

# Payroll Management System

---

## 1. Project Overview

The Payroll Management System is a **web-based application** designed to automate and streamline employee management, payroll processing, and leave management for an organization.

It provides functionalities based on user roles: **Admin** and **Employee**.

- **Backend:** Spring Boot, Spring Security, MySQL
  - **Frontend:** React with Bootstrap
  - **Testing Tools:** Postman, Swagger
- 

## 2. Features

### 2.1 Admin Role

- **Employee Management:** Add, update, delete, and view employee details.
- **Payroll Processing:** Generate monthly salary based on employee and job details.
- **Leave Management:** Approve or reject employee leave requests.
- **Reports:** Generate month-wise and department-wise salary reports.
- **Departments & Jobs:** Define departments and job roles with base salaries.

### 2.2 Employee Role

- **Profile Management:** View personal details.
  - **Leave Requests:** Apply for leave and track status.
  - **Salary Slip:** View monthly salary slips.
- 

## 3. Technology Stack

### 3.1 Backend

- **Framework:** Spring Boot 3.5.5
- **Security:** Spring Security with JWT authentication
- **Database:** MySQL
- **Build Tool:** Maven
- **Testing:** Postman, Swagger UI
- **Dependencies:**
  - Spring Boot Starter Web

- Spring Boot Starter Data JPA
- Spring Boot Starter Security
- Springdoc OpenAPI
- JWT (JWT implementation)
- Jakarta Validation API

### 3.2 Frontend

- **Framework:** React 19.1.1
  - **UI Library:** Bootstrap 5.3.8, React Bootstrap 2.10.10
  - **Form Handling & Validation:** Formik & Yup
  - **Charts & Reports:** Recharts
  - **PDF Generation:** jsPDF
  - **HTTP Requests:** Axios
  - **JWT Handling:** jwt-decode
  - **Routing:** react-router-dom 7.8.2
  - **Icons:** react-icons 5.5.0
- 

## 4. System Architecture

- **Authentication:** JWT-based token authentication
  - **Role-Based Access Control:** Separate routes and features for Admin and Employee
  - **Data Persistence:** Hibernate JPA manages ORM between backend entities and MySQL
  - **Reporting & Analytics:** Backend generates payroll and department cost reports, visualized in frontend using Recharts
- 

## 5. Database Design

**Database Name:** payroll\_db

### 5.1 Tables

- **users** – Stores login credentials and roles
- **employees** – Stores employee details, job, and department info
- **departments** – Stores department names and descriptions
- **jobs** – Stores job roles and base salaries
- **leaves** – Stores leave requests and status

- **payroll\_runs** – Stores monthly payroll run details
- **payroll\_items** – Stores payroll details for individuals and departments

## 5.2 Relationships

- One-to-one: user ↔ employee
  - One-to-many: department ↔ employees
  - One-to-many: job ↔ employees
  - One-to-many: employee ↔ leaves
  - One-to-many: employee ↔ payroll\_runs and payroll\_items
- 

## 6. API Documentation

The backend exposes **RESTful APIs**, secured with **JWT authentication**, with access controlled by role.

### 6.1 Leave Controller

- PUT /api/v1/leaves/{leaveld}/status – Update leave status
- GET /api/v1/leaves – Retrieve all leave requests
- POST /api/v1/leaves – Submit a new leave request
- GET /api/v1/leaves/employee/{employeeid} – Get leave requests for a specific employee

### 6.2 Job Controller

- PUT /api/v1/jobs/{id} – Update a job role
- DELETE /api/v1/jobs/{id} – Delete a job role
- GET /api/v1/jobs – List all job roles
- POST /api/v1/jobs – Create a new job role

### 6.3 Employee Controller

- GET /api/v1/employees/{id} – Get details of a specific employee
- PUT /api/v1/employees/{id} – Update employee info
- DELETE /api/v1/employees/{id} – Remove employee record
- GET /api/v1/employees – List all employees
- POST /api/v1/employees – Add a new employee

### 6.4 Department Controller

- PUT /api/v1/departments/{id} – Update department
- DELETE /api/v1/departments/{id} – Delete department

- GET /api/v1/departments – Retrieve all departments
- POST /api/v1/departments – Create a new department

