AttendEase – Complete Project Documentation

This document explains the complete architecture, database design, security model, Spring Boot fundamentals, and technical notations used in the AttendEase project. It is intended for academic submission, resume review, and interview explanation.

1. Project Overview

AttendEase is a multi-college attendance management system designed for educational institutions. It supports role-based access for Admins (HODs/Principals) and Teachers. Admins manage college-level data, while teachers manage classes and attendance within their college.

2. Key Features

- Multi-college (multi-tenant) architecture
- Role-based access control (ADMIN, TEACHER)
- Secure authentication using JWT access tokens and refresh tokens
- College join code mechanism for teacher onboarding
- Attendance tracking per class per date
- Admin approval workflow for college creation

3. System Architecture

AttendEase follows a standard client-server architecture. The frontend is built using React, which communicates with a Spring Boot REST API. Spring Boot handles authentication, authorization, business logic, and database interaction. The database is a relational system such as MySQL or PostgreSQL.

4. Authentication and Security Model

The project uses a token-based authentication model. After successful login, the backend issues two tokens:
1. Access Token (JWT): Short-lived, used to access protected APIs.
2. Refresh Token: Long-lived, stored securely and used to generate new access tokens.

JWTs are stateless and verified on every request using Spring Security filters.

5. Database Design

The database is normalized and designed to support multi-college isolation. Core tables include colleges, users, classes, students, attendance sessions, attendance records, and refresh tokens. Each table has clear ownership and foreign key relationships.

6. ER Diagram Notations Used

- PK: Primary Key, uniquely identifies a record
- FK: Foreign Key, references another table
- 1 : N relationship: One-to-many relationship

- M : N relationship: Many-to-many relationship using a join table
- ENUM: Fixed set of allowed values (e.g., role, status)
- Composite Key: Combination of multiple columns as a primary key

7. Spring Boot Fundamentals

Spring Boot is a framework built on top of the Spring Framework that simplifies backend development. It provides auto-configuration, embedded servers, and dependency management. Key concepts include Dependency Injection (DI), Inversion of Control (IoC), and Beans.

8. How Spring Boot Works

When a Spring Boot application starts, it initializes the Spring Application Context, scans components, creates beans, and applies auto-configuration based on dependencies. Embedded Tomcat runs the application, and incoming HTTP requests are handled by controllers, processed through filters, services, and repositories.

9. Authorization Logic

Authorization is enforced at the backend using roles and college scope. Admins can access all data belonging to their college. Teachers can access only their own classes and attendance records. All access rules are enforced server-side using Spring Security.

10. Conclusion

AttendEase is designed to reflect real-world backend architecture practices. It demonstrates secure authentication, proper database modeling, and scalable multi-tenant design. This project is suitable for academic evaluation and professional portfolio presentation.