

**SSN College of Engineering**  
**Department of Computer Science and Engineering**  
**UCS 1617 Mini Project Lab**  
**Model Practicals**

---

**Name:** S. Vishakan  
**Class:** CSE-C  
**Reg. No.:** 18 5001 196  
**Batch:** 6

**Problem Statement:**

To develop a system to automate the payroll process, wherein the payroll administrator can manage his/her employees. Employees details are supposed to be added into the database and persistently stored for future reference. Automation of payroll also consists of the administrator being able to credit the basic salary to a specific employee according to his/her designation, and let the payroll system calculate the various other dependencies like HRA, DA, PF and other deductions necessary, to output the gross pay and a payslip to the specified employee.

The application should enable hassle-free, one-stop shop solutions to all needs that a company might have relating to its employee's financials. The administrator has a privileged, secure login to the payroll system with credentials and each employee is also given his/her account credentials to view their records. The payroll system must also abide and accomplice the legal statutory requirements for allotting salaries to employees, making sure that no staff is underpaid. It should also enable transparency and a reliable means of bookkeeping.

Employees should also be able to effectively file taxes from the same payroll portal to comply with government tax laws. For this purpose, the software must provide support to popular third-party tax vendor systems as well. An employee should be able to view & generate his payslip as and when required.

Administrators should be able to view the details of his/her employees at his/her will. The system should promote ease-of-use and must have a low mean-time-to-repair & a low failure rate to ensure that all the required components are working smoothly, ensuring better productivity for the application's consumers.

The application must also be capable of servicing multiple employee requests at once without overhead or significant lag, and should not be hindered by scale of the company.

## **Actors & Use Cases:**

### **Actors:**

1. Employee
2. Administrator
3. Tax System

### **Use Cases:**

#### **Employee:**

1. Login
2. Change Credentials
3. View Payslip
4. View Profile Details
5. Pay Taxes
6. Raise Issue To Admin
7. Get Tax Slip
8. Logout