### 6.5 User Controller

- POST /api/v1/users – Register a new user
- GET /api/v1/users/me – Retrieve logged-in user details

### 6.6 Payroll Controller

- GET /api/v1/payroll/runs – Retrieve payroll runs
- POST /api/v1/payroll/runs – Create a new payroll run
- POST /api/v1/payroll/runs/{id}/process – Process payroll run
- POST /api/v1/payroll/runs/{id}/lock – Lock payroll run
- GET /api/v1/payroll/runs/{id}/items – Get payroll items of a run
- GET /api/v1/payroll/runs/my/{year}/{month} – Get payroll for employee

### 6.7 Auth Controller

- POST /api/v1/auth/login – Authenticate and obtain JWT

### 6.8 Reports Controller

- GET /api/v1/reports/payroll/summary – Payroll summary reports
- GET /api/v1/reports/department-cost – Department cost reports

---

## 7. Application Configuration

The **application.properties** file was used to configure:

- **Database connection** (URL, username, password, dialect)
- **JPA behavior** (schema update, SQL logging)
- **Swagger/OpenAPI settings** (API docs enabled, Swagger UI enabled)
- **JWT settings** (secret key and token expiration time)

---

## 8. Project Setup

### 8.1 Backend

1. Clone repository
2. Open in Eclipse (or any Spring Boot IDE)
3. Configure MySQL credentials in application.properties
4. Run mvn clean install to build

5. Start the Spring Boot application

## 8.2 Frontend

1. Navigate to frontend directory
  2. Run npm install to install dependencies
  3. Run npm start to launch React development server
- 

## 9. Day-wise Progress (with Problems & Solutions)

### Day 1 – Friday, 29th August 2025

#### Work Done:

- Understood problem statement & listed features
- Planned modules for Admin & Employee
- Set up backend project with MySQL + Spring Boot
- Created **UserController** & **EmployeeController**
- Designed entities (Jobs, Departments, Leaves, PayrollRuns, PayrollItems)

#### Problems & Solutions:

- **Entity Mapping Issues** → Infinite recursion during JSON serialization  
Fixed using @JsonBackReference, @JsonManagedReference, @JsonIgnore.
  - **Employee Basic Pay not updating** → Fixed in EmployeeService.java by setting job basic pay explicitly.
  - **403 Errors in Swagger** → Solved by adding JWT Bearer token support in Swagger config.
  - **Swagger UI dependency issues** → Fixed Maven dependency with springdoc-openapi.
- 

### Day 2 – Saturday, 30th August 2025

#### Work Done:

- Defined entity relationships (One-to-One, One-to-Many, Many-to-One)
- Built repositories, services, and controllers for **Jobs, Departments, Leaves**
- Used **DTOs** for clean data transfer
- Implemented **JWT authentication** with role-based access
- Made app stateless by disabling default sessions

### Problems & Solutions:

- **Confusion between @RequestParam & @RequestBody** → Learned correct usage, fixed Leave module API & frontend calls.
  - **JWT not validating** → Fixed filter chain execution order.
  - **Mismatch between backend & frontend API calls** → Adjusted controller signatures and Axios requests.
  - **Role access issues** → Restricted employee from accessing admin APIs with proper security config.
- 

### Day 3 – Sunday, 31st August 2025

#### Work Done:

- Developed **Payroll Module** (Payroll Run + Payroll Items)
- Created payroll processing & salary slip APIs
- Populated database with admin/employee users
- Set up frontend with **React + Axios + JWT**
- Built pages for Admin & Employee
- Integrated API calls, aligned frontend-backend communication

