

Kërkesat e detyrës

Web Aplikacion me Spring Boot & Thymeleaf

1. Përshkrimi i përgjithshëm

Studentët të ndarë në **grupe** prej **3-4 persona** duhet të zhvillojnë një **aplikacion web server-side** duke përdorur **Spring Boot dhe Thymeleaf**, i cili ofron funksionalitete të plota CRUD, autentifikim dhe autorizim me role, validim të avancuar të të dhënave, një endpoint REST të integruar me UI, si dhe një strukturë të rregullt të layout-it përmes **Thymeleaf Layout & Fragments**.

Aplikacioni duhet të përdorë **Spring Security**, të lidhet me një **databazë në Docker**, dhe të respektojë arkitekturën me shtresa (Controller → Service → Repository).

2. Teknologjitë e detyrueshme

- Java 17+
- Spring Boot 3.x
- Spring MVC
- Thymeleaf
- Thymeleaf Fragments & Layout
- Spring Security (Form Login)
- Spring Data JPA
- PostgreSQL, MySQL ose MSSQL (në Docker)
- Bean Validation (jakarta.validation)
- Docker & Docker Compose

3. Kërkesat funksionale

3.1 Autentifikim & Autorizim (Spring Security)

1. **Custom Login Page** (jo login default)
2. Funksionaliteti i **Logout**
3. **Role-based access control**
 - `ROLE_USER`
 - `ROLE_ADMIN`
4. Kufizim i URL-ve:
 - `/admin/**` → vetëm ADMIN
 - `/app/**` → USER & ADMIN
5. **Custom Access Denied Page** (`/access-denied`)

3.3 CRUD & Search / Filter (Server-side)

1. Të paktën **1 entitet kryesor**
2. CRUD i plotë nga UI:
 - List
 - Create
 - Edit
 - Delete
3. **Search ose filter** sipas së paku **2 kriterëve**
4. Validim i formave + shfaqje e gabimeve në Thymeleaf

3.4 Relacionet (lidhjet) në Databazë

- Të paktën **1 relacion i detyrueshëm**:
 - `@OneToMany` **ose**
 - `@ManyToMany`
- Relacioni duhet të përdoret në UI

3.5 Custom Validator

Duhet të implementoni të paktën 1 Custom Validation Annotation (@Constraint + ConstraintValidator).

- Duhet të përdoret në Create/Edit
- Mesazhi i gabimit duhet të shfaqet në UI

Shembuj:

- @UniqueEmail
- @ValidISBN
- @StrongPassword
- @UniqueCode
- @FutureDate

3.6 REST Endpoint + AJAX

1. 1 REST endpoint
2. Thirrje nga Thymeleaf (fetch / AJAX)

Shembuj:

- live search
- statistika
- availability check

4. Layout & Fragments (Thymeleaf)

Qëllimi

Të shmangët duplikimi i HTML-it dhe të përdoret një **layout i përbashkët** për të gjitha faqet.

4.1 Struktura e rekomanduar e templates

```
templates/  
  shared/  
    layout.html  
    header.html  
    footer.html  
  auth/  
    login.html  
  product/  
    index.html  
    add.html
```

5. Struktura e rekomanduar e projektit

```
com.example.app  
  configs/  
    SecurityConfig.java  
  controllers/  
    HomeController.java  
    AuthController.java  
    ProductController.java  
    ProductRestController.java  
  services/  
    ProductService.java  
  repositories/  
    ProductRepository.java  
  entities/  
    Product.java  
    Category.java  
    User.java  
    Role.java  
  validations/  
    CustomAnnotation.java  
    CustomValidator.java  
  resources/  
    templates/  
      shared/  
        auth/  
        product/
```

Shembuj Tema:

1. Address Book (Menaxhimi i Kontakteve)

Aplikacion për menaxhimin e kontakteve personale ose biznesore.