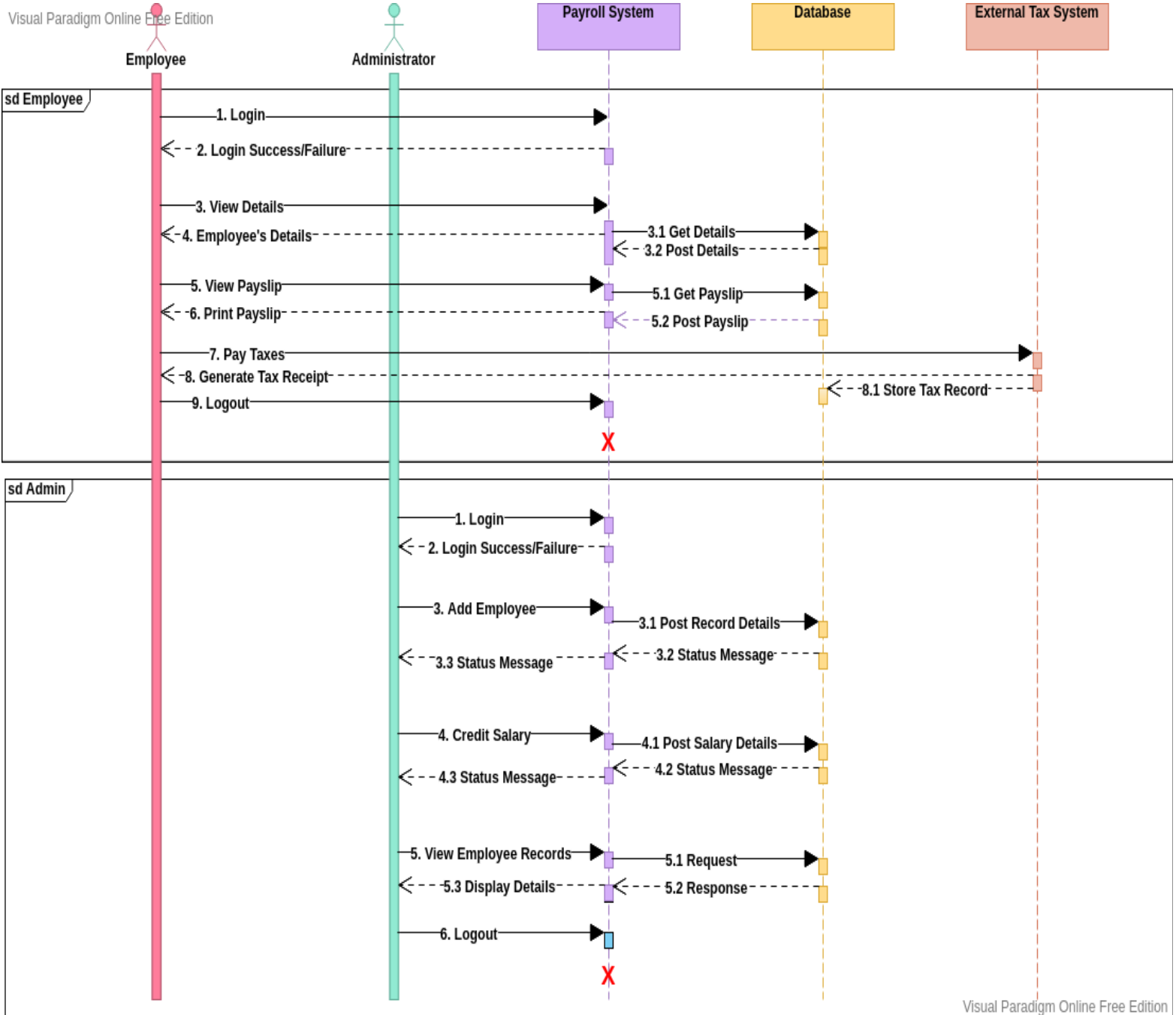
#### **Admin:**

1. Login
2. Change Credentials
3. View Employee Records
4. Add Employee
5. Credit Salary to an Employee
6. Update Employee Details
7. Delete Employee
8. Export Tax Data
9. Logout

#### **Tax System:**

1. Generate Tax Statement
2. Calculate Tax
3. Provide Tax Records

# Interaction Diagram:



## Test Cases:

Test ID	Scenario	Data	Predicted Result	Actual Result	Status
1.	Admin - Login	Credentials: Invalid	Alert Invalid Login	Alert Invalid Login	Pass
2.	Admin - Login	Credentials: Valid	Proceed to Admin Page	Proceed to Admin Page	Pass
3.	Employee - Login	Credentials: Invalid	Alert Invalid Login	Alert Invalid Login	Pass
4.	Employee - Login	Credentials: Valid	Proceed to Employee Page	Proceed to Employee Page	Pass
5.	Admin - Add Record	Details: Insufficient /Invalid	Alert Insufficient/Invalid Details	Alert Insufficient/Invalid Details	Pass
6.	Admin - Add Record	Details: Valid	Add Success	Add Success	Pass
7.	Admin - Credit Salary	Details: Invalid Employee ID	Alert Invalid Details	Alert Invalid Details	Pass
8.	Admin - Credit Salary	Details: Valid	Salary Credited	Salary Credited	Pass
9.	Admin - View Records	Database: Empty	No Records	No Records	Pass
10.	Admin - View Records	Database: Has Records	Display Records	Display Records	Pass
11.	Admin - Delete Records	Details: Valid Employee ID	Delete Success	Delete Success	Pass

12.	Admin - Update Records	Details: Valid Employee ID	Update Success	Update Success	Pass
13.	Employee - View Details	Database: Employee Exists	Show Details	Show Details	Pass
14.	Employee - View Payslip	Database: Employee Exists	Show Payslip	Show Payslip	Pass
15.	Employee - Pay Taxes	Database: Employee Exists	Proceed to Tax System	Proceed to Tax System	Pass
16.	Employee - View Tax Records	Database: Employee Exists	Show Tax Records	Show Tax Records	Pass
17.	Admin/Employee - Change Credentials	Valid Previous Credentials	Change Success	Change Success	Pass
18.	Admin/Employee - Change Credentials	Invalid Previous Credentials	Change Failure	Change Failure	Pass
19.	Admin - Export Tax Data	Valid Credentials	Export Success	Export Success	Pass
20.	Admin/Employee - Logout	nil	Logout Success	Logout Success	Pass

## **Improvisation with Design Patterns:**

Design Patterns help the developer to implement the programming methodology in a tried and tested manner that has achieved high results.

Here, we can make use of the MVC Pattern to separate out the three different entities, thus ensuring low coupling and high cohesion between different interfaces.

The Model is provided by the database of the application, which holds all the records and necessary data.

The View is provided by the User Interface of the application, which separates the concerns of obtaining/modifying the underlying data, and just concerns the presentation and the interaction with the user.

The Controller is the underlying system's backend, which in my implementation is done by the Java Servlets which take care of obtaining the data from the user interface and performing the necessary calculations and manipulations before sending it to the Model layer for persistent storage. This makes programming easier and debugging easier.

Another interesting design pattern is the adapter pattern, which can be put to good use here. Many external tax calculator systems are available on the market, and each has its own set of protocols and dependencies. Connecting a few of these popular systems to our payroll system is made easy with an adapter class with which multiple different tax systems can be accommodated and used with a single interface. The subtleties are abstracted away from the user and he/she needs to only worry about the data.

We can keep the database implementations away from the actual data classes, like the Employee/Admin class. These two classes act as Information Expert classes and should not be concerned with database functionalities. This keeps coupling low. It also enables separation of concerns and benefits the application by providing greater extensibility and modularity.

Test case planning is greatly simplified and high maintainability is also achieved with the use of these design patterns.

