Reg no :24RP02000

Module: Backend using Java

Department: ICT

Option: Information Technology

Institution: RP KARONGI COLLEGE

On 26 February 2026

Phase 5 - Backend Project Setup Report

Repository Link: https://github.com/Etienne-2004/SMART-UNIVERSITY-DEVICES-AND-MATERIALS-MAINTENANCE-SYSTEM

1. Executive Summary

This report documents the completion of **Phase 5 (Backend Project Setup)**for the Smart University Devices and Materials Maintenance System. The primary goal of this phase was to initialize a robust, scalable backend architecture using **Spring Boot 3.x** and **Java 17**, ensuring a professional foundation for forthcoming API development and system integration.

2. Technical Implementation Details

A. Framework and Build Configuration

The project was initialized using the **Spring Boot 4.0.3** (latest stable release) framework with **Gradle** as the build automation tool. This choice ensures:

- Simplified dependency management through Gradle.
- Ready-to-use production features via Spring Boot Starters.
- High-performance execution on Java 17+ environments.

B. Standardized Package Architecture

To maintain high code quality and follow MVC (Model-View-Controller) best practices, the following package structure was implemented:

Package	Purpose
`com.example...config`	Handles application-wide configurations (Security, JPA, etc.).
`com.example...controllers`	Manages REST end-points and HTTP request mapping.
`com.example...entities`	Defines database models using Java Persistence API (JPA).
`com.example...repositories`	Data Access Object (DAO) layer using Spring Data JPA.
`com.example...services`	Houses core business logic and service implementations.
`com.example...dto`	Data Transfer Objects for optimized data communication.
`com.example...exceptions`	Custom global error and exception handling.

C. Database and Environment Configuration

The application is configured to connect to a **MySQL**database via `application.properties`.

- **Database Name:** `smart_university_maintenance_db`
- **JPA Strategy:** `update` (Ensures that database schemas are always synchronized with Java entities).
- **Default Port:**`8080`

3. Key Dependencies Included

Professional-grade libraries were integrated to support system functionality:

1. **Spring Web:** For RESTful API development.
2. **Spring Data JPA:** For seamless database interaction.
3. **MySQL Driver:** For connectivity with the local XAMPP/MySQL server.
4. **Lombok:**To reduce boilerplate code (Getters/Setters/Constructors).
5. **Spring Test:** Required for implementing Unit and Integration tests.

4. Conclusion and Next Steps

With the project foundation successfully laid out and the architecture solidified, the system is prepared for **Phase 6: Core API Developmen**. This next stage will involve the creation of database entities and the implementation of full CRUD capabilities for Universities, Devices, and Materials.