Funksionalitetet

- CRUD kontakte: emër, mbiemër, telefon, email, adresë
- Search sipas emrit, mbiemrit, email-it, telefonit
- ADMIN mund të fshijë kontakte

Relacion DB

- `Category` → `Contact` (OneToMany)

Custom Validator

- `@UniqueEmail` për kontaktin

REST Endpoint

- `/api/contacts/search?query=...` (live search nga UI)

2. Videotekë Online

Funksionalitetet

- CRUD filma (titull, zhanër, vit, aktor kryesor)
- Search sipas titullit, zhanrit, aktorit

Relacion DB

- `Movie` ↔ `Actor` (ManyToMany)

Custom Validator

- `@ValidReleaseYear`

REST Endpoint

- `/api/movies/filter?genre=...`

3. Librari Muzikore

Funksionalitetet

- CRUD këngë & albume
- Search sipas këngëtarit, zhanrit, albumit

Relacion DB

- `Album → Song (OneToMany)`

Custom Validator

- `@UniqueSongTitlePerAlbum`

REST Endpoint

- `/api/songs/search?artist=...`

4. Librari Librash

Funksionalitetet

- CRUD libra (titull, autor, ISBN, gjini)
- Search sipas titullit, autorit

Relacion DB

- `Author → Book (OneToMany)`

Custom Validator

- `@ValidISBN ose @UniqueISBN`

REST Endpoint

- `/api/books/check-isbn?isbn=...`

5. Reklamim Produktesh (Mini Catalog)

Funksionalitetet

- CRUD produkte (emër, kategori, çmim, stok)
- Search sipas emrit dhe kategorisë

Relacion DB

- `Category → Product (OneToMany)`

Custom Validator

- `@PositivePrice`

REST Endpoint

- `/api/products/check-stock/{id}`

6. Receta Gatimi

Funksionalitetet

- CRUD receta (emër, kategori, koha, përshkrimi)
- Search sipas kohës, kategorisë

Relacion DB

- `Recipe → Ingredient (OneToMany)`

Custom Validator

- `@MaxCookingTime`

REST Endpoint

- `/api/recipes/filter?maxTime=...`

7. Inventar

Funksionalitetet

- CRUD inventar (kod, emër, sasi, lokacion)
- Search sipas kodit dhe kategorisë

Relacion DB

- `Warehouse` → `Item` (OneToMany)

Custom Validator

- `@UniqueItemCode`

REST Endpoint

- `/api/items/check-code?code=...`

8. Help Desk

Funksionalitetet

- CRUD tickets (titull, kategori, datë, status)
- USER krijon ticket
- ADMIN ndryshon status

Relacion DB

- `User` → `Ticket` (OneToMany)

Custom Validator

- `@ValidTicketStatus`

REST Endpoint

- `/api/tickets/stats`

9. Reklamim Barnash (Pharmacy Catalog)

Funksionalitetet

- CRUD barna (emër, kategori, përshkrim)
- Search sipas emrit dhe kategorisë

Relacion DB

- `Category` → `Medicine` (OneToMany)

Custom Validator

- `@AllowedMedicineCategory`

REST Endpoint

- `/api/medicines/filter?category=...`

10. Reklamim Kurseve

Funksionalitetet

- CRUD kurse (emër, mësimdhënës, orar)
- Search sipas emrit dhe mësimdhënësit

Relacion DB

- `Instructor` → `Course` (OneToMany)

Custom Validator

- `@ValidCourseSchedule`

REST Endpoint

- `/api/courses/search?name=...`

11. Menaxhim i Shpenzimeve

Funksionalitetet

- CRUD shpenzime (datë, kategori, shumë)
- Search sipas datës dhe kategorisë

Relacion DB

- `Category` → `Expense` (OneToMany)

Custom Validator

- `@PositiveAmount`

REST Endpoint

- `/api/expenses/stats`

12. Reklamim Mobilieri

Funksionalitetet

- CRUD produkte mobilieri
- Search sipas kategorisë dhe çmimit

Relacion DB

- `Category` → `Furniture` (OneToMany)

Custom Validator

- `@ValidDimensions`

REST Endpoint

- `/api/furniture/filter`

13. Autosallon

Funksionalitetet

- CRUD vetura
- Search sipas modelit, fuqisë, dyerve

Relacion DB

- Brand → Car (OneToMany)

Custom Validator

- @ValidHorsePower

REST Endpoint

- /api/cars/filter

14. Pajisje IT

Funksionalitetet

- CRUD pajisje IT
- Search sipas prodhuesit dhe llojit

Relacion DB

- Brand → Device (OneToMany)

Custom Validator

- @ValidSpecification

REST Endpoint

- /api/devices/search