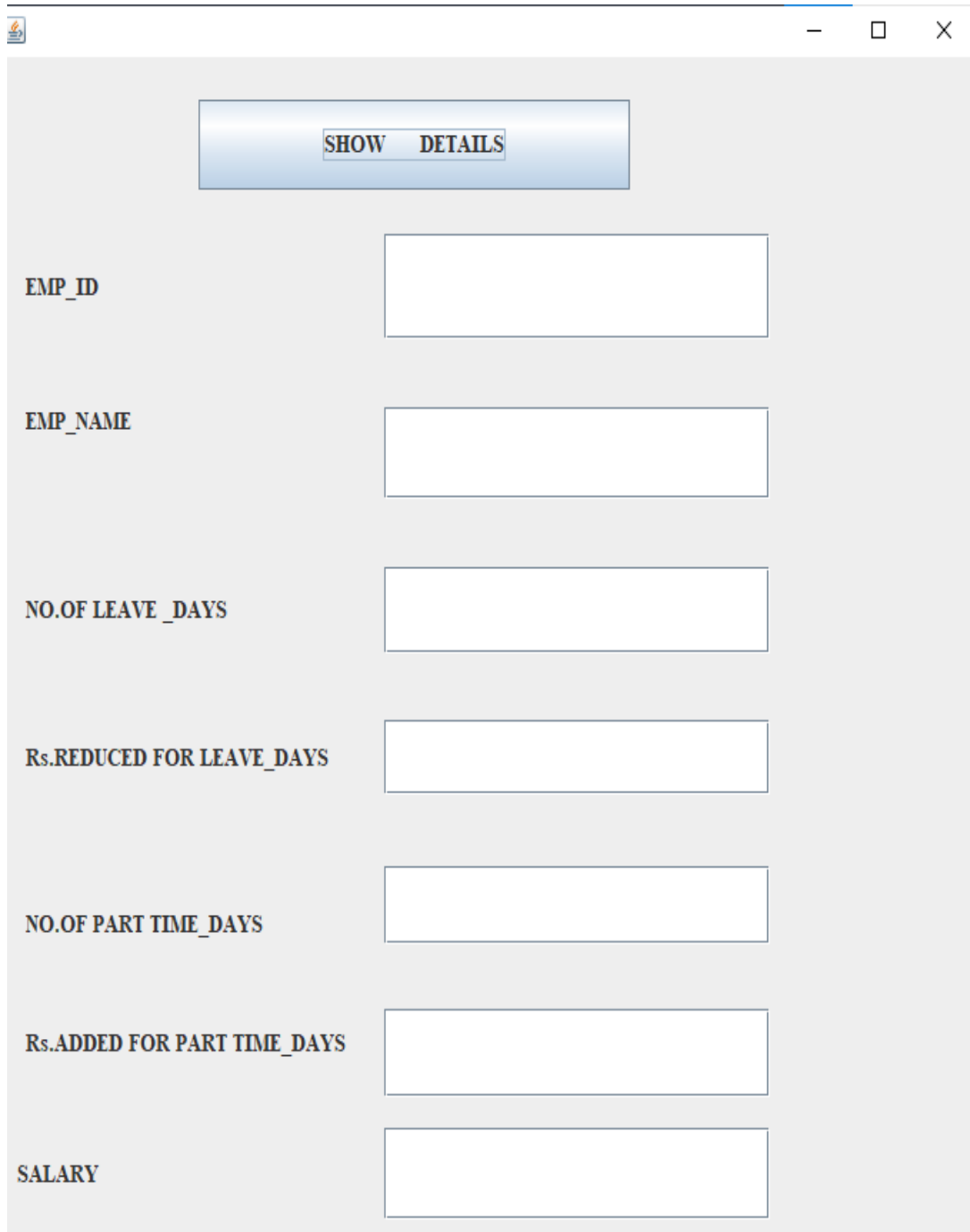
SALARY GENERATION PAGE AFTER GENERATING SALARY OF THE EMPLOYEE:



A screenshot of the same web application window after the salary has been generated. The "EMP_ID" field now contains the value "100" and the "EMP_NAME" field contains the value "ramesh". The "SALARY" field now contains the value "91750". The "PAY" and "NEXT" buttons are still present at the bottom.

