PROJECT REPORT

Student Course Management & Analytics System

Submitted by: *Sanatan Dev Sharma*
GitHub Repository:
https://github.com/sanatan24bsa10087-jpg/student-course-mgmt
Course: Java Project
Date: *24-11-2025*

1. Abstract

This project implements a Java-based web application for managing students, instructors, courses, enrollments, attendance, assignments and generating analytics reports.
It demonstrates practical usage of **Spring Boot**, **Java 17**, **JPA/Hibernate**, **MVC architecture**, **role-based authentication**, and **unit testing** using JUnit & Mockito.
The system provides clean UI pages, CRUD operations and a structured analytics dashboard.

2. Objectives

- Provide an efficient digital system for managing student data.
- Automate course creation and enrollment.
- Track attendance effectively.
- Generate analytics such as attendance percentage & performance reports.
- Demonstrate Java web development skills with proper architecture and documentation.

3. Functional Requirements

1. **User Management**
    a. Login, registration, roles (Admin/Instructor/Student)
2. **Course Management**

    a. Create courses
    b. Edit/Delete courses
    c. Enroll students

3. **Attendance Module**
    a. Instructor can mark attendance
    b. Students can view attendance
4. **Assignments**
    a. Instructors can upload assignments
    b. Students can submit
5. **Analytics**
    a. Attendance percentage
    b. Grade distribution
    c. Pass/Fail graph

4. Non-Functional Requirements

- Secure password hashing (BCrypt)
- Fast response time (<2s)
- Clean UI
- Modular architecture (Controller → Service → Repository)
- Database consistency using JPA/Hibernate

5. System Architecture Diagram

Architecture Flow:
Client → Controller → Service → Repository → Database (H2/MySQL)

6. UML Class Diagram

- User → Role (Many-to-One)
- Course → Enrollment (One-to-Many)
- Enrollment → Attendance (One-to-Many)
- Course → Assignment (One-to-Many)

7. Database Schema

Tables:

- users
- roles
- courses
- enrollments
- attendance
- assignments

8. Implementation

Key Java packages:

- `controller` – handles HTTP requests
- `service` – business logic
- `repository` – database operations
- `model` – JPA entity classes
- `dto` – data transfer objects

Sample Code Snippet

```
public Attendance markAttendance(Long studentId, Long courseId,
LocalDate date) {
    Enrollment e =
enrollmentRepo.findByStudentIdAndCourseId(studentId, courseId)
        .orElseThrow(() -> new IllegalArgumentException("Not
enrolled"));
    Attendance a = new Attendance(e, date, true);
    return attendanceRepo.save(a);
}
```

9. Testing

Tools Used

- JUnit 5
- Mockito

Sample Test Case

```
@Test
void testMarkAttendance() {
    Enrollment e = new Enrollment();
    when(enrollmentRepo.findByStudentIdAndCourseId(1L,
1L)).thenReturn(Optional.of(e));
    when(attendanceRepo.save(any())).thenAnswer(i ->
i.getArgument(0));

    Attendance result = attendanceService.markAttendance(1L, 1L,
LocalDate.now());

    assertNotNull(result);
}
```

11. Conclusion

The Student Course Management & Analytics System successfully demonstrates how Java, Spring Boot, and JPA can be combined to build a scalable and efficient educational management platform. The project fulfills all functional requirements and includes proper software engineering practices.