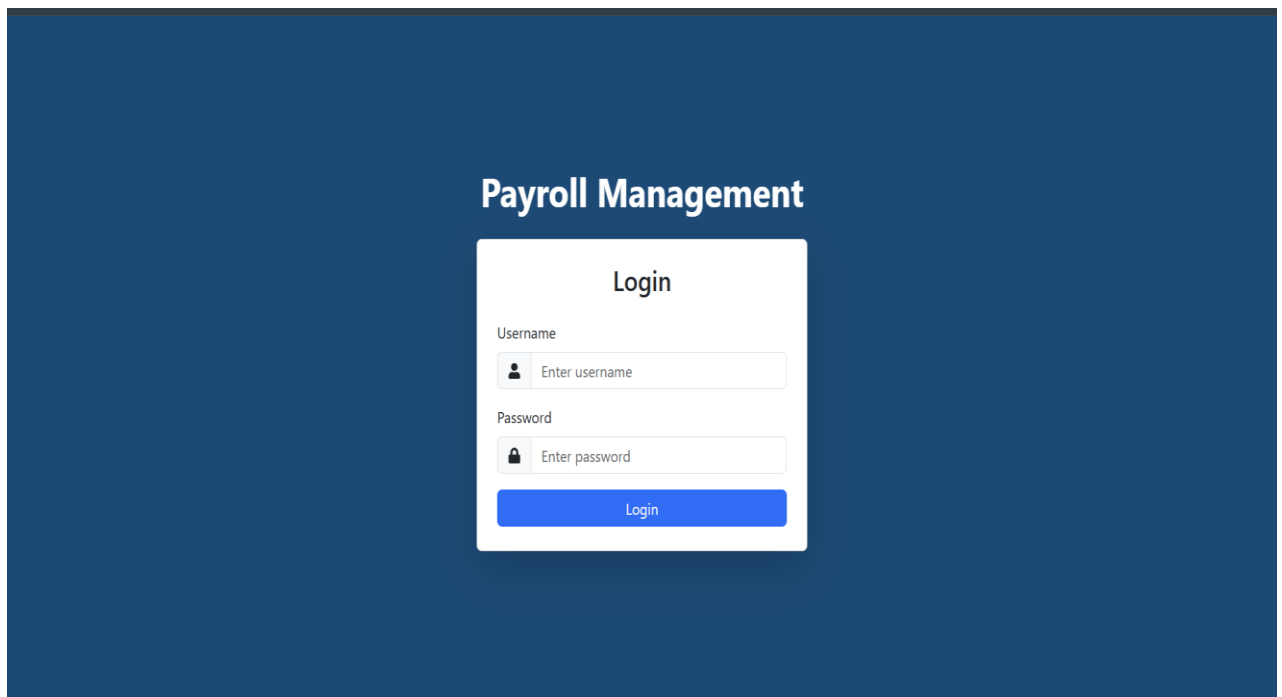
#### Problems & Solutions:

- **Infinite recursion in Payroll entities** → Solved with JSON annotations.
- **Cascade delete not working** → Used cascade + orphan removal + FK constraints.
- **CORS errors** → Enabled allowed origins in Spring config.
- **Axios not sending token** → Implemented Axios interceptors with localStorage.
- **Token expiration handling** → Added auto-logout & redirect to login.

# User Interface

## 1. Login Page

This is the initial login interface where both Admin and Employee users enter their credentials. Based on the role, the system redirects them to their respective dashboards after successful authentication.

