

## 주요 소스 및 설명 (함수 설명은 주석으로 첨부)

### CourseController.java

```
package kr.ac.hansung.cse.controller;

import jakarta.validation.Valid;
import kr.ac.hansung.cse.model.Course;
import kr.ac.hansung.cse.service.CourseService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.validation.ObjectError;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;

import java.util.List;

@Controller
public class CourseController { // 요청을 처리하고 결과를
    //Controller -> Service -> Dao
    //뷰로 전달
    @Autowired
    private CourseService courseService; //
    CourseController는 CourseService를 통해 데이터를 가져옴

    // 메인 페이지 (home.jsp로 이동)
    @GetMapping("/")
    public String home() {
        return "home";
    }

    // 전체 학기별 이수 학점을 조회하고 총 학점 계산
    @GetMapping("/credits")
    public String showCreditSummary(Model model) {
        List<Object[]> creditSummary =
        courseService.getCreditSummary();
        int total = courseService.getTotalCredit();

        model.addAttribute("summary", creditSummary);
        model.addAttribute("totalCredit", total);
        return "credits";
    }

    // 특정 연도 학기의 수강 과목 리스트 상세 조회
    @GetMapping("/credits/{year}/{semester}")
    public String
    showCourseDetail(@PathVariable("year") int year,
    @PathVariable("semester") int semester, Model model)
    {
```

- 사용자 요청을 처리하고, 처리 결과를 jsp에 전달하는 Spring MVC 컨트롤러.
- Controller는 Service를 통해 데이터를 전달 받고 사용자에게 jsp 뷰를 통해 보여줌
-

```

        List<Course> courses =
courseService.getCoursesBySemester(year, semester);
        model.addAttribute("courses", courses);
        return "coursedetail";
    }

    // 수강 신청 입력 폼
    @GetMapping("/register")
    public String showForm(Model model) {
        Course course = new Course();
        course.setYear(2025); // 기본값으로 설정
        course.setSemester(2); // 기본값으로 설정
        model.addAttribute("course", course);
        return "register";
    }

    // 수강 신청 폼을 제출하면 유효성 검사 수행 후 Courses
테이블에 저장
    // 유효성 실패 시 입력 폼 다시 출력
    @PostMapping("/register")
    public String doRegister(@Valid
    @ModelAttribute("course") Course course,
    BindingResult result) {
        if(result.hasErrors()) {
            System.out.println("== Form data does not
validated ==");

            List<ObjectError> errors =
result.getAllErrors();

            for(ObjectError error:errors) {
System.out.println(error.getDefaultMessage());
            }
            return "register"; // 오류 발생 시 다시
form 으로
        }

        // 오류 검증에 성공하면
        // Controller -> Service -> Dao
        courseService.insertCourse(course);
        return "registered";
    }

    // 2025년 2 학기 수강 신청한 과목 목록 조회하여 보여줌
    @GetMapping("registerView")
    public String showRegisterView(Model model) {
        List<Course> list =
courseService.getCoursesBySemester(2025, 2);
        model.addAttribute("courses", list);
        return "registerView";
    }
}

```

CourseDao.java

```
package kr.ac.hansung.cse.dao;
```

- Course 객체  
와 관련된

```

import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;
import kr.ac.hansung.cse.model.Course;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Repository
@Transactional
public class CourseDao {

    @PersistenceContext
    private EntityManager entityManager;

    // Courses 테이블의 모든 수강 과목 조회
    public List<Course> getAllCourses() {
        return entityManager.createQuery("From Course",
            Course.class).getResultList();
    }

    // 특정 연도와 학기의 과목 리스트 반환
    public List<Course> getCoursesBySemester(int year,
        int semester) {
        return entityManager.createQuery("From Course
            WHERE year=:year AND semester=:semester",
            Course.class)
            .setParameter("year", year)
            .setParameter("semester", semester)
            .getResultList();
    }

    // 수강 신청 과목 DB에 저장
    public void insertCourse(Course course) {
        entityManager.persist(course);
    }

    /// 연도/학기별 총 이수 학점을 그룹으로 집계하여 반환
    public List<Object[]> getCreditSummary() {
        return entityManager.createQuery(
            "SELECT c.year, c.semester,
            SUM(c.credit) FROM Course c GROUP BY c.year,
            c.semester ORDER BY c.year, c.semester",
            Object[].class
        ).getResultList();
    }

    // 전체 수강 과목의 이수 학점을 합산하여 반환
    public int getTotalCredit() {
        Long sum = entityManager.createQuery("SELECT
            SUM(credit) FROM Course",
            Long.class).getSingleResult();
        return sum != null ? sum.intValue() : 0;
    }
}