## Implementation:

### Login.HTML

```
<!DOCTYPE html>
<html>

<head>
    <title>PMS Portal</title>
    <link rel="stylesheet" href="index.css">
</head>

<body>
    <h1 class="titleText">Welcome to Payroll Mangement System</h1>
    <hr>
    <h2 class="titleText">Login Portal</h2><br><br>
    <div align="center" class="reg-page">
        <label for="id">Username:</label>
        <input type="text" name="uname" id="uname" /><br><br>
        <label for="id">Password:</label>
        <input type="password" name="pwd" id="pwd" /><br><br>
        <input type="submit" value="Login" id="button" class="btn"
onclick="validateUser()" />
    </div>

    <script>
        function validateUser() {
            //To validate the login of the administrator

            var user = document.getElementById("uname").value;
            var pass = document.getElementById("pwd").value;

            if (user === "admin" && pass === "admin") {
                document.getElementById("uname").value = "";
                document.getElementById("pwd").value = "";
                window.open("admin.html", target = "_self");
            }
        }
    </script>
</body>
</html>
```



```
else if (user === "user" && pass === "user") {
    document.getElementById("uname").value = "";
    document.getElementById("pwd").value = "";
    window.open("employee.html", target = "_self");
}
else {
    alert("Invalid credentials. Kindly check and try again.");
    document.getElementById("uname").value = "";
    document.getElementById("pwd").value = "";
    document.getElementById("uname").focus();
}
}
</script>

</body>

</html>
```

# Employee.HTML

```
<!DOCTYPE html>
<html>

<head>
  <title>PMS Portal</title>
  <link rel="stylesheet" href="index.css">
</head>

<body>
  <h1 class="titleText">Payroll Management System</h1>
  <hr>
  <h2 class="titleText">Employee Navigation Menu</h2><br><br>
  <div align="center" class="reg-page">
    <form name="formview" method="GET" action="ViewRecords">
      <label for="idv">Employee ID:</label>
      <input type="text" name="idv" id="idv" required/><br><br>
      <input type="submit" class="btn" value="View Your Profile"
id="button" /><br>
    </form>
  </div>

  <br><br>

  <div align="center" class="reg-page">
    <form name="formsalary" method="GET" action="ViewPayslip">
      <label for="idp">Employee ID:</label>
      <input type="text" name="idp" id="idp" required/><br><br>
      <input type="submit" class="btn" value="View Your Payslip"
id="button" /><br>
    </form>
  </div>

  <br><br>

  <div align="center" class="reg-page">
    <button class="btn" onclick="logoutUser()"><a
class="links">Logout</a></button><br><br>
```

```
</div>
```

```
<script>
```

```
  logoutUser = () => {  
    //redirect to home page  
    location.href = "login.html";  
  }
```

```
</script>
```

```
</body>
```

```
</html>
```

# Admin.HTML

```
<!DOCTYPE html>
<html>

<head>
  <title>PMS Portal</title>
  <link rel="stylesheet" href="index.css">
</head>

<body>
  <h1 class="titleText">Payroll Management System</h1>
  <hr>
  <h2 class="titleText">Admin Navigation Menu</h2><br><br>
  <div align="center" class="reg-page">
    <button class="btn"><a href="add.html" , target="_blank"
class="links">Add An Employee</a></button><br><br>
    <button class="btn"><a href="credit.html" , target="_blank"
class="links">Credit Salary to Employee</a></button><br><br>

    <form name="formsalary" method="GET" action="ViewAll">
      <input type="submit" class="btn" value="View Employees"
id="button" /><br>
    </form>

    <button class="btn" onclick="logoutUser()"><a
class="links">Logout</a></button><br><br>
  </div>

  <script>
    logoutUser = () => {
      //redirect to home page
      location.href = "login.html";
    }
  </script>

</body>

</html>
```

# Add.HTML

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>PMS - Add Record</title>
    <link href="index.css" rel="stylesheet" type="text/css" />
</head>

<body>
    <h1 class="titleText">Payroll Management System</h1>
    <hr>
    <h2 class="titleText">Add A Record</h2>
    <br><br>

    <div class="reg-page">
        <form name="regform" id="regform" action="AddRecord" method="GET">
            <label for="id">Employee ID</label><br>
            <input type="text" id="id" name="id" placeholder="P2008"
size="10" required pattern="^P[0-9]+$" title="Enter in the specified
format">
            <br><br>

            <label for="pname">Name</label><br>
            <input type="text" id="pname" name="pname" placeholder="John
Doe" size="40" required pattern="^[a-zA-Z\s]+$" title="Enter English
alphabets only">
            <br><br>

            <label for="age">Age</label><br>
            <input type="number" id="age" name="age" min="1" max="120"
step="1" placeholder="21" required>
            <br><br>

            <label>Gender</label>
```

```
<br>
<input type="radio" id="male" name="gender" value="Male"
required>
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="Female"
required>
<label for="female">Female</label>
<input type="radio" id="other" name="gender" value="Other"
required>
<label for="other">Other</label>