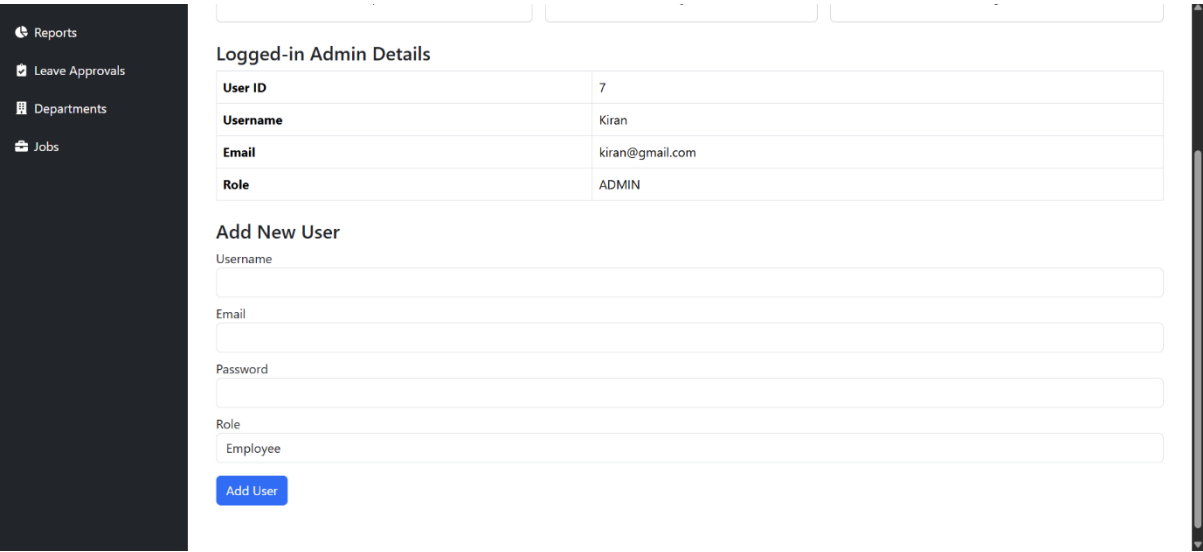
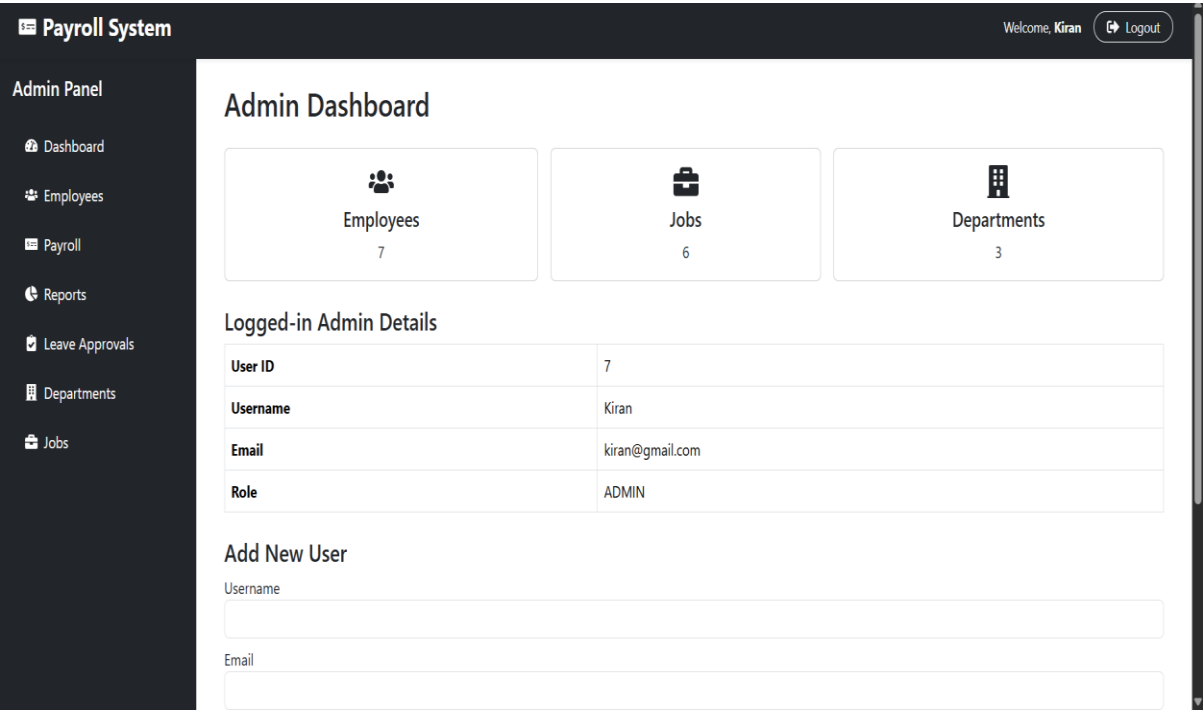


The image shows a login page for a system titled "Payroll Management". The page has a dark blue background. In the center, there is a white login form. The form has a title "Login" at the top. Below the title, there are two input fields: "Username" and "Password". The "Username" field has a placeholder text "Enter username" and a small user icon. The "Password" field has a placeholder text "Enter password" and a small lock icon. Below the input fields, there is a blue button labeled "Login".

## 2. Admin Module

- **Admin Dashboard**

The central hub for the Admin, showing quick access to employee management, payroll processing, leave requests, reports, departments, and jobs.





## Employee Creation

Admin can add a new employee by filling in personal details, job, and department information.