```

데이터베이스 접근 기능을 처리하는 DAO 클래스

- JPA의 EntityManager를 사용하여 데이터 조회, 저장, 쿼리 수행

## Course.java

```
package kr.ac.hansung.cse.model;

import jakarta.persistence.*;
import jakarta.validation.constraints.*;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@Entity
@Getter
@Setter
@NoArgsConstructor
@ToString
@Table(name="Courses")
public class Course {
    @Id
    @GeneratedValue(strategy =
GenerationType.IDENTITY)
    private int id;

    @Column(name="수강년도")
    private int year;

    @Column(name="학기")
    private int semester;

    @NotBlank(message = "교과코드를 입력하세요.")
    @Column(name="교과코드")
    private String code;

    @NotBlank(message = "교과목명을 입력하세요.")
    @Column(name="교과목명")
    private String name;

    @NotBlank(message = "교과구분을 입력하세요. (예: 전필,
전선)")
    @Column(name="교과구분")
    private String type;

    @NotBlank(message = "담당교수를 입력하세요.")
    @Column(name="담당교수")
    private String professor;

    @Min(value = 1, message = "학점은 1 이상이어야
합니다.")
    @Max(value = 3, message = "학점은 3 이하이어야
합니다.")
    @Column(name="학점")
    private int credit;
}
```

- JPA를 이용해 Courses 테이블과 1:1 매핑되는 엔티티 클래스
- 각 수강 과목의 정보를 속성으로 가지며, DB에 저장됨

## CourseService.java

- DAO와 컨트롤

```

package kr.ac.hansung.cse.service;

import kr.ac.hansung.cse.dao.CourseDao;
import kr.ac.hansung.cse.model.Course;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class CourseService {
    @Autowired
    private CourseDao courseDao;

    // 전체 수강 과목 조회
    public List<Course> getAllCourses() {
        return courseDao.getAllCourses();
    }

    // 특정 연도/학기의 수강 과목 리스트 반환
    public List<Course> getCoursesBySemester(int year,
int semester) {
        return courseDao.getCoursesBySemester(year,
semester);
    }

    // 신청한 수강 과목을 DB 에 저장
    public void insertCourse(Course course) {
        courseDao.insertCourse(course);
    }

    // 연도/학기별 총 이수 학점을 집계하여 반환
    public List<Object[]> getCreditSummary() {
        return courseDao.getCreditSummary();
    }

    // 전체 이수 학점 총합을 계산하여 반환
    public int getTotalCredit() {
        return courseDao.getTotalCredit();
    }
}

```

롤러 사이에  
 서 비즈니스  
 로직 수행하  
 는 서비스  
 계층  
 - 컨트롤러에  
 서 요청이  
 들어오면 데  
 이터를 DAO  
 를 통해 DB  
 와 통신

coursedetail.jsp

```

<%--
    Created by IntelliJ IDEA.
    User: kimsojin
    Date: 25. 4. 15.
    Time: 오후 6:58
    To change this template use File | Settings | File
    Templates.
--%>
<%@ page contentType="text/html; charset=UTF-8"

```

```

language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<html>
<head>
    <title>수강 과목 상세</title>
    <link rel="stylesheet" type="text/css"
href="