<br><br>

<label for="job">Designation</label><br>
<select id="job" name="job" required>
    <option>Trainee</option>
    <option>Team Leader</option>
    <option>Manager</option>
    <option>HR</option>
</select>
<br><br>

<label for="addr">Address</label><br>
<textarea cols="30" rows="3" id="addr" name="addr"
required></textarea>
<br><br>

<label for="phone">Phone Number</label><br>
<input type="text" id="phone" name="phone"
placeholder="1234567890" size="20" required>
<br><br>

<label for="doj">Date of Join</label><br>
<input type="date" id="doj" name="doj" required><br><br>

<input type="submit" class="btn" name="Submit" id="Submit"
value="Register"></input><br>
<input type="reset" class="btn" name="Reset" id="Reset"
value="Clear All"></input>
</form>
```

```
</div>
```

```
<script>
```

```
    localStorage.setItem("action", "Record Added Successfully!");
```

```
</script>
```

```
</body>
```

```
</html>
```

# Credit.HTML

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>PMS - Add Record</title>
  <link href="index.css" rel="stylesheet" type="text/css" />
</head>

<body>
  <h1 class="titleText">Payroll Management System</h1>
  <hr>
  <h2 class="titleText">Credit Salary</h2>
  <br><br>

  <div class="reg-page">
    <form name="regform" id="regform" action="CreditSalary"
method="GET">
      <label for="id">Employee ID</label><br>
      <input type="text" id="id" name="id" placeholder="P2008"
size="10" required pattern="^P[0-9]+$" title="Enter in the specified
format">
      <br><br>

      <label for="job">Designation</label><br>
      <select id="job" name="job" required>
        <option>Trainee</option>
        <option>Team Leader</option>
        <option>Manager</option>
        <option>HR</option>
      </select>
      <br><br>

      <label for="salary">Basic Pay</label><br>
      <input type="number" id="salary" name="salary"
placeholder="66000" size="20" required>
```



```
<br><br>
```

```
    <input type="submit" class="btn" name="Submit" id="Submit"
value="Credit Salary"></input><br>
```

```
    <input type="reset" class="btn" name="Reset" id="Reset"
value="Clear All"></input>
```

```
  </form>
```

```
</div>
```

```
<script>
```

```
    localStorage.setItem("action", "Salary Credited Successfully!");
```

```
</script>
```

```
</body>
```

```
</html>
```

# Index.CSS

```
@import
url("https://fonts.googleapis.com/css2?family=Montserrat&display=swap");

body {
    background: rgb(238, 174, 202);
    background: radial-gradient(
        circle,
        rgba(238, 174, 202, 1) 0%,
        rgba(148, 187, 233, 1) 100%
    );
    font-family: "Montserrat", sans-serif;
}

.titleText,
.responseText {
    text-align: center;
}

.reg-page {
    width: 500px;
    position: relative;
    z-index: 1;
    background: #ffffff;
    max-width: 500px;
    margin: 0 auto 100px;
    padding: 45px;
    text-align: left;
    box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0,
0.24);
}

.reg-page input,
.reg-page textarea,
.reg-page select {
    outline: 0;
    background: #f2f2f2;
    width: 100;
```

```

border: 0;
margin: 15px 0px 15px 40px;
padding: 15px;
box-sizing: border-box;
font-size: 16px;
font-family: "Montserrat", sans-serif;
}

.reg-page label {
  font-weight: 650;
}

.reg-page textarea {
  resize: none;
}

.reg-page .btn {
  margin-left: 140px;
  font-family: "Montserrat", sans-serif;
  font-size: 16px;
  font-weight: 300;
  text-transform: uppercase;
  outline: 0;
  background: #4bd6f4;
  width: 200px;
  border: 0;
  padding: 15px;
  color: #ffffff;
  transition: all 0.3 ease;
  cursor: pointer;
  user-select: none;
}

.btn:hover,
.btn:focus,
.btn:active {
  font-weight: bold;
  background: #32afc9;
}

```

```
footer {  
    color: black;  
    font-weight: bold;  
    border-top: 3px solid black;  
    padding: 5px 5px 5px 5px;  
    text-align: center;  
    width: 100%;  
}
```

```
table,  
td,  
th {  
    font-size: 25px;  
    border: 2px solid black;  
    padding: 5px;  
    border-collapse: collapse;  
    text-align: center;  
}
```

```
table {  
    width: 75%;  
}
```

```
tr:hover {  
    background-color: azure;  
}
```

```
.links {  
    text-decoration: none;  
    cursor: pointer;  
    color: white;  
}
```

```
.links:hover {  
    font-weight: bold;  
}
```