Payroll System

Welcome, **Kiran** [Logout](#)

Admin Panel

[Dashboard](#)[Employees](#)[Payroll](#)[Reports](#)[Leave Approvals](#)[Departments](#)[Jobs](#)

Employees

Add Employee

Add Employee

User ID

First Name

Email

Designation

Job

Date of Birth

Last Name

Phone

Department

Address

Select Job

dd-mm-yyyy

Select Department

Add

Cancel

## Employee Edit

Admin can update existing employee information such as department, job role, or personal details.

Payroll System

Welcome, **Kiran** [Logout](#)

Admin Panel

[Dashboard](#)[Employees](#)[Payroll](#)[Reports](#)[Leave Approvals](#)[Departments](#)[Jobs](#)

Employees

Add Employee

Edit Employee

2

John

john.doe@example.com

Software Engineer

Job

Date of Birth

Doe

9876543210

Department

Address

Software Engineer

IT

dd-mm-yyyy

Update

Cancel

# Employee Deletion

Admin can remove an employee record from the system.

Payroll System

Welcome, KiranLogout

Admin Panel

- Dashboard
- Employees
- Payroll
- Reports
- Leave Approvals
- Departments
- Jobs

Employees

Add Employee

ID	Name	Email	Phone	Job	Department	Actions
1	John Doe	johndoe@example.com	9876543210	Software Engineer	IT	<div>EditDelete</div>
2	raman raj	ramanraj@gmail.com	9837139101	Benefits Analyst	Human Resources	<div>EditDelete</div>
3	rohan jha	rohan@gmail.com	8135138981	Benefits Analyst	Human Resources	<div>EditDelete</div>
4	pavan kumar	pavan@gmail.com	9397853106	Software Engineer	IT	<div>EditDelete</div>
5	Madhav Sharma	maddy@gmail.com	9324724761	Help desk analyst	customer care	<div>EditDelete</div>
6	Shyam Kumar	shyam@gmail.com	9011054353	Recruiter	Human Resources	<div>EditDelete</div>
8	User1 tydg	User@gmail.com	9441121231	Devops engineer	IT	<div>EditDelete</div>

# Leave Approvals

Admin reviews pending leave requests and can approve or reject them directly.

Payroll System

Welcome, KiranLogout

Admin Panel

- Dashboard
- Employees
- Payroll
- Reports
- Leave Approvals
- Departments
- Jobs

Leave Approvals

Leave ID	Employee Name	Start Date	End Date	Leave Type	Status	Actions
1	pavan kumar	2025-08-30	2025-09-10	sick	REJECTED	<div>Change Status</div>
2	rohan jha	2025-09-09	2025-09-12	casual	REJECTED	<div>Change Status</div>
3	Shyam Kumar	2025-09-04	2025-09-06	sick	APPROVED	<div>Change Status</div>
4	pavan kumar	2025-09-25	2025-09-26	earned	APPROVED	<div>Change Status</div>
5	pavan kumar	2025-09-18	2025-09-20	casual	REJECTED	<div>Change Status</div>

PENDING

APPROVED

REJECTED

## Creating Payroll

The Admin can initiate a new payroll run for a selected month and year. Once created, the payroll processing generates salary details for all employees based on their job roles, department, and leave records. The Admin can then review, lock, or download the payroll run for reporting purposes.

Admin Panel

Dashboard

Employees

Payroll

Reports

Leave Approvals

Departments

Jobs

Payroll Summary

2025

9

Create Payroll Run

ID	Year	Month	Status	Actions
1	2025	8	LOCKED	<div>View Items</div>
4	2025	9	LOCKED	<div>View Items</div>
6	2025	7	LOCKED	<div>View Items</div>
7	2025	10	LOCKED	<div>View Items</div>
8	2025	11	LOCKED	<div>View Items</div>

Payroll Items for Run #8

Employee ID	Name	Basic Pay	Allowances	Deductions	Net Pay
1	John Doe	80000	8000	16000	72000
2	raman raj	45000	2250	4500	42750
3	rohan jha	45000	2250	4500	42750
4	pavan kumar	80000	8000	16000	72000

## Department Management (Add, Update, Delete)

Admin can define and manage departments within the organization. This includes creating new departments, updating existing ones, or deleting unused departments.

