

Payroll Management System

1. Introduction

The Payroll Management System is a web application designed to automate employee payroll, leave management, and reporting tasks. The system offers role-based access control with two main user roles: Admin and Employee. It ensures secure operations using JWT (JSON Web Tokens) for authentication.

Technology Stack:

- **Backend:** Java with Spring Boot, Spring Security (JWT), MySQL, JPA/Hibernate
- **Frontend:** React, React Router, Axios, React Toastify, Bootstrap, JWT Authentication

2. Overview

This system provides a secure, efficient platform for managing employee payrolls, leave requests, departments, job roles, and reports.

2.1 Authentication

- **Login:** Users log in with a username and password, receiving a JWT token for secure access.
- **Roles:** There are two roles, **Admin** and **Employee**.
- **Access Control:** Endpoints are protected based on the user's role (Admin or Employee).

2.2 Employee Management

- **Admin Functionality:** The admin can create, view, and update employee profiles. Each employee is linked to a department and job role.
- **Employee Functionality:** Employees can only view their personal profile.

2.3 Department & Job Role Management

- **Admin:** Admins can manage departments and job roles by performing CRUD (Create, Read, Update, Delete) operations.

- **Validation:** Admin cannot delete a department or job role if it is linked to an employee.

2.4 Salary & Payroll Management

- **Salary Structure:** Admin defines each employee's salary structure like basic salary, tax, bonus, and deductions.
- **Payroll Run:** Admin can generate payroll for a specific month, which includes salary calculation, bonus, tax, and deductions.
- **Employee View:** Employees can view their payroll for any specific month/year.

2.5 Leave Management

- **Leave Request:** Employees can request leave, and Admin can approve or reject them.

2.6 Payroll Reports

- **Payroll Summary:** Admin can generate reports showing total payroll for all employees over a specific period.
- **Department Report:** Admin can generate reports showing payroll costs per department.

3. Security Measures

- **JWT Token:** Protects API endpoints, ensuring only authorized users can access specific resources.
- **Password Encryption:** User passwords are securely stored using BCrypt.
- **Role-Based Access Control:** Different access permissions are provided based on user roles (Admin vs Employee).

4. API Endpoints

Overview of the API endpoints available for both **Admin** and **Employee** roles:

Authentication

- **POST /auth/login:** Logs in with username/password and returns a JWT token.

User & Common

- **GET** /users/me: Fetch the current logged-in user profile (accessible by both Admin and Employee).
- **GET** /employees/{id}: Fetch employee details by ID (accessible by Admin and Employee).

User Management

- **POST** /users: Create a new user (Admin).
- **PATCH** /users/{id}/status: Activate or deactivate a user (Admin).

Employee Management

- **POST** /employees: Add a new employee (Admin).
- **GET** /employee: Gets list of employees (Admin).
- **PUT** /employees/{id}: Update employee details (Admin).
- **POST** /employees/{id}/salary-structure: Assigns a new salary structure to an employee. (Admin).
- **GET** /employees/{id}/salary-structure: View employee salary structure (Admin).

Department Management

- **GET** /departments: List all departments (Admin).
- **POST** /departments: Create a department (Admin).
- **PUT** /departments: Updates a department's details (Admin).
- **DELETE** /departments/{id}: Delete department (Admin).

Job Role Management

- **GET** /jobs: List all job roles (Admin).

- **POST** /jobs: Create a job role (Admin).
- **PUT** /jobs: Updates a job role details (Admin).
- **DELETE** /jobs/{id}: Delete a job role (Admin).

Leave Management

- **GET** /leaves/pending: View pending leave requests (Admin).
- **PUT** /leaves/{leaveId}: Approve or reject leave requests (Admin).
- **POST** /leaves: Request leave (Employee).
- **GET** /leaves: View leave requests for the logged-in employee (Employee).

Payroll Management

- **GET** /payroll/my/{year}/{month}: View payroll for the employee (Employee).
- **POST** /payroll/runs: Create a payroll run (Admin).
- **POST** /payroll/runs/{id}/process: Process payroll (Admin).
- **GET** /payroll/runs/{id}/items: View payroll data in a specific run (Admin).
- **POST** /payroll/runs/:id/ lock: Locks a payroll run to prevent further changes. (Admin).

Reports

- **GET** /reports/payroll-summary: Generate payroll summary report (Admin).
- **GET** /reports/department-cost: Generate department cost report (Admin).

5. Frontend Dashboard Features

The system features two main dashboards:

Employee Dashboard:

- **Dashboard:** Displays personal details, account info, and leave statistics.
- **Leave Management:** Allows employees to view and request leaves.
- **Payroll:** Displays current and past payroll data, including salary, bonuses, deductions, and net pay.

Admin Dashboard:

- **Dashboard:** Display the total number of employees, departments, job roles and active user status.
- **Employee Management:** View, create, and update employee records.
- **Department Management:** Manage departments by adding, editing, and deleting them.
- **Job Management:** Create, update, and delete job roles.
- **Leave Management:** Approve or reject leave requests.
- **Payroll Management:** Admin can run payroll, view payroll data, and generate reports.

6. Error Handling

Error messages are handled through **React Toastify** and displayed as toast notifications for:

- Unauthorized access, Failed API calls and Validation errors.

7. Conclusion

The Payroll Management System enables seamless management of employee data, payroll, leave requests, and departmental reports. The system is highly secure, with role-based access, encrypted passwords, and JWT authentication. Admins have full control over employee data, payroll processing, and report generation, while employees can manage their profiles, leave requests, and view payroll information.

Login Access based on ROLE:

ADMIN -

username - admin1

password - password

EMPLOYEE -

username - employee1

password - password