# AddRecord.JAVA

```
import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class AddRecord extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException{

        //DB_URL: jdbc:mysql:location:port/DatabaseName?Parameters

        String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
        String DB_URL =
"jdbc:mysql://localhost:3306/mysql?autoReconnect=true&useSSL=false";
        String USER = "admin";
        String PASS = "password";

        PrintWriter out = response.getWriter();

        String eid = request.getParameter("id");
        String name = request.getParameter("pname");
        int age = Integer.parseInt(request.getParameter("age"));
        String gdr = request.getParameter("gender");

        switch(gdr){
            case "Male":
                gdr = "M";
                break;
            case "Female":
                gdr = "F";
                break;
            case "Other":
                gdr = "O";
                break;
            default:
```

```

        gdr = null;
    }

    String job = request.getParameter("job");
    String addr = request.getParameter("addr");
    long phone = Long.parseLong(request.getParameter("phone"));
    String doj = request.getParameter("doj");

    try{
        Class.forName(JDBC_DRIVER);
        Connection conn = DriverManager.getConnection(DB_URL, USER,
PASS);

        String insertCmd = "INSERT INTO Employee VALUES(?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?)";
        PreparedStatement stmt = conn.prepareStatement(insertCmd);
        stmt.setString(1, eid);
        stmt.setString(2, name);
        stmt.setInt(3, age);
        stmt.setString(4, gdr);
        stmt.setString(5, job);
        stmt.setString(6, addr);
        stmt.setLong(7, phone);
        stmt.setString(8, doj);
        stmt.setDouble(9, 0.0);
        stmt.setDouble(10, 0.0);
        stmt.setDouble(11, 0.0);
        stmt.setDouble(12, 0.0);
        stmt.setDouble(13, 0.0);
        stmt.setDouble(14, 0.0);

        stmt.executeUpdate();    //execute the PreparedStatement

        //To open the admin.html page after adding the record
        RequestDispatcher rd =
request.getRequestDispatcher("/admin.html");
        rd.forward(request, response);

        stmt.close();
        conn.close();
    }
}

```

```
}

    catch(SQLException se){
        se.printStackTrace();
        out.println(se);
    }

    catch(Exception e){
        e.printStackTrace();
        out.println(e);
    }
}
}
```

# CreditSalary.JAVA

```
import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class CreditSalary extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException{

        //DB_URL: jdbc:mysql:location:port/DatabaseName?Parameters

        String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
        String DB_URL =
"jdbc:mysql://localhost:3306/mysql?autoReconnect=true&useSSL=false";
        String USER = "admin";
        String PASS = "password";

        PrintWriter out = response.getWriter();

        String eid = request.getParameter("id");
        String job = request.getParameter("job");
        Double bpay = Double.parseDouble(request.getParameter("salary"));
        Double gpay = 0.0, da = 0.0, hra = 0.0, pf = 0.0, ddns = 0.0;

        switch(job){
            case "Trainee":
                da = 0.1 * bpay;
                hra = 0.05 * bpay;
                pf = 0.02 * bpay;
                ddns = 0.01 * bpay;
                break;
            case "Team Leader":
                da = 0.15 * bpay;
                hra = 0.1 * bpay;
                pf = 0.05 * bpay;
```



```

        ddns = 0.03 * bpay;
        break;
    case "Manager":
        da = 0.2 * bpay;
        hra = 0.12 * bpay;
        pf = 0.07 * bpay;
        ddns = 0.03 * bpay;
        break;
    case "HR":
        da = 0.3 * bpay;
        hra = 0.15 * bpay;
        pf = 0.1 * bpay;
        ddns = 0.05 * bpay;
        break;
}

gpay = bpay + da + hra - pf - ddns;

try{
    Class.forName(JDBC_DRIVER);
    Connection conn = DriverManager.getConnection(DB_URL, USER,
PASS);

    String updateCmd = "UPDATE Employee SET bpay = ?, da = ?, hra =
?, pf = ?, ddns = ?, gpay = ? WHERE eid = ?";
    PreparedStatement stmt = conn.prepareStatement(updateCmd);
    stmt.setDouble(1, bpay);
    stmt.setDouble(2, da);
    stmt.setDouble(3, hra);
    stmt.setDouble(4, pf);
    stmt.setDouble(5, ddns);
    stmt.setDouble(6, gpay);
    stmt.setString(7, eid);

    stmt.executeUpdate();    //execute the PreparedStatement

    //To open the admin.html page after adding the record
    RequestDispatcher rd =
request.getRequestDispatcher("/admin.html");
    rd.forward(request, response);

```

```
        stmt.close();  
        conn.close();  
    }  
  
    catch(SQLException se){  
        se.printStackTrace();  
        out.println(se);  
    }  
  
    catch(Exception e){  
        e.printStackTrace();  
        out.println(e);  
    }  
}  
}
```

