# **Spring Advanced – June 2024** **Individual Project Assignment**

**Overview**

This is the Individual Project Assignment for the [Spring Advanced Course @ SoftUni](https://softuni.bg/trainings/4532/spring-advanced-june-2024) Your web application must meet the following general and additional requirements.

**General Requirements**

Your web application should use the following technologies, frameworks, and development techniques:

**Core Technologies**

* **Spring Framework**
  + The application must have at least:
    - 12 web pages
    - 5 independent entity models
    - 5 controllers
    - 1 Rest Controller
    - 5 services
    - 5 repositories
  + Implement a separate service in a distinct project, which will be consumed by your main project through a REST API.
    - This service must include at least three endpoints: GET, POST, DELETE.
* **Database**
  + Use MySQL, Oracle, PostgreSQL, or MariaDB.
  + Access the database using Spring Data.
  + Use Hibernate or any other JPA provider.
* **Security**
  + Use standard Spring Security for managing users and roles.
  + Roles: user and administrator.
  + Ensure role management is secured and error-safe.
  + *For Retake:* Users and administrators should be able to edit their usernames.
* **Validation and Error Handling**
  + Implement client-side and server-side validation.
  + Display appropriate validation messages to the user.
* **Internationalization (i18n)**
  + Support multiple languages.
* **Scheduling**
  + Implement scheduled jobs affecting the application, e.g., once/twice a day.
* **Mapping**
  + Use MapStruct, ModelMapper or another mapping library.
* **Testing**
  + Write Unit & Integration tests for logic, services, repository query methods, helpers, etc.
  + Achieve at least 60% coverage on business logic (Line Coverage).
  + *For Retake:* Achieve at least 70% coverage on business logic (Line Coverage).
* **Front-end Design**
  + Ensure a visually appealing and intuitive front-end design for an good user experience (UX).
  + Use the Thymeleaf template engine or a JavaScript framework/library such as React, Angular, or Vue.js, consuming REST services from a Web API.

**Additional Requirements**

* **Object-Oriented Design**
  + Follow best practices for high-quality code:
    - Data encapsulation
    - Proper exception handling
    - Appropriate use of inheritance, abstraction, and polymorphism
    - Strong cohesion and loose coupling principles
    - Well-formatted and structured code with readable identifiers
    - Thin controllers concept
* **User Interface (UI)**
  + Ensure a well-designed UI.
* **User Experience (UX)**
  + Ensure a good UX.

**Source Control**

* Use a source control system like GitHub, GitLab or BitBucket.
  + Submit a link to your public source code repository.
  + Commit on at least 5 different days.
  + Make at least 20 commits.

**IMPORTANT:** The Source Control Requirements are **MANDATORY**. Failure to follow these requirements will result in **DIRECT DISQUALIFICATION** from the Project Defenses.

**Submission Deadline**

* **Deadline:** Submit your project before **23:59** on **27-July-2024** using a survey that will be provided on **20-July-2024**.
* **Presentation Schedule:** A schedule will be available on 01-August-2024 and will include only the projects submitted on time. Non-submitted projects will NOT be evaluated.
* **Working Schedule:** You can work on your project before **23:59** on **7th August 2024**.

**Online Project Defense**

Each student must deliver an online defense of their work in front of a trainer jury. Students will have 20 minutes to:

1. Demonstrate how the application works (briefly).
2. Show and explain the source code.
3. Answer questions from the jury.

**Note:** Be strict with timing! On the 20th minute, you will be interrupted. Be well-prepared to present the maximum of your work in the minimum time.

**Assessment Criteria**

* **General Requirements (85%)**
  + Functionality – 0…35
  + Implementing controllers correctly – 0...5
  + Implementing views correctly – 0…5
  + Testing (unit and integration tests for controllers using mocking) – 0…10
  + Security (preventing SQL injection, XSS, CSRF, parameter tampering, etc.) – 0…5
  + Data validation (models and input models) – 0…10
  + Using mapper and inversion of control – 0…5
  + Using layers with multiple layouts – 0…10
  + Code quality (well-structured code, following MVC pattern, SOLID principles, etc.) – 0…15
* **Answering Questions (15%)**
  + Answer 3 questions related to the project (and best practices in general) or theoretical common java question
* **Bonuses (up to 15%)**
  + Use Spring Event in your application.
  + Implement one or more Advice (AOP).
  + Implement HATEOAS.
  + Use Spring WebFlux.
  + Use Apache Kafka.
  + Use Angular/React/Vue for the front-end.
  + Host the application in a cloud environment e.g., Azure.
  + Use a file storage cloud API, e.g., Cloudinary, Dropbox, Google Drive, etc.
  + Implement Microservice architecture in your application.
  + Any additional feature with practical use not described in the assignment.