EMPLOYEE PAYROLL	
EMP_ID	<input type="text" value="100"/>
EMP_NAME	<input type="text" value="ramesh"/>
SALARY FOR ONE DAY : 500	
ONE DAY SALARY FOR PART TIME : 250	
SALARY	<input type="text" value="91750"/>
<input type="button" value="PAY"/>	
<input type="button" value="NEXT"/>	

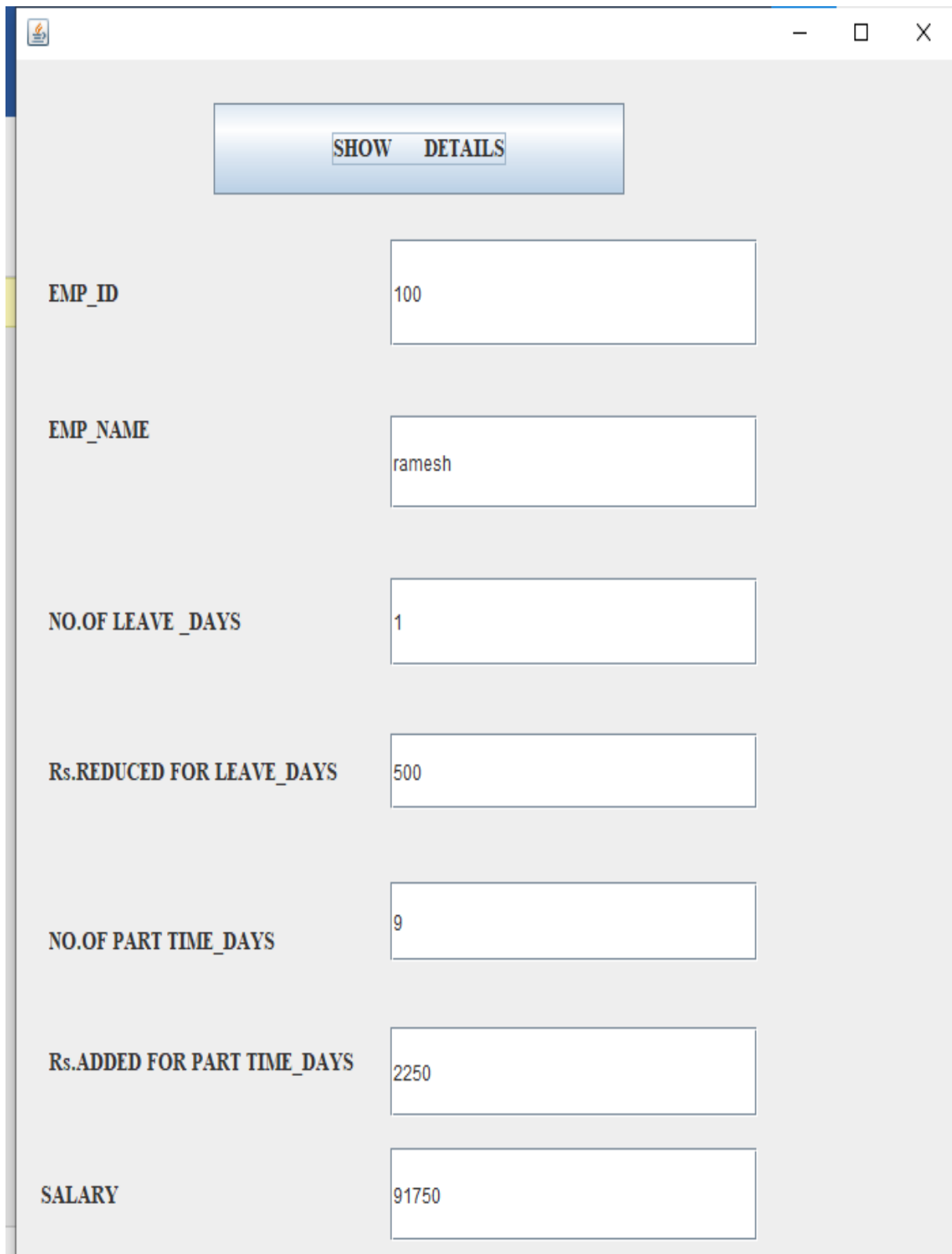
PAYROLL DETAILS:



A screenshot of a web application window titled "PAYROLL DETAILS:". The window has a standard title bar with a minimize button, a maximize button, and a close button. Inside the window, there is a light gray background. At the top center, there is a blue button with the text "SHOW DETAILS". Below this button, there are seven rows of input fields. Each row consists of a label on the left and a text input box on the right. The labels are: "EMP_ID", "EMP_NAME", "NO.OF LEAVE_DAYS", "Rs.REDUCED FOR LEAVE_DAYS", "NO.OF PART TIME_DAYS", "Rs.ADDED FOR PART TIME_DAYS", and "SALARY". The input boxes are empty.