Payroll System

Welcome, Kiran

Logout

Admin Panel

Dashboard

Employees

Payroll

Reports

Leave Approvals

Departments

Jobs

Departments

Add Department

Add Department

Department Name

Description

Add

Cancel

ID	Name	Description	Actions
1	IT	Information Technology	<div>EditDelete</div>
2	Human Resources	Manages employee recruitment, onboarding and training.	<div>EditDelete</div>
3	customer care	communicating with customer	<div>EditDelete</div>

## Job Management (Add, Update, Delete)

Admin can create and manage job roles along with their base salaries. Updates or deletions of existing job roles can also be done here.

Payroll System

Welcome, Kiran Logout

Admin Panel

- Dashboard
- Employees
- Payroll
- Reports
- Leave Approvals
- Departments
- Jobs

Jobs

Add Job

Add Job

Job Title

Description

Basic Pay

Add

Cancel

ID	Title	Description	Basic Pay	Actions
1	Software Engineer	Develops and maintains code	80000	<div>EditDelete</div>
3	Benefits Analyst	Designs salary structures and benefits packages	45000	<div>EditDelete</div>
4	Help desk analyst	Answers incoming inquiries by phone or email	30000	<div>EditDelete</div>

## Payroll Reports

The **Payroll Management System** provides comprehensive payroll reports to help the organization analyze salary expenditures and maintain transparency. These reports are auto-generated based on payroll runs and displayed in both tabular and graphical formats.

Visual payroll reports transform raw payroll data into charts and graphs, making it easier for management and employees to understand salary distributions and organizational costs.

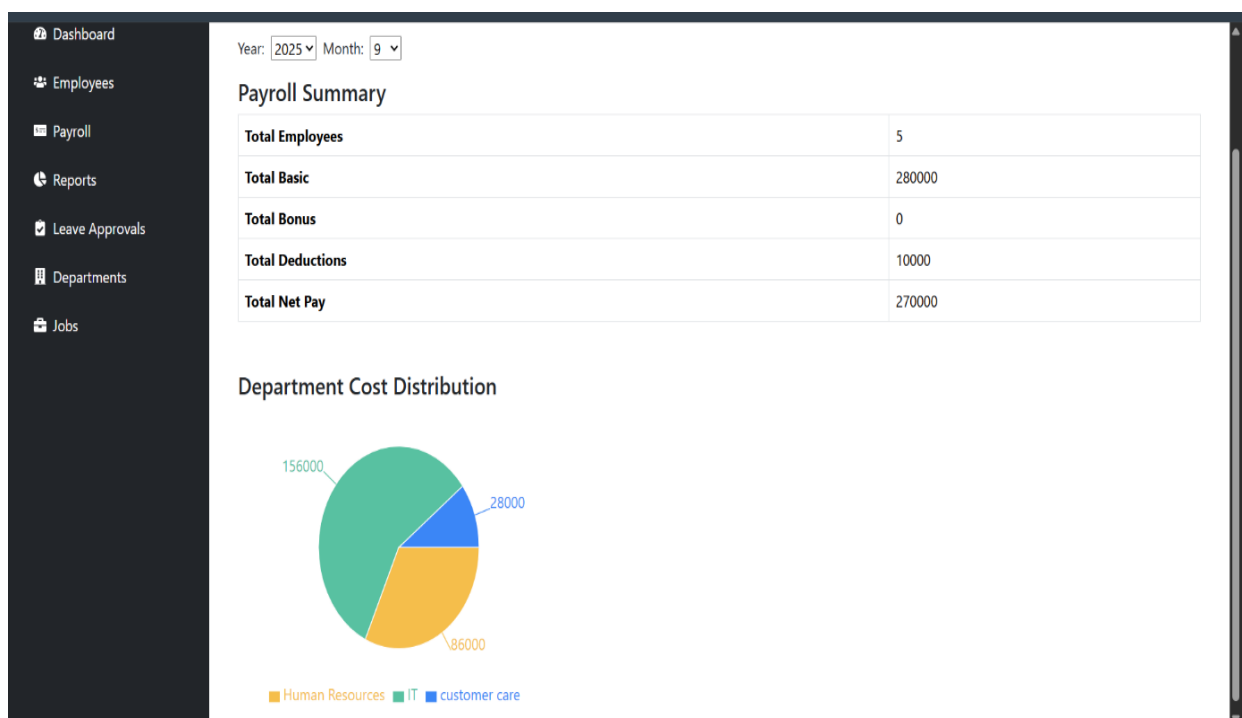
The system calculates salaries for all employees based on job role, department, attendance, and approved leaves.

