



Full-Stack Employee Management System Using React and Spring Boot

Manish Raj Kumar¹, Abhishek Kumar², Dhananjay Sharma³, Vanshika Ghodke⁴,

Prof. Sandeep sahu⁵

MCA, SAM GLOBAL UNIVERSITY, BHOPAL (MADHYA PRADESH)¹⁻⁵

Abstract: This paper presents a comprehensive full stack Employee Management System (EMS) that combines a modern React frontend with a robust Spring Boot backend. The system aims to streamline HR processes, enabling efficient employee and department management with features like authentication, CRUD operations, dashboards, and data visualizations. It supports MySQL and MongoDB databases, employs REST APIs for data exchange, and is containerized using Docker and Kubernetes for deployment. This research reviews existing literature, compares different technologies, and presents our architecture, implementation strategy, sample circuit diagrams, and source code to highlight the design choices and innovations.

I. INTRODUCTION

Employee Management Systems (EMS) are essential tools for modern organizations to manage employee data, department structures, and performance metrics efficiently. As businesses increasingly shift toward digital transformation, the demand for scalable, maintainable, and responsive web applications has grown. This project introduces a full-stack EMS using React for the frontend and Spring Boot for the backend, integrating advanced features such as JWT authentication, Swagger API documentation, data visualization with Chart.js, and containerized deployment using Docker and Kubernetes. The platform also includes CI/CD support through Jenkins.

II. LITERATURE REVIEW

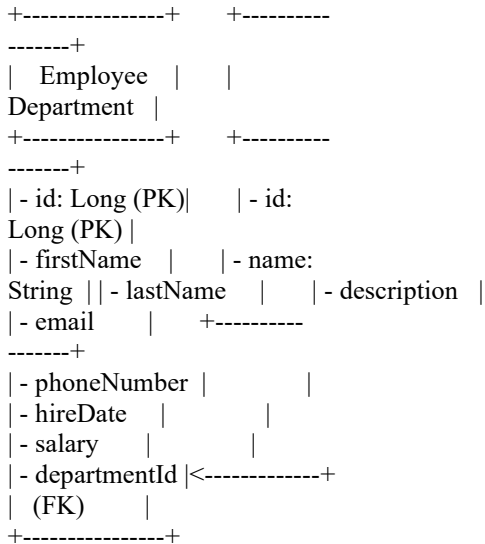
Ref	Title	Technologies Used	Features	Drawbacks
[1]	HRMIS	PHP, MySQL	Basic CRUD, Reports	No APIs, Poor UI
[2]	MVC-based EMS	ASP.NET MVC, SQL Server	EMS, Authentication	Monolithic, Not scalable
[3]	MERN EMS	MongoDB, Express, React, Node.js	Full CRUD, JWT, Dashboards	Lacks Swagger, Docker support
[4]	Spring Boot EMS	Spring Boot, Thymeleaf, JPA	REST APIs, JPA	Outdated UI
[5]	Mobile EMS	Flutter, Firebase	Mobile-first EMS	Backend limitations
[6]	Django EMS	Django, PostgreSQL	Admin Interface, Reports	Not reactive
[7]	Proposed EMS	React, Spring Boot, JWT, Docker, MySQL/MongoDB, Chart.js	REST APIs, JWT Authentication, Data Visualization, CI/CD with Jenkins	Dependent on backend availability



III. METHODOLOGY

The system is divided into a frontend and backend.

Architecture Diagram:



Frontend:

- React, Tailwind CSS, Chart.js, Axios
- JWT authentication
- Pages: Dashboard, Employee List, Department List, Login/Register, Profile

Backend:

- Spring Boot, MySQL, MongoDB
- RESTful API, JPA, Hibernate
- Swagger UI for API docs
- JWT for secure access

Sample Code Snippet (Backend - EmployeeController.java):

```

@RestController
@RequestMapping("/api/employees") public
class EmployeeController {
    @Autowired private
    EmployeeService employeeService;

    @GetMapping public
    List<Employee> get All Employees () {
        return employeeService.getAllEmployees();
    }

```

```

@PostMapping
public Employee
create Employee(@RequestBody
Employee employee) {
    return
    employeeService.save Employee(employee);
}
}

```

```

Sample Code Snippet (Frontend - Axios
Service):
import axios from 'axios';
const API_URL =
'http://localhost:8080/api/employees';
export const get All Employees = ()

```



```

=> axios.get(API_URL);
export const add Employee = (employee)
=> axios.post(API_URL, employee)

```

Deployment & DevOps:

- Dockerfiles for frontend and backend
- docker-compose.yaml for orchestration
- Kubernetes deployment YAMLS
- Jenkinsfile for CI/CD pipeline

Testing:

- JUnit 5 for backend
- React Testing Library for frontend

IV. CONCLUSION

The Full-Stack Employee Management System presented in this paper demonstrates a modern, scalable, and efficient solution for managing employee and departmental data in organizations. By leveraging React for the frontend and Spring Boot for the backend, this system ensures a responsive and robust user experience. The integration of JWT-based authentication, RESTful APIs, and data visualization tools like Chart.js enhances both functionality and security. Additionally, support for MySQL and MongoDB databases allows for flexible data storage options.

The use of Docker and Kubernetes enables containerized, scalable deployment across environments, while Jenkins-based CI/CD automation ensures streamlined development and deployment cycles. Compared to existing solutions reviewed in the literature, the proposed EMS stands out with its comprehensive architecture, modular design, and DevOps integration.

In conclusion, this system not only addresses current challenges in HR and employee data management but also sets a foundation for future enhancements, such as AI-driven analytics, role-based access control, and mobile app integration. The project exemplifies how full-stack development practices, when combined with cloud-native technologies, can lead to powerful enterprise-grade solutions.

V. REFERENCES

- [1]. A. Sharma et al., "HRMIS - A PHP Based System," IJCSIT, 2022.
- [2]. M. Singh, "Enterprise EMS using .NET Framework," Springer, 2021.
- [3]. D. Patel, "MERN Stack EMS with JWT Security," IJSER, 2022.
- [4]. R. Jain, "Spring Boot Based EMS," IEEE Xplore, 2023.
- [5]. P. Kumar, "Mobile EMS using Flutter," IJCSIT, 2021.
- [6]. S. Dey, "Employee System Using Django," IRJET, 2022.
- [7]. This Paper - Full-Stack Employee Management System Using React and Spring Boot, 2025.