

Weekly Report 1

Report Period: 9 – 15 Dec 2025

Prepared By: 21 Dec 2025

Reporter and Team Lead: Ang Panha

Summary:

We set up the spring boot project with Maven, added required dependencies, and initialized basic security configurations.

Task / Objective / Achievement:

Task	Assignee	Status	Integrated
Project Setup	Ang Panha	Done	Done
Security Implementation	Et Anchhy	Done	Late

Note: Members' achievements are displayed in the table above via rows. The team lead's achievements may be displayed there too, but for the task **Code Integration**, that is mandatory for all members' commits and PRs, it will be displayed as a status column called '**Integrated**'.

Details

- Project Setup (by Ang Panha):
 - Set up springboot via Spring Initializr:
 - Build: Maven
 - Java: 21 LTS
 - Springboot: 3.5.8
 - Group: `edu.itc`
 - Artifact: `enrollment_scheduling_system`
 - Name: EnrollmentSchedulingSystem
 - Packaging: Jar
 - Configuration: Properties
 - Added dependencies:
 - Spring Framework:
 - Spring Boot Starter Web
 - Spring Boot Starter Actuator
 - Spring Boot Starter Data JPA

- Spring Boot Starter Security
 - Spring Boot Starter Thymeleaf
 - Spring Boot Starter Validation
 - Spring Boot Devtools
 - Spring Boot Starter Test
 - Spring Security Test
 - Flyway:
 - Flyway Core
 - Flyway MySQL
 - Thymeleaf: Thymeleaf Extra Spring Security 6
 - MYSQL: MySQL Java Connector
 - Lombok
 - H2 Database
- Designed the initial project layout:
- Note: `root` is at '`src/main/java/edu/itc/enrollment_scheduling_system'`
- `<root>/config/`
 - `<root>/controller/`
 - `<root>/dto/`
 - `<root>/model/`
 - `<root>/model/Classroom.java`
 - `<root>/model/Course.java`
 - `<root>/model/Enrollment.java`
 - `<root>/model/Role.java`
 - `<root>/model/Schedule.java`
 - `<root>/model/User.java`
 - `<root>/repository/`
 - `<root>/service/`
- Security Implementation (by Et Anchhy):
 - Configurations:
 - Security Config: configures HTTP request filter and password encryption.
 - Data Initializer: implements `CommandLineRunner` to populate default roles, users, and password encoder at application startup.
 - Role-based Success Handler: implements `AuthenticationSuccessHandler` to redirect users to pages corresponding to their roles.
 - Model:
 - Role Entity: populates this entity with fields and methods.
 - User Entity: populates this entity with fields and methods.
 - Repositories:
 - Role Repository: extends JPA repository for role-specific behaviors.
 - User Repository: extends JPA repository for user-specific behaviors.

- Services:
 - User Details Service: implements the user details service interface.
 - Registration Service: defines the registration service for newcomers.
- Controllers:
 - Authentication Controller: route users to the authentication pages.
 - Admin Controller: route users to the admin dashboard and related pages.
- DTOs:
 - Registration Form: holds the form validation used in the authentication pages.
- Thymeleaf Pages:
 - Authentication Pages:
 - Login Page
 - Register Page
 - Dashboards
 - Admin Dashboard
 - Student Dashboard
 - Teacher Dashboard