## Monthly Payroll Reports

The Admin can generate month-wise payroll reports that show employee salaries for a particular payroll run. These reports provide a detailed breakdown of gross pay, deductions, and net salary for each employee in that month.

## Department-Wise Reports

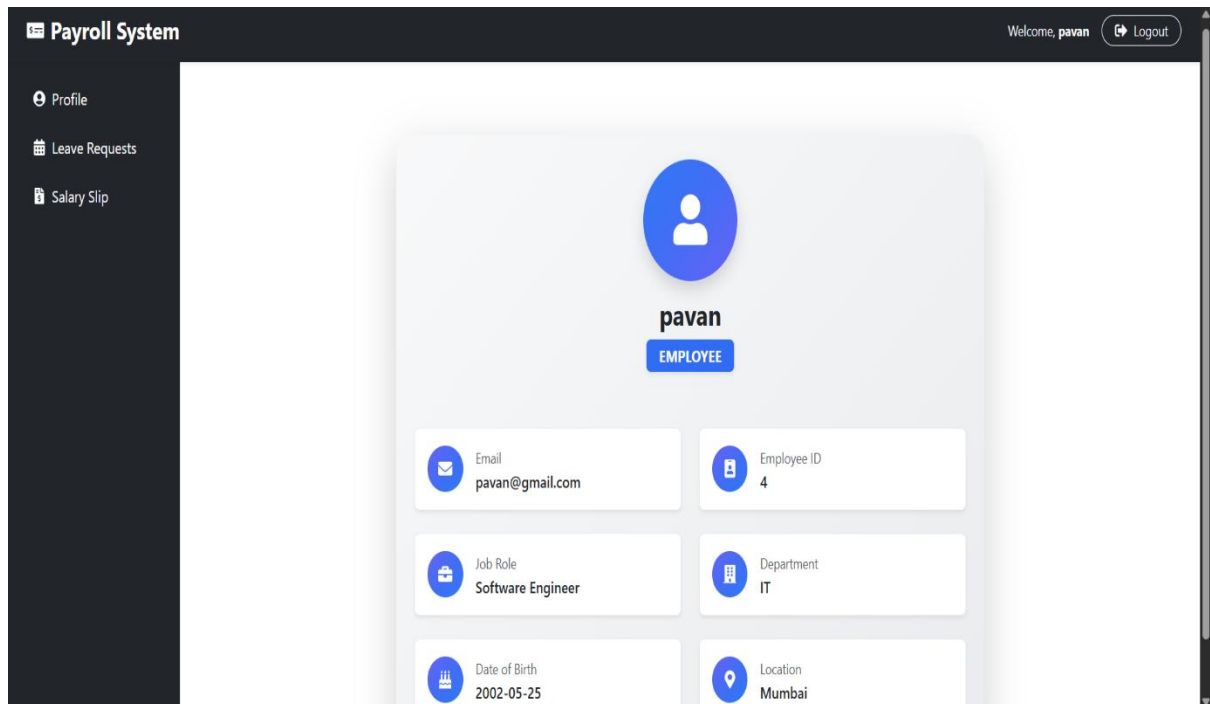
The Admin can view department-level payroll expenditure reports. These reports summarize total salary costs across different departments, helping management analyze budget distribution and cost centers.



## 4. Employee Module

### Profile Page

Employees can view their personal information, including department and job details, in a read-only format.



### Leave Application

Employees can submit leave requests by specifying the type and duration of leave. They can also track the approval status.

Employees can apply for leave by selecting the **start date** and **end date**, entering the **reason for leave**, and submitting the request.

Each leave request is tracked with a **status** (Pending, Approved, or Rejected).

Employees can view the history of their leave applications and check the decision made by the Admin.

Payroll System

Welcome, pavan 

Logout

Profile

Leave Requests

Salary Slip

My Leave Requests

Apply for Leave

Start Date

dd-mm-yyyy

End Date

dd-mm-yyyy

Leave Type

Sick Leave

Apply Leave

Leave History

#	Start Date	End Date	Leave Type	Status
1	2025-08-30	2025-09-10	sick	REJECTED
2	2025-09-25	2025-09-26	earned	APPROVED
3	2025-09-18	2025-09-20	casual	APPROVED

## Salary Slips

Employees can view their salary slips for a specific month, which include details such as basic pay, deductions, and net salary.