```

credits.jsp

```

<%--
    Created by IntelliJ IDEA.
    User: kimsojin
    Date: 25. 4. 15.
    Time: 오후 6:58
    To change this template use File | Settings | File
    Templates.
--%>
<%@ page contentType="text/html; charset=UTF-8"
language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"

```

```

prefix="c" %>
<html>
<head>
  <title>학점 요약</title>
  <link rel="stylesheet" type="text/css"
href="

```

home.jsp

```

<%--
  Created by IntelliJ IDEA.
  User: kimsojin
  Date: 25. 4. 15.
  Time: 오후 6:58
  To change this template use File | Settings | File
  Templates.
--%>
<%@ page contentType="text/html; charset=UTF-8"
language="java" %>

```

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
  <head>
    <title>학사 정보 시스템</title>
    <link rel="stylesheet" type="text/css"
href="

```

## register.jsp

```

<!--
  Created by IntelliJ IDEA.
  User: kimsojin
  Date: 25. 4. 15.
  Time: 오후 6:58
  To change this template use File | Settings | File
Templates.
-->
<%@ page contentType="text/html; charset=UTF-8"
language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://www.springframework.org/tags/form"
prefix="form" %>
<html>
<head>
  <title>수강 신청</title>
  <link rel="stylesheet" type="text/css"
href="

```



```

        <form:errors path="semester" cssClass="error"/>

        <label>교과코드</label>
        <form:input path="code" />
        <form:errors path="code" cssClass="error"/>

        <label>교과목명</label>
        <form:input path="name" />
        <form:errors path="name" cssClass="error"/>

        <label>교과구분 (예: 전필, 전선)</label>
        <form:input path="type" />
        <form:errors path="type" cssClass="error"/>

        <label>담당교수</label>
        <form:input path="professor" />
        <form:errors path="professor"
cssClass="error"/>

        <label>학점</label>
        <form:input path="credit" />
        <form:errors path="credit" cssClass="error"/>

        <input type="submit" value="신청" />
    </form:form>
</div>
</body>
</html>

```

registered.jsp

```

<%--
    Created by IntelliJ IDEA.
    User: kimsojin
    Date: 25. 4. 15.
    Time: 오후 6:59
    To change this template use File | Settings | File
    Templates.
--%>
<%@ page contentType="text/html; charset=UTF-8"
    language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
    prefix="c" %>

