Student Attendance and Performance Tracking System

1. Introduction

1.1 Project Overview

The **Student Attendance and Performance Tracking System** is a full-stack web application designed to help educational institutions efficiently manage student attendance, academic performance, and analytical insights. Traditional manual systems are time-consuming, error-prone, and lack meaningful analytics. This system digitizes the entire process and provides real-time dashboards for students, faculty, and administrators.

The application is developed using **Spring Boot** for the backend, **MySQL** as the database, and **React (Vite)** for the frontend. RESTful APIs are used for communication between the frontend and backend.

1.2 Objectives

- To automate student attendance management
- To record and track student academic performance
- To provide analytical insights using charts and dashboards
- To ensure data consistency, security, and scalability
- To reduce paperwork and manual effort

1.3 Scope of the Project

The system supports: - Student management - Attendance tracking - Marks and performance management - Role-based access (Admin/Faculty/Student) - Visual analytics using charts

This project can be extended to include features such as notifications, report generation, and mobile application support.

---

2. System Architecture and Technologies

2.1 System Architecture

The system follows a **three-tier architecture**: 1. **Presentation Layer** – React (Vite) 2. **Business Logic Layer** – Spring Boot REST APIs 3. **Data Layer** – MySQL Database

The frontend interacts with the backend through RESTful APIs using HTTP methods such as GET, POST, PUT, and DELETE.

2.2 Technology Stack

Frontend: - React.js (Vite) - JavaScript (ES6) - Axios for API communication - Recharts for analytics - CSS for styling

Backend: - Java 17 - Spring Boot - Spring Data JPA - Spring Web - Lombok

Database: - MySQL

Tools: - IntelliJ IDEA / VS Code - Postman - Git

---

3. Module Description

3.1 Student Module

- Add, update, delete, and view student details
- Each student is uniquely identified by an ID
- Stores name, email, department, and other details

3.2 Attendance Module

- Faculty can mark attendance for students
- Attendance is stored date-wise
- Supports Present/Absent status
- Students can view their attendance history

3.3 Performance Module

- Faculty can add marks for students
- Supports multiple subjects
- Calculates average and overall performance
- Displays performance trends

3.4 Analytics Dashboard

- Attendance percentage visualization
- Performance comparison using charts
- Department-wise and student-wise analysis

---

4. Database Design

4.1 Entity Description

Student Table: - id (Primary Key) - name - email - department

Attendance Table: - id (Primary Key) - student_id (Foreign Key) - date - status

Marks Table: - id (Primary Key) - student_id (Foreign Key) - subject - marks

4.2 Relationships

- One student can have multiple attendance records
- One student can have multiple marks records

The database follows normalization rules to avoid redundancy and maintain data integrity.

---

5. Functional Requirements

- Admin can manage students and faculty
- Faculty can mark attendance and enter marks
- Students can view attendance and performance
- System must provide analytical dashboards
- Data must be stored securely and reliably

6. Non-Functional Requirements

- Scalability to support large student data
- High availability and reliability
- Secure API access
- Responsive user interface

---

7. Implementation Details

7.1 Backend Implementation

- REST controllers handle client requests
- Services contain business logic
- Repositories interact with the database using JPA
- DTOs are used for clean data transfer

7.2 Frontend Implementation

- React components are modular and reusable
- Axios handles API calls
- Charts are rendered using Recharts
- State management using React Hooks

---

8. Testing

- Unit testing of backend services
- API testing using Postman
- Manual UI testing for frontend

Test cases ensure correct functionality of attendance marking, marks entry, and data visualization.

---

9. Advantages of the System

- Reduces manual workload
- Improves data accuracy
- Provides real-time analytics
- User-friendly interface
- Easy to maintain and extend

---

10. Conclusion and Future Enhancements

The Student Attendance and Performance Tracking System successfully automates academic data management and provides meaningful insights through analytics. The system is efficient, scalable, and user-friendly.

Future Enhancements: - Role-based authentication and authorization - Export reports as PDF/Excel - Email and SMS notifications - Mobile application support

---

References

- Spring Boot Documentation
- React Documentation
- MySQL Official Documentation