



## Lab 7

### Skill Forge

#### **Objective**

**The goal of this phase is to implement all functionalities for Students and Instructors, focusing on course creation, learning progress, and lesson interaction.**

**The system should support teaching and learning workflows before adding administrative oversight.**

#### **1. User Account Management**

##### **Backend (Java)**

- **Implement signup, login, and logout for Students and Instructors.**
- **Validate inputs (email format, required fields).**
- **Hash passwords (SHA-256).**
- **Identify user roles upon login.**

##### **Frontend (Swing)**

- **Create signup and login forms with validation and error handling.**
- **Show role-based dashboards:**
  - **Student: browse/enroll in courses.**
  - **Instructor: create/manage courses.**
- **Add logout functionality.**

---

**Dr. Layla Abou-Hadeed**

Eng. Ahmed ElSayed

Eng. Ahmed Ashraf

Eng. Miar Mamdouh

Eng. AbdElaziz Mohamed

Eng. Mazen Sallam

Eng. Shams Zayan

Eng. Abdelrahman Wael

Eng. Mohamed Zaytoon

Eng. Muhamad Bashar



## 2. Instructor Management

### Backend

- **Manage instructor data (userId, role, username, email, passwordHash, createdCourses).**
- **Implement:**
  - Course creation and editing.
  - Lesson add/edit/delete.
  - View enrolled students.
- **Persist data in users.json and courses.json.**

### Frontend

- **Instructor Dashboard:**
  - Create, edit, and delete courses.
  - Manage lessons.
  - View enrolled students.

---

**Dr. Layla Abou-Hadeed**

Eng. Ahmed ElSayed  
Eng. Miar Mamdouh  
Eng. Mazen Sallam  
Eng. Abdelrahman Wael  
Eng. Muhamnad Bashar

Eng. Ahmed Ashraf  
Eng. AbdElaziz Mohamed  
Eng. Shams Zayan  
Eng. Mohamed Zaytoon



### 3. Student Management

#### Backend

- **Manage student data (userId, role, username, email, passwordHash, enrolledCourses, progress).**
- **Implement:**
  - Course browsing and enrollment.
  - Lesson access and progress tracking.

#### Frontend

- **Student Dashboard:**
  - View available and enrolled courses.
  - Enroll courses and access lessons.
  - Mark lessons as completed.

### 4. Course Management

#### Backend

- **Manage courses with attributes: courseId, title, description, instructorId, lessons[], students[].**
- **Allow instructors to create/edit and students to browse/enroll.**
- **Save data in courses.json.**

#### Frontend

- **Instructor: manage courses and lessons, view students.**
- **Student: browse/enroll, view lessons.**

---

**Dr. Layla Abou-Hadeed**

Eng. Ahmed ElSayed

Eng. Ahmed Ashraf

Eng. Miar Mamdouh

Eng. AbdElaziz Mohamed

Eng. Mazen Sallam

Eng. Shams Zayan

Eng. Abdelrahman Wael

Eng. Mohamed Zaytoon

Eng. Muhamad Bashar



## 5. Lesson and Learning Features

### Backend (Java)

- Implement the Lesson class with attributes:
  - lessonId, title, content, and optional resources[].
- Provide methods to:
  - Fetch lessons by course.
- Save progress and updates in courses.json .

### Frontend (Swing)

- Display lessons.
- Mark lessons as *completed*.

## 6. Frontend Structure (Swing UI)

LoginFrame	Authenticate users and detect roles.
SignupFrame	Register new users with validation.
StudentDashboardFrame	Browse/enroll in courses, view lessons.
InstructorDashboardFrame	Create/manage courses and lessons.

## 7. Database Files (JSON-Based)

users.json	Stores Student and Instructor data.
courses.json	Stores courses and related lessons.

## 8. Security and Validation

- All passwords stored **hashed (SHA-256)**.

<b>Dr. Layla Abou-Hadeed</b>	Eng. Ahmed ElSayed	Eng. Ahmed Ashraf
	Eng. Miar Mamdouh	Eng. AbdElaziz Mohamed
	Eng. Mazen Sallam	Eng. Shams Zayan
	Eng. Abdelrahman Wael	Eng. Mohamed Zaytoon
	Eng. Muhamad Bashar	



- Validate all user inputs on forms.
- Prevent duplicate course or user IDs through validation in JsonDatabaseManager.

**Required:**

1. You are required to obey the OOP concepts (inheritance, polymorphism, abstraction, ...).
2. A discussion will be made with you at your lab next week on what you delivered.
3. The deadline for the delivery on the form is Sunday 26/10/2025.

**What to be delivered**

1. On the form, you should deliver a zipped file that contains the .java files of your classes.
2. Your zip file should be named as id1\_id2\_id3\_id4\_groupNumber. For example, 4678\_4557\_4558\_4559\_G2.

**Policies:**

1. You should work in groups of four (Same Groups unless you get permission to change).
2. Cheating will be severely penalized, so delivering nothing is so much better than cheating.
3. No late submission is allowed.
4. Each group must use GitHub throughout their lab work.

---

**Dr. Layla Abou-Hadeed**

Eng. Ahmed ElSayed

Eng. Ahmed Ashraf

Eng. Miar Mamdouh

Eng. AbdElaziz Mohamed

Eng. Mazen Sallam

Eng. Shams Zayan

Eng. Abdelrahman Wael

Eng. Mohamed Zaytoon

Eng. Muhamad Bashar