# ViewRecords.JAVA

```
import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ViewRecords extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException{

        //DB_URL: jdbc:mysql:location:port/DatabaseName?Parameters

        String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
        String DB_URL =
"jdbc:mysql://localhost:3306/mysql?autoReconnect=true&useSSL=false";
        String USER = "admin";
        String PASS = "password";

        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        String eid = request.getParameter("idv");

        try{
            Class.forName(JDBC_DRIVER);
            Connection conn = DriverManager.getConnection(DB_URL, USER,
PASS);

            String viewCmd = "SELECT * FROM Employee where eid = '" + eid +
"'" ;

            //to navigate forwards and backwards, and also to make the
result set read-only
            Statement stmt =
conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_READ_ONLY);
```

```

ResultSet rs = stmt.executeQuery(viewCmd);

if(rs.first()){    //to check if the database has any rows

    rs.beforeFirst();    //to set the cursor back before the
first row

    out.println("<!DOCTYPE HTML>");
    out.println("<html><head><title>Employee
Details</title><link rel='stylesheet' href='index.css'></head>");
    out.println("<body><h1 class='titleText'>Employee
Details</h1><hr><br><table align='center'>");
    out.println("<tr><th>Employee
ID</th><th>Name</th><th>Age</th><th>Gender</th>" +
        "<th>Designation</th><th>Address</th><th>Phone</th><th>Date
of Join</th></tr>");

    while(rs.next()){
        String pid = rs.getString("eid");
        String pname = rs.getString("pname");
        int age = rs.getInt("age");
        String gender = rs.getString("gender");

        switch(gender){
            case "M":
                gender = "Male";
                break;
            case "F":
                gender = "Female";
                break;
            case "O":
                gender = "Other";
                break;
            default:
                gender = "Null";
        }

        String job = rs.getString("job");
        String addr = rs.getString("addr");
        long phone = rs.getLong("phone");
    }
}

```

```

        String dov = rs.getString("doj");

        out.println("<tr><td>" + pid + "</td><td>" + pname +
"</td><td>" + age + "</td><td>" + gender +
        "</td><td>" + job + "</td><td>" + addr + "</td><td>" +
phone + "</td><td>" + dov + "</td></tr>");

    }

    out.println("</table></body></html>");

}

else{
    out.println("The employee does not exist.<br>");
}

stmt.close();
conn.close();
}

catch(SQLException se){
    se.printStackTrace();
    out.println(se);
}

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

```

# ViewPayslip.JAVA

```
import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ViewPayslip extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException{

        //DB_URL: jdbc:mysql:location:port/DatabaseName?Parameters

        String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
        String DB_URL =
"jdbc:mysql://localhost:3306/mysql?autoReconnect=true&useSSL=false";
        String USER = "admin";
        String PASS = "password";

        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        String eid = request.getParameter("idp");

        try{
            Class.forName(JDBC_DRIVER);
            Connection conn = DriverManager.getConnection(DB_URL, USER,
PASS);

            String viewCmd = "SELECT * from Employee where eid = \"" + eid
+ "\"";

            //to navigate forwards and backwards, and also to make the
result set read-only
            Statement stmt =
conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_READ_ONLY);
```

```

ResultSet rs = stmt.executeQuery(viewCmd);

if(rs.first()){          //to check if the database has any rows

    rs.beforeFirst();    //to set the cursor back before the
first row

    out.println("<!DOCTYPE HTML>");
    out.println("<html><head><title>Employee
Details</title><link rel='stylesheet' href='index.css'></head>");
    out.println("<body><h1 class='titleText'>Employee
Details</h1><hr><br><table align='center'>");
    out.println("<tr><th>Employee ID</th><th>Basic
Pay</th><th>HRA</th><th>DA</th>" +
        "<th>Provident Fund</th><th>Deductions</th><th>Gross
Pay</th></tr>");

    while(rs.next()){
        String pid = rs.getString("eid");
        Double bpay = rs.getDouble("bpay");
        Double hra = rs.getDouble("hra");
        Double da = rs.getDouble("da");
        Double pf = rs.getDouble("pf");
        Double ddns = rs.getDouble("ddns");
        Double gpay = rs.getDouble("gpay");

        out.println("<tr><td>" + pid + "</td><td>" + bpay +
"</td><td>" + hra + "</td><td>" + da +
        "</td><td>" + pf + "</td><td>" + ddns + "</td><td>" +
gpay + "</td></tr>");

    }

    out.println("</table></body></html>");

}

else{
    out.println("The employee does not exist.<br>");
}

```

```
        stmt.close();
        conn.close();
    }

    catch(SQLException se) {
        se.printStackTrace();
        out.println(se);
    }

    catch(Exception e) {
        e.printStackTrace();
        out.println(e);
    }
}
}
```



