JAVA SPRING FRAMEWORK

Lab Guides

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.0
Effective Date	01/09/2024

Hanoi, 08/2024

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	06/08/2024	Create a new Lab	Create new		VinhNV

Contents

CODE:	**JSFW_Lab_07_Opt2**
TYPE:	**MEDIUM**
LOC:	**200**
DURATION:	**120 MINUTES**

Java Spring Framework Introduction

Objectives:

- Implement CRUD operations for the Subject entity.
- Use DAO pattern to manage subject data.
- Extend the web interface for subject management using Thymeleaf..

Lab Specifications:

Trainees are required to:

- Create a `Subject` entity.
- Implement CRUD operations using `SubjectRepository`.
- Create Thymeleaf views to manage subjects (create, read, update, and delete)..

Problem Description:

Trainees are required to:

- Implement user signup functionality.
- Implement user login.
- Use Thymeleaf for the user interface (UI).

Prerequisites:

- Completed **JSFW_Lab_07_Opt1**.

Guidelines:

Step 1: Create Subject Entity

Define Subject Class: In com.example.model, create Subject:

```
package com.example.model;

import javax.persistence.*;


@Entity
public class Subject {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;

    // Getters and Setters

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

Step 2: Create SubjectRepository:

In com.example.repository, create the SubjectRepository:

```
package com.example.repository;

import com.example.model.Subject;
import org.springframework.data.jpa.repository.JpaRepository;

public interface SubjectRepository extends JpaRepository<Subject, Long> {
}
```

Step 3: Create SubjectService:

In com.example.service, create a SubjectService.java:

```
package com.example.service;

import com.example.model.Subject;
import com.example.repository.SubjectRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class SubjectService {

    @Autowired
    private SubjectRepository subjectRepository;

    public List<Subject> findAll() {
        return subjectRepository.findAll();
    }

    public Subject save(Subject subject) {
        return subjectRepository.save(subject);
    }

    public Subject findById(Long id) {
        return subjectRepository.findById(id).orElse(null);
    }

    public void deleteById(Long id) {
        subjectRepository.deleteById(id);
    }
}
```

Step 4: Create SubjectController

In com.example.controller, create a SubjectController.java:

```
package com.example.controller;

import com.example.model.Subject;
import com.example.service.SubjectService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

@Controller
@RequestMapping("/subjects")
public class SubjectController {

    @Autowired
    private SubjectService subjectService;

    @GetMapping
    public String listSubjects(Model model) {
        model.addAttribute("subjects", subjectService.findAll());
        return "subjects/list";
    }

    @GetMapping("/new")
    public String createSubjectForm(Model model) {
        model.addAttribute("subject", new Subject());
        return "subjects/form";
    }

    @PostMapping
    public String saveSubject(@ModelAttribute Subject subject) {
        subjectService.save(subject);
        return "redirect:/subjects";
    }

    @GetMapping("/edit/{id}")
    public String editSubjectForm(@PathVariable Long id, Model model) {
        Subject subject = subjectService.findById(id);
        if (subject != null) {
            model.addAttribute("subject", subject);
            return "subjects/form";
        }
        return "redirect:/subjects";
    }

    @PostMapping("/update/{id}")
    public String updateSubject(@PathVariable Long id, @ModelAttribute Subject
subject) {
        subject.setId(id);
        subjectService.save(subject);
        return "redirect:/subjects";
    }
    @GetMapping("/delete/{id}")
    public String deleteSubject(@PathVariable Long id) {
        subjectService.deleteById(id);
        return "redirect:/subjects";
    }
}
```

Step 5: Create Thymeleaf Templates

Create the following templates in src/main/resources/templates/subjects:

a. **list.html - List of Subjects:**

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Subjects</title>
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h2 class="text-center">Subject List</h2>
    <!-- Back to Dashboard button -->
    <a href="/dashboard" class="btn btn-secondary mb-3">Back to Dashboard</a>
    <a href="/subjects/new" class="btn btn-primary mb-3">Add New Subject</a>

    <!-- Check if there are any subjects -->
    <div th:if="${subjects.isEmpty()}">
        <p class="text-center text-muted">Don't have any data yet.</p>
    </div>

    <!-- Display table if subjects exist -->
    <table class="table table-striped" th:if="${!subjects.isEmpty()}">>
        <thead>
        <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Actions</th>
        </tr>
        </thead>
        <tbody>
        <tr th:each="subject : ${subjects}">
            <td th:text="${subject.id}">1</td>
            <td th:text="${subject.name}">Math</td>
            <td>
                <a th:href="@{'/subjects/edit/' + ${subject.id}}" class="btn
btn-warning">Edit</a>
                <a th:href="@{'/subjects/delete/' + ${subject.id}}" class="btn
btn-danger" onclick="return confirm('Are you sure you want to delete this
subject?');">Delete</a>
            </td>
        </tr>
        </tbody>
    </table>
</div>

<!-- Bootstrap JS and dependencies -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
```

```
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.0.7/dist/umd/popper.min.js"><
/script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></s
cript>
</body>
</html>
```