Payroll System

Welcome, pavan 

Logout

Profile

Leave Requests

Salary Slip

Salary Slip

Year

2025

Month

October

Fetch Salary

pavan kumar

Employee ID: 4

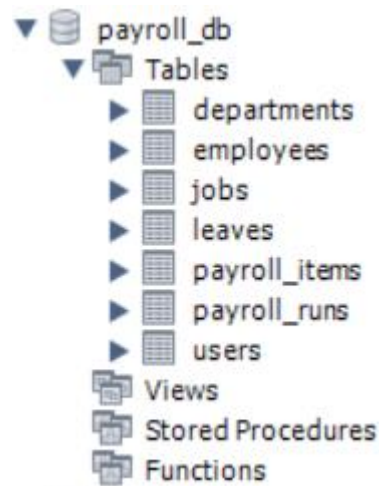
October 2025

Basic Pay	₹ 80000
Bonus	₹ 8000
Deductions	₹ 16000
Net Pay	₹ 72000

Save

## Data Base

The **Payroll Management System** uses a relational database (**MySQL**) named payroll\_db to store and manage all employee, payroll, and organizational data. The schema is designed with proper relationships to ensure data integrity, efficient querying, and support for payroll processing.



## Swagger API Documentation

The **Payroll Management System backend** is fully documented using **Swagger / OpenAPI**. This interactive documentation allows developers and testers to explore and test all REST APIs directly from the browser.

### Authentication Support

Swagger UI is configured to accept **JWT Bearer Tokens**, enabling secure testing of Admin and Employee APIs after login.

### Role-Based Access

Endpoints are grouped based on functionality (Users, Employees, Jobs, Departments, Leaves, Payroll, Reports). Admin and Employee roles have access restrictions clearly visible in Swagger.

### Interactive Testing

Developers can send API requests (GET, POST, PUT, DELETE) directly from Swagger UI and view real-time responses from the backend.











Servers









http://localhost:8080 - Generated server url

Authorize





## leave-controller

PUT	/api/v1/leaves/{leaveId}/status	 
GET	/api/v1/leaves	 
POST	/api/v1/leaves	 
GET	/api/v1/leaves/employee/{employeeId}	 












## job-controller

PUT	/api/v1/jobs/{id}	 
DELETE	/api/v1/jobs/{id}	 
GET	/api/v1/jobs	 
POST	/api/v1/jobs	 

## user-controller

POST	/api/v1/users	 
GET	/api/v1/users/me	 

## payroll-controller

GET	/api/v1/payroll/runs	 
POST	/api/v1/payroll/runs	 
POST	/api/v1/payroll/runs/{id}/process	 
POST	/api/v1/payroll/runs/{id}/lock	  
GET	/api/v1/payroll/runs/{id}/items	 
GET	/api/v1/payroll/runs/my/{year}/{month}	 

## auth-controller

POST	/api/v1/auth/login	 
------	--------------------	---

reports-controller			^
GET	/api/v1/reports/payroll/summary	🔒	▼
GET	/api/v1/reports/department-cost	🔒	▼
employee-controller			^
GET	/api/v1/employees/{id}	🔒	▼
PUT	/api/v1/employees/{id}	🔒	▼
DELETE	/api/v1/employees/{id}	🔒	▼
GET	/api/v1/employees	🔒	▼
POST	/api/v1/employees	🔒	▼
department-controller			^
PUT	/api/v1/departments/{id}	🔒	▼
DELETE	/api/v1/departments/{id}	🔒	▼
GET	/api/v1/departments	🔒	▼
POST	/api/v1/departments	🔒	▼

## API Groups

- **Auth APIs** → Login and JWT generation
- **User APIs** → Manage user registration and retrieve logged-in user details
- **Employee APIs** → CRUD operations for employee data
- **Department & Job APIs** → Manage departments and job roles
- **Leave APIs** → Submit and approve leave requests
- **Payroll APIs** → Payroll runs, processing, and salary slips
- **Reports APIs** → Payroll summary and department cost analysis