INSTITUTION OF TECHNOLOGY OF CAMBODIA

DEPARTMENT: GIC I4-B

Report of Software Engineering

Project 3: Course Enrollment and Classroom Scheduling System

Name of Students	Students ID
1. PANG Lythong	e20220161
2. SOEURY Sreyno	e20220908
3. VICHETH Sokhsedtha	e202201549
4. NGET Darapich	e20221646
5. KEO Chanponlork	e20220660

Lecturer: ROEUN Pacharoth

Academic year 2025-2026

1. Security and Authentication Lead (Vichet Soksedtha)

Goal: You are the gatekeeper. You ensure only the right people get in and see the right things.

- **Key Entities:** `User`, `Role`, `Privilege`.
- **Specific Tasks:**
  - **Spring Security Config:** Create the `SecurityConfig.java` class. Configure the `SecurityFilterChain` to define which URLs are public (e.g., `/login`, `/css/**`) and which are private.
  - **User Management:** Create the `UserDetailsService` implementation to load users from the database.
  - **Roles:** Implement 3 distinct roles:
    - `ROLE_ADMIN`: Can create courses, assign lecturers, and manage classrooms.
    - `ROLE_LECTURER`: Can view their assigned courses and see the student list.
    - `ROLE_STUDENT`: Can browse courses and enroll.
  - **Registration:** Build the logic to register new accounts (encrypting passwords using `BCryptPasswordEncoder`).
- **Deliverables:** Login page, Registration page, and "Access Denied" error handling.

2. Main Entity CRUD Lead (The Core Data) (Keo Chanponlok)

Goal: You manage the "nouns" of the system—the physical things that exist regardless of the schedule.

- **Key Entities:** `Course`, `Classroom`, `Department` (optional).
- **Specific Tasks:**
  - **Course Management:**
    - Create `CourseController`, `CourseService`, and `CourseRepository`.
    - Fields needed: `courseName`, `courseCode` (e.g., CS101), `credits`, `description`, `capacity`.
    - **Validation:** Ensure `courseCode` is unique and `capacity` is a positive number.
  - **Classroom Management:**
    - Create `ClassroomController` etc.
    - Fields needed: `roomNumber`, `building`, `maxCapacity`.
  - **Lecturer Profile:** (If not handled by Security) Create a `Lecturer` entity to store specific details like "Department" or "Office Hours."

- **Deliverables:** Admin pages to "Add New Course," "Edit Classroom," and "List All Courses."

3. Secondary Module CRUD Lead (The Actions/Logic) (Nget Darapich)

Goal: You manage the "verbs"—the relationships and actions that happen between the main entities. This is the hardest logic part.

- **Key Entities:** `Enrollment`, `ClassSchedule`.
- **Specific Tasks:**
  - **Scheduling Logic (Crucial):**
    - Link a `Course` to a `Classroom` at a specific `DayOfWeek` and `Time`.
    - **Validation:** You must write logic to prevent **double booking**. (e.g., "Error: Room 304 is already occupied on Monday at 9 AM").
  - **Enrollment Logic:**
    - Create the "Enroll" button logic.
    - **Validation:** Check if the course is full (`currentStudents < capacity`). Check if the student is already enrolled.
  - **Views:** Logic for "My Schedule" (showing a student only *their* classes).
- **Deliverables:** The "Class Schedule" view and the "My Enrolled Courses" dashboard.

4. Frontend/Thymeleaf Lead (Soeury Sreyno)

Goal: You make the application usable and ensure the UI adapts to the user.

- **Key Technologies:** Thymeleaf, HTML5, CSS (Bootstrap or Tailwind), JavaScript.
- **Specific Tasks:**
  - **Master Layout:** Create a `layout.html` fragment (Header, Footer, Sidebar) so every page looks consistent.
  - **Role-Based UI:** Use `sec:authorize` tags to hide buttons.
    - *Example:* Only show the "Delete Course" red button if the user is an `ADMIN`.
    - *Example:* Show "Enroll" button only to `STUDENT` users.
  - **Feedback:** Design alert boxes for success messages ("Enrolled successfully!") and error messages ("Course is full!").
  - **Mockups:** You often need to write the HTML *before* the CRUD leads finish their logic so they have a template to work with.
- **Deliverables:** All `.html` templates in the `src/main/resources/templates` folder.

5. Database Lead (Pang Lythong)

Goal: You provide the foundation. If you change the database halfway through, everyone else breaks, so you must plan early.

- **Key Technologies:** MySQL/PostgreSQL, Flyway, JPA Relationships.
- **Specific Tasks:**
    - **ER Diagram:** Draw the map.
        - *One-to-Many:* One `Course` has many `ClassSchedule` entries.
        - *Many-to-Many:* `Students` and `Courses` are linked via the `Enrollment` table.
    - **Flyway Migrations:**
        - `V1__Create_Tables.sql`: The initial script to create all tables.
        - `V2__Insert_Dummy_Data.sql`: Add 5 dummy courses, 2 classrooms, and 3 users so the team has data to test with immediately.
    - **Optimization:** Ensure columns like `email` or `course_code` have `UNIQUE` constraints.
- **Deliverables:** The `db/migration` folder content and the ER Diagram image for the final report.