b. form.html: Subject Form (Create/Update):

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create/Edit Subject</title>
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h2 th:text="${#strings.isEmpty(subject.id) ? 'Create Subject' : 'Edit
Subject'}"></h2>
    <form th:action="@{/subjects}" th:object="${subject}" method="post">
        <div class="form-group">
            <label for="name">Subject Name</label>
            <input type="text" th:field="*{name}" class="form-control" id="name"
required>
        </div>
        <button type="submit" class="btn btn-primary">Save</button>
        <a href="/subjects" class="btn btn-secondary">Cancel</a>
    </form>
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.0.7/dist/umd/popper.min.js"><
/script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></s
cript>
</body>
</html>
```

Setup 6: Update dashboard.html:

Add Subject function to dashboard:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>

<!-- Navigation Bar -->
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Dashboard</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ml-auto">
            <li class="nav-item active">
                <a class="nav-link" href="#">Home</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/profile">Profile</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/subjects">Subjects</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/logout">Logout</a>
            </li>
        </ul>
    </div>
</nav>

<!-- Main Content -->
<div class="container mt-5">
    <div class="row">
        <div class="col-md-12 text-center">
            <h1>Welcome, <span th:text="${username}">User</span>!</h1>
            <p>This is your dashboard where you can manage your profile, view
modules, and more.</p>
        </div>
    </div>

    <!-- Cards Section -->
    <div class="row mt-4">
        <div class="col-md-3">
            <div class="card">
                <img src="https://cdn-icons-
png.flaticon.com/512/3135/3135715.png" class="card-img-top" alt="Profile">
                <div class="card-body">
                    <h5 class="card-title">Profile</h5>
                    <p class="card-text">Manage your profile settings - Under
construction.</p>
                    <a href="/profile" class="btn btn-primary">Go to Profile</a>
                </div>
            </div>
        </div>
        <div class="col-md-3">
            <div class="card">
                <img src="https://cdn-icons-
png.flaticon.com/512/4136/4136899.png" class="card-img-top" alt="Modules">
                <div class="card-body">
                    <h5 class="card-title">Modules</h5>
                    <p class="card-text">Manage your modules - Under
construction.</p>
                    <a href="/modules" class="btn btn-primary">Go to Modules</a>
                </div>
            </div>
        </div>
        <div class="col-md-3">
            <div class="card">
                <img src="https://cdn-icons-
png.flaticon.com/512/4789/4789715.png" class="card-img-top" alt="Subjects">
                <div class="card-body">
```

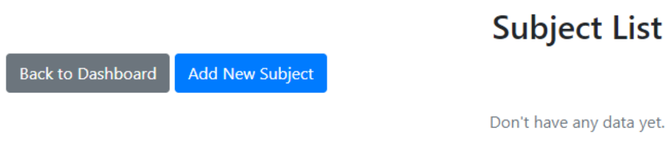
```
                    <h5 class="card-title">Subjects</h5>
                    <p class="card-text">Manage and access your subjects.</p>
                    <a href="/subjects" class="btn btn-primary">Go to
Subjects</a>
                </div>
            </div>
        </div>
        <div class="col-md-3">
            <div class="card">
                <img src="https://cdn-icons-
png.flaticon.com/512/2099/2099058.png" class="card-img-top" alt="Settings">
                <div class="card-body">
                    <h5 class="card-title">Settings</h5>
                    <p class="card-text">Manage your account settings - Under
construction..</p>
                    <a href="/settings" class="btn btn-primary">Go to
Settings</a>
                </div>
            </div>
        </div>
    </div>
</div>


<!-- Footer -->
<footer class="footer mt-auto py-3 bg-light">
    <div class="container text-center">
        <span class="text-muted">&copy; 2024 Your Company. All Rights
Reserved.</span>
    </div>
</footer>

<!-- Bootstrap JS and dependencies -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"><
/script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></s
cript>
</body>
</html>
```