# ViewAll.JAVA

```
import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ViewAll extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException{

        //DB_URL: jdbc:mysql:location:port/DatabaseName?Parameters

        String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
        String DB_URL =
"jdbc:mysql://localhost:3306/mysql?autoReconnect=true&useSSL=false";
        String USER = "admin";
        String PASS = "password";

        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        try{
            Class.forName(JDBC_DRIVER);
            Connection conn = DriverManager.getConnection(DB_URL, USER,
PASS);

            String viewCmd = "SELECT * FROM Employee";
            //to navigate forwards and backwards, and also to make the
result set read-only
            Statement stmt =
conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_READ_ONLY);

            ResultSet rs = stmt.executeQuery(viewCmd);

            if(rs.first()){        //to check if the database has any rows
```

```

        rs.beforeFirst();    //to set the cursor back before the
first row

        out.println("<!DOCTYPE HTML>");
        out.println("<html><head><title>Employee
Details</title><link rel='stylesheet' href='index.css'></head>");
        out.println("<body><h1 class='titleText'>Employee
Details</h1><hr><br><table align='center'>");
        out.println("<tr><th>Employee
ID</th><th>Name</th><th>Age</th><th>Gender</th>" +
        "<th>Designation</th><th>Address</th><th>Phone</th><th>Date
of Join</th></tr>");

        while(rs.next()){
            String pid = rs.getString("eid");
            String pname = rs.getString("pname");
            int age = rs.getInt("age");
            String gender = rs.getString("gender");

            switch(gender){
                case "M":
                    gender = "Male";
                    break;
                case "F":
                    gender = "Female";
                    break;
                case "O":
                    gender = "Other";
                    break;
                default:
                    gender = "Null";
            }

            String job = rs.getString("job");
            String addr = rs.getString("addr");
            long phone = rs.getLong("phone");
            String dov = rs.getString("doj");

```

```

        out.println("<tr><td>" + pid + "</td><td>" + pname +
"</td><td>" + age + "</td><td>" + gender +
        "</td><td>" + job + "</td><td>" + addr + "</td><td>" +
phone + "</td><td>" + dov + "</td></tr>");

    }

    out.println("</table></body></html>");

}

else{
    out.println("Database is empty.<br>");
}

stmt.close();
conn.close();
}

catch(SQLException se){
    se.printStackTrace();
    out.println(se);
}

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

```

# Script.SQL

```
USE mysql;
```

```
/* Records Table Creation */
```

```
DROP TABLE Employee;
```

```
CREATE TABLE Employee(  
    eid      CHAR(5) PRIMARY KEY,  
    pname    VARCHAR(40),  
    age      NUMERIC(3),  
    gender   CHAR(1),  
    job      VARCHAR(25),  
    addr     VARCHAR(100),  
    phone    NUMERIC(12),  
    doj      DATE,  
    bpay     NUMERIC(12, 2),  
    da       NUMERIC(10, 2),  
    hra      NUMERIC(10, 2),  
    pf       NUMERIC(10, 2),  
    ddns     NUMERIC(10, 2),  
    gpay     NUMERIC(12, 2)  
);
```

# Web.XML

```
<web-app>
  <servlet>
    <servlet-name>AddRecord</servlet-name>
    <servlet-class>AddRecord</servlet-class>
  </servlet>

  <servlet>
    <servlet-name>CreditSalary</servlet-name>
    <servlet-class>CreditSalary</servlet-class>
  </servlet>

  <servlet>
    <servlet-name>ViewRecords</servlet-name>
    <servlet-class>ViewRecords</servlet-class>
  </servlet>

  <servlet>
    <servlet-name>ViewAll</servlet-name>
    <servlet-class>ViewAll</servlet-class>
  </servlet>

  <servlet>
    <servlet-name>ViewPayslip</servlet-name>
    <servlet-class>ViewPayslip</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>AddRecord</servlet-name>
    <url-pattern>/AddRecord</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>CreditSalary</servlet-name>
    <url-pattern>/CreditSalary</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
```