EMP_ID	<input type="text"/>
EMP_NAME	<input type="text"/>
NO.OF LEAVE_DAYS	<input type="text"/>
Rs.REDUCED FOR LEAVE_DAYS	<input type="text"/>
NO.OF PART TIME_DAYS	<input type="text"/>
Rs.ADDED FOR PART TIME_DAYS	<input type="text"/>
SALARY	<input type="text"/>

PAYROLL DETAILS AFTER GENERATING DETAILS OF EMPLOYEE:



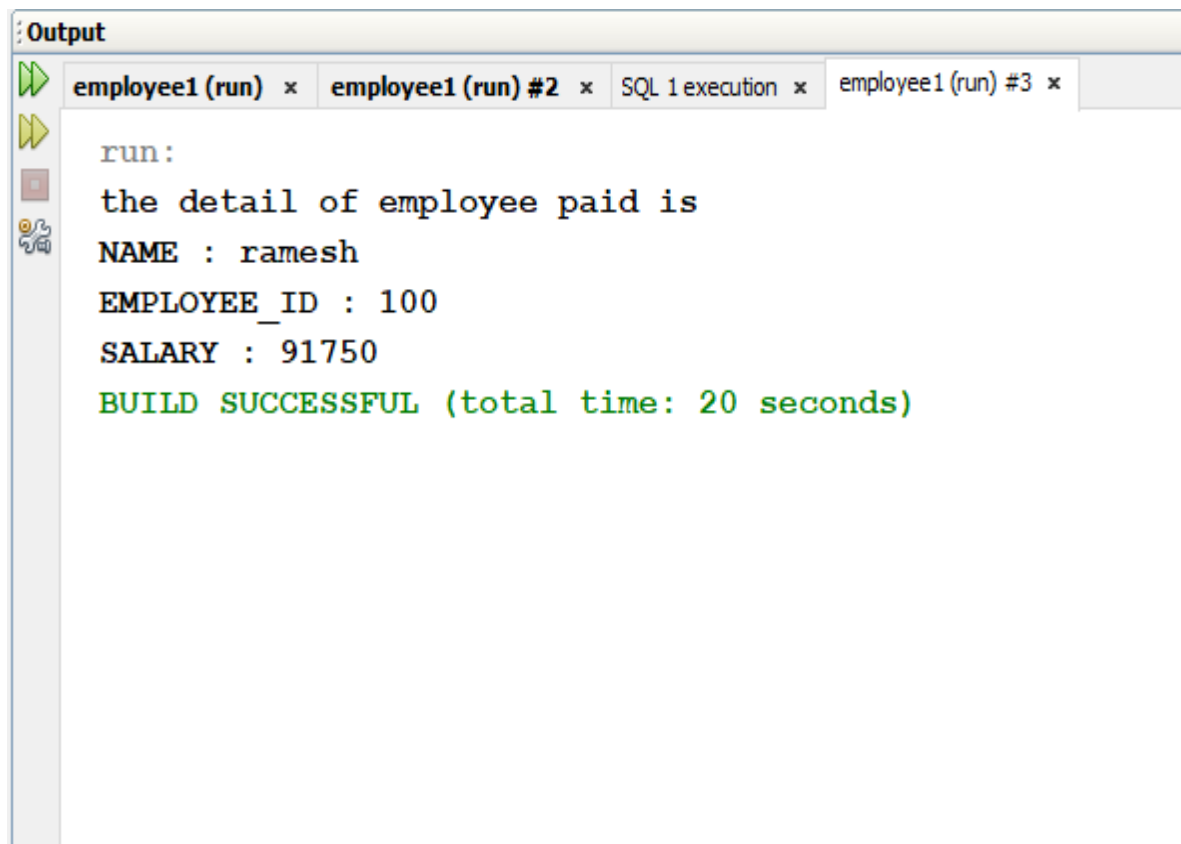
The screenshot shows a web application window with a title bar containing a logo and standard window controls (minimize, maximize, close). The main content area has a light gray background. At the top center is a blue button with the text "SHOW DETAILS". Below this, there are seven rows of labels and input fields. The labels are aligned to the left, and the input fields are to their right. The labels are: "EMP_ID", "EMP_NAME", "NO.OF LEAVE_DAYS", "Rs.REDUCED FOR LEAVE_DAYS", "NO.OF PART TIME_DAYS", "Rs.ADDED FOR PART TIME_DAYS", and "SALARY". The corresponding values entered in the input fields are: "100", "ramesh", "1", "500", "9", "2250", and "91750".