Step 7: Run and Test

Run the Spring Boot application, and here are some screenshots:

For the [**Subject**] function: http://localhost:8080/subjects

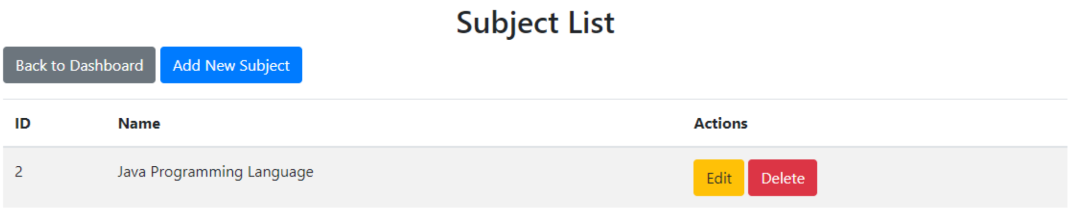
When you run it the first time (or when we haven’t added any subject yet):



Subject List

Don't have any data yet.

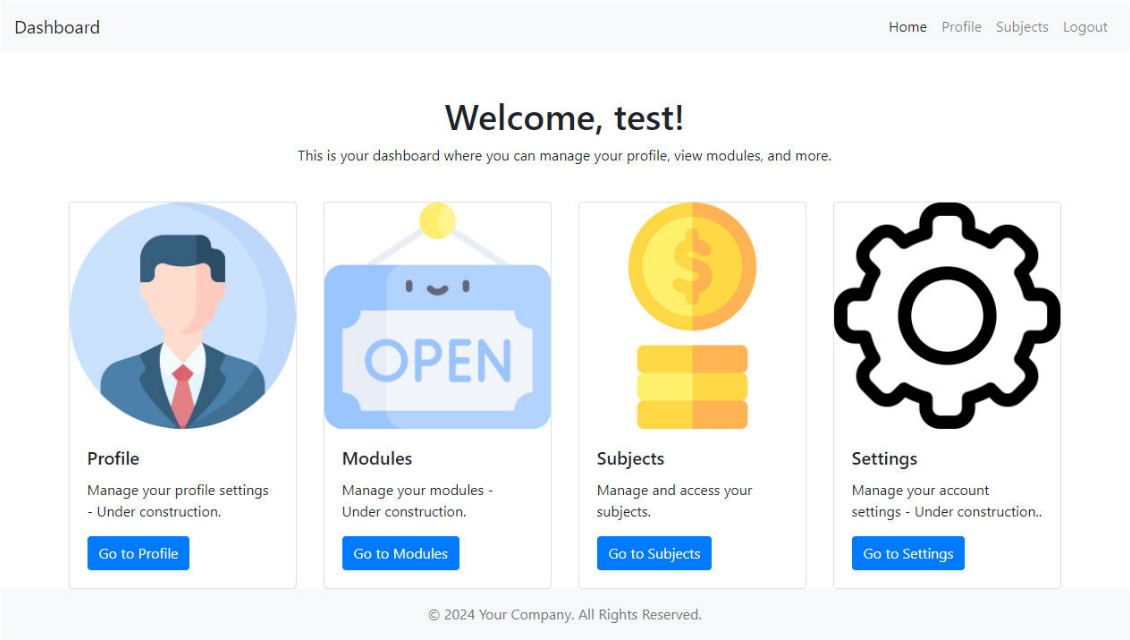
After adding a subject, here the result:



Subject List

ID	Name	Actions
2	Java Programming Language	Edit Delete

For the [**Dashboard**] function: http://localhost:8080/dashboard



Welcome, test!

This is your dashboard where you can manage your profile, view modules, and more.

Profile

Manage your profile settings - Under construction.

Modules

Manage your modules - Under construction.

Subjects

Manage and access your subjects.

Settings

Manage your account settings - Under construction..

© 2024 Your Company. All Rights Reserved.

----o0o-----

THE END