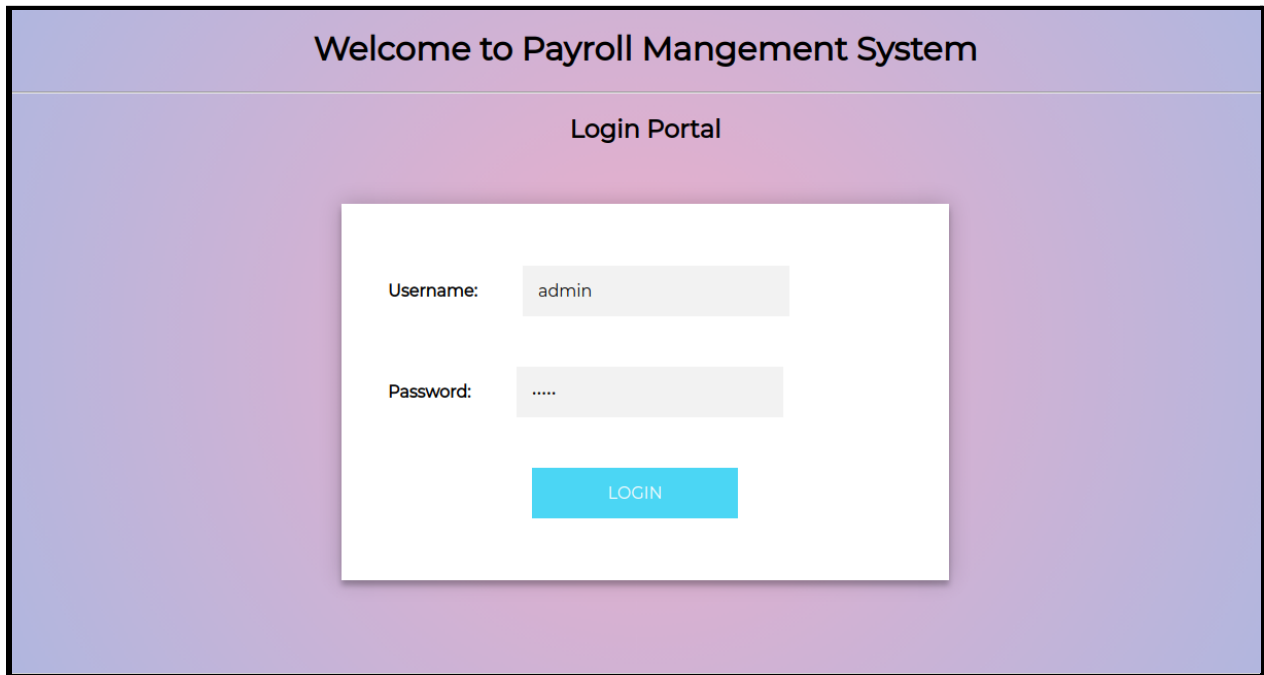
```
        <servlet-name>ViewRecords</servlet-name>
        <url-pattern>/ViewRecords</url-pattern>
    </servlet-mapping>

    <servlet-mapping>
        <servlet-name>ViewAll</servlet-name>
        <url-pattern>/ViewAll</url-pattern>
    </servlet-mapping>

    <servlet-mapping>
        <servlet-name>ViewPayslip</servlet-name>
        <url-pattern>/ViewPayslip</url-pattern>
    </servlet-mapping>

    <welcome-file-list>
        <welcome-file>login.html</welcome-file>
    </welcome-file-list>
</web-app>
```

## Results:



Welcome to Payroll Mangement System

Login Portal

Username: admin

Password: .....

LOGIN

This screenshot shows the login interface of the Payroll Management System. It features a light purple header with the text 'Welcome to Payroll Mangement System'. Below this is a 'Login Portal' section with a white background. The login form includes a 'Username' field with the text 'admin' and a 'Password' field with masked characters '.....'. A blue 'LOGIN' button is positioned below the password field.

## Login & Employee Portal



Payroll Management System

Admin Navigation Menu

ADD AN EMPLOYEE

CREDIT SALARY TO EMPLOYEE

VIEW EMPLOYEES

LOGOUT

This screenshot displays the admin navigation menu of the Payroll Management System. It has a light purple header with the text 'Payroll Management System'. Below the header is an 'Admin Navigation Menu' section with a white background. The menu contains four blue buttons: 'ADD AN EMPLOYEE', 'CREDIT SALARY TO EMPLOYEE', 'VIEW EMPLOYEES', and 'LOGOUT'.

## Payroll Management System

### Add A Record

Employee ID

P2004

Name

Jake Peralta

Age

39

Gender

☒ Male ☐ Female ☐ Other

Designation

Team Leader ▾

Address

99, Holt Avenue, Brooklyn, NY

Phone Number

9887395321

Date of Join

06/05/2021



REGISTER

CLEAR ALL

Add Employee Portal



# Payroll Management System

## Credit Salary

Employee ID

P2003

Designation

Trainee

Basic Pay

23000

CREDIT SALARY

CLEAR ALL

## Credit Salary Portal & View Employees Page

### Employee Details

Employee ID	Name	Age	Gender	Designation	Address	Phone	Date of Join
P2001	John Doe	25	Male	Trainee	25, Sunset Blvd, NY	8887654321	2021-05-05
P2002	Dwight Schrute	35	Male	Manager	27, Office Avenue, Scranton, PA	9887543211	2021-02-14
P2003	Kim Wexler	45	Female	HR	44, Fring Street, Albuquerque, NM	87692294783	2020-08-07
P2004	Jake Peralta	39	Male	Team Leader	99, Holt Avenue, Brooklyn, NY	9887395321	2021-05-06

## Payroll Management System

### Employee Navigation Menu

Employee ID:

VIEW YOUR PROFILE

Employee ID:

VIEW YOUR PAYSリップ

LOGOUT

### Employee Details

Employee ID	Name	Age	Gender	Designation	Address	Phone	Date of Join
P2004	Jake Peralta	39	Male	Team Leader	99, Holt Avenue, Brooklyn, NY	9887395321	2021-05-06

### Employee Details

Employee ID	Basic Pay	HRA	DA	Provident Fund	Deductions	Gross Pay
P2004	23000.0	1150.0	2300.0	460.0	230.0	25760.0

**Employee Portal, Employee Details Page & Employee Payslip Page**