EMP_ID	100
EMP_NAME	ramesh
NO.OF LEAVE_DAYS	1
Rs.REDUCED FOR LEAVE_DAYS	500
NO.OF PART TIME_DAYS	9
Rs.ADDED FOR PART TIME_DAYS	2250
SALARY	91750

FILE LOCATION:

<div> <div> <div>←</div> <div>→</div> <div>⌵</div> <div>⬆</div> </div> <div> <div>📁</div> <div>> This PC > Documents > NetBeansProjects > employee1 > src > employee1</div> </div> </div>				
	Name	Date modified	Type	Size
<div> <div>★ Quick access</div> <div> <div>🖥 Desktop</div> <div>⬇ Downloads</div> <div>📄 Documents</div> <div>🖼 Pictures</div> <div>📁 best</div> <div>📁 IYAPPAN VIJAYAM-I</div> <div>📁 lab-4year</div> <div>📁 NetBeansProjects</div> <div>☁ OneDrive</div> <div>🖥 This PC</div> <div>📦 3D Objects</div> <div>🖥 Desktop</div> <div>📄 Documents</div> <div>⬇ Downloads</div> <div>🎵 Music</div> <div>🖼 Pictures</div> </div> </div>	🖼 .png	12-11-2022 13:35	PNG File	46 KB
	🖼 background1.jpg	12-11-2022 13:53	JPG File	47 KB
	🖼 bef.jpeg	12-11-2022 13:33	JPEG File	10 KB
	🖼 Capture.PNG	12-11-2022 14:00	PNG File	450 KB
	📄 details.java	12-11-2022 11:34	Java Source File	4 KB
	🖱 Docker Desktop	12-11-2022 13:39	Shortcut	3 KB
	📄 Employee1.java	12-11-2022 10:32	Java Source File	5 KB
	📄 EXAMPLE.form	12-11-2022 14:50	FORM File	3 KB
	📄 EXAMPLE.java	12-11-2022 14:50	Java Source File	4 KB
	📄 FINAL.form	26-11-2022 12:51	FORM File	12 KB
	📄 FINAL.java	26-11-2022 12:51	Java Source File	14 KB
	📄 generatingsalary.java	12-11-2022 10:11	Java Source File	2 KB
	🖼 gog.jpeg	12-11-2022 13:33	JPEG File	6 KB
	📄 main.form	25-11-2022 22:11	FORM File	6 KB
	📄 main.java	25-11-2022 22:11	Java Source File	6 KB
	📄 main1.form	26-11-2022 12:51	FORM File	9 KB
	📄 main1.java	26-11-2022 12:51	Java Source File	9 KB
	🖼 Screenshot 2021-04-03 141257.png	12-11-2022 13:27	PNG File	40 KB
	📄 user.form	12-11-2022 15:22	FORM File	9 KB
	📄 user.java	12-11-2022 15:22	Java Source File	9 KB

FINAL OUTPUT:



The screenshot shows an IDE's Output window with a title bar labeled "Output". Below the title bar, there are four tabs: "employee1 (run)", "employee1 (run) #2", "SQL 1 execution", and "employee1 (run) #3". The "employee1 (run)" tab is currently selected. The output text in this tab is as follows:

```
run:
the detail of employee paid is
NAME : ramesh
EMPLOYEE_ID : 100
SALARY : 91750
BUILD SUCCESSFUL (total time: 20 seconds)
```


CONCLUSION:

This project is built keeping in mind that it is to be used by only one user that is the admin. It is built for use in small scale organization where the number of employees is limited.

According to the requested requirement the admin can add, manipulate, update and delete all employee data in his organization. The admin can add new departments and delete them. The Admin can also add predefined pay grades for the employees. The required records can be easily viewed by the admin anytime time he wants in an instant. The payment of the employee is based on monthly basis. Numerous validations implemented would enable the admin to enter accurate data. The main objective of this framework is to save time, make the system cost effective and management records efficiently

REFERENCE:

- <https://www.vogella.com/tutorials/JavaPDF/article.html>
- [https://blog.aspose.com/2020/07/20/create-fill-edit-fillable-pdf-form-](https://blog.aspose.com/2020/07/20/create-fill-edit-fillable-pdf-form-field-java/)

[field-java/](#)

- <https://en.m.wikipedia.org/wiki/IText>
- <https://www.tutorialspoint.com/jdbc/jdbc-db-connections.htm>
- <https://www.guru99.com/java-swing-gui.html>
- <https://netbeans.apache.org/kb/docs/java/gui-functionality.html>