<html>
<head>
    <title>수강 신청 완료</title>
    <style>

    </style>
    <link rel="stylesheet" type="text/css"
href="

```

```
<div class="message">
    수강 신청이 성공적으로 완료되었습니다!
</div>
<a href="<c:url value='/'/>">홈으로 돌아가기</a>
</body>
</html>
```

## registerView.jsp

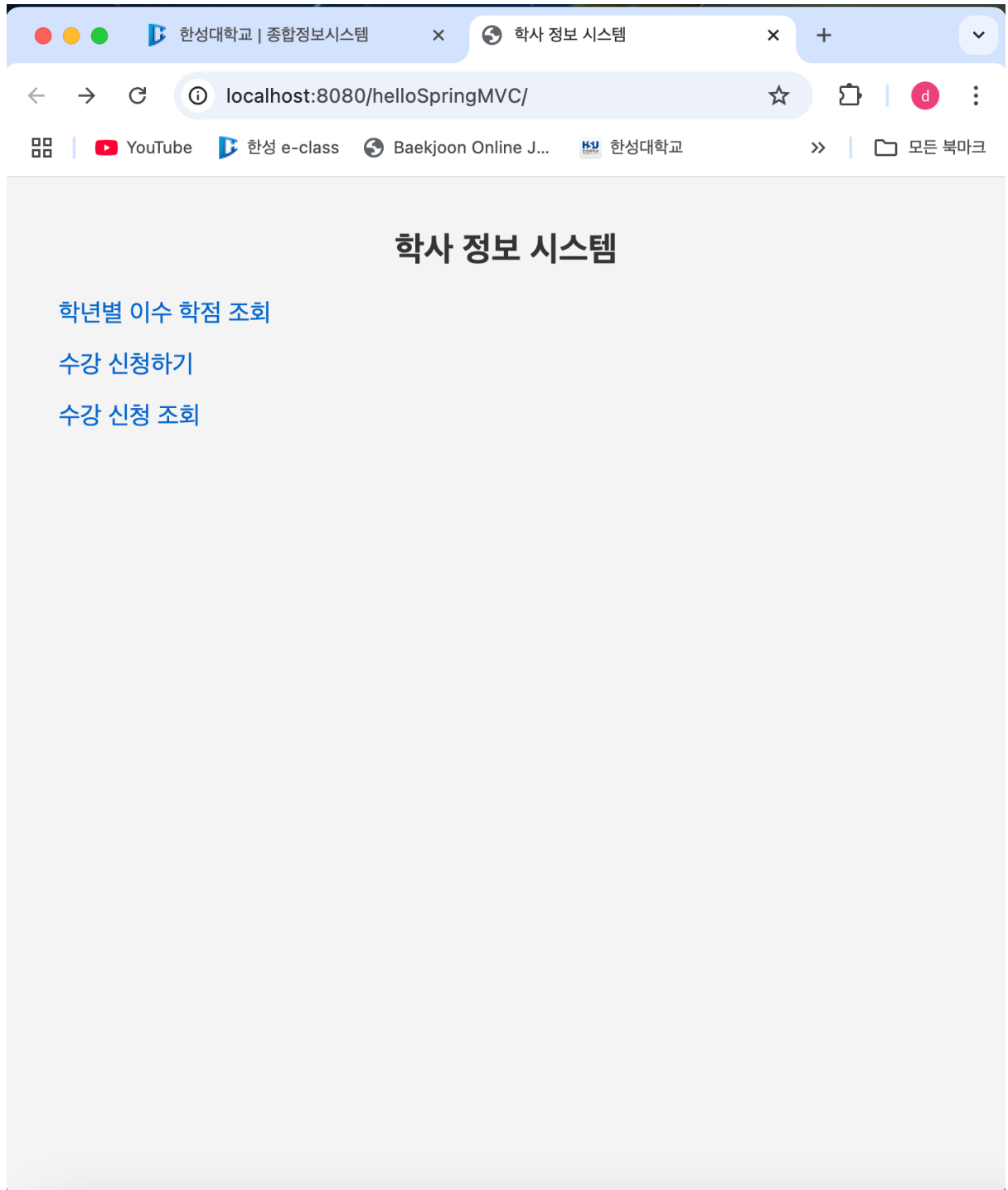
```
<%--
    Created by IntelliJ IDEA.
    User: kimsojin
    Date: 25. 4. 15.
    Time: 오후 6:59
    To change this template use File | Settings | File
    Templates.
--%>
<%@ page contentType="text/html; charset=UTF-8"
    language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
    prefix="c" %>
<html>
<head>
    <title>2025 년 2 학기 수강 내역</title>
    <link rel="stylesheet" type="text/css"
href="<c:url value='/resources/css/registerView.css'
/>">
</head>
<body>
<h2 style="text-align:center;">2025 년 2 학기 수강 신청
내역</h2>

<table>
    <thead>
        <tr>
            <th>년도</th>
            <th>학기</th>
            <th>교과목명</th>
            <th>교과구분</th>
            <th>담당교수</th>
            <th>학점</th>
        </tr>
    </thead>
    <tbody>
        <c:forEach var="course" items="${courses}">
            <tr>
                <td>${course.year}</td>
                <td>${course.semester}</td>
                <td>${course.name}</td>
                <td>${course.type}</td>
                <td>${course.professor}</td>
                <td>${course.credit}</td>
            </tr>
        </c:forEach>
    </tbody>
```

```
</table>
<div style="text-align: center; margin-top: 20px;">
  <a href="<c:url value='/'/>">홈으로 돌아가기</a>
</div>
</body>
</html>
```

## 단계별 수행 결과를 보이는 스크린샷

### 1. 홈페이지



## 2. 초기 DB (Courses)

[illegible]

### 3. 학년별 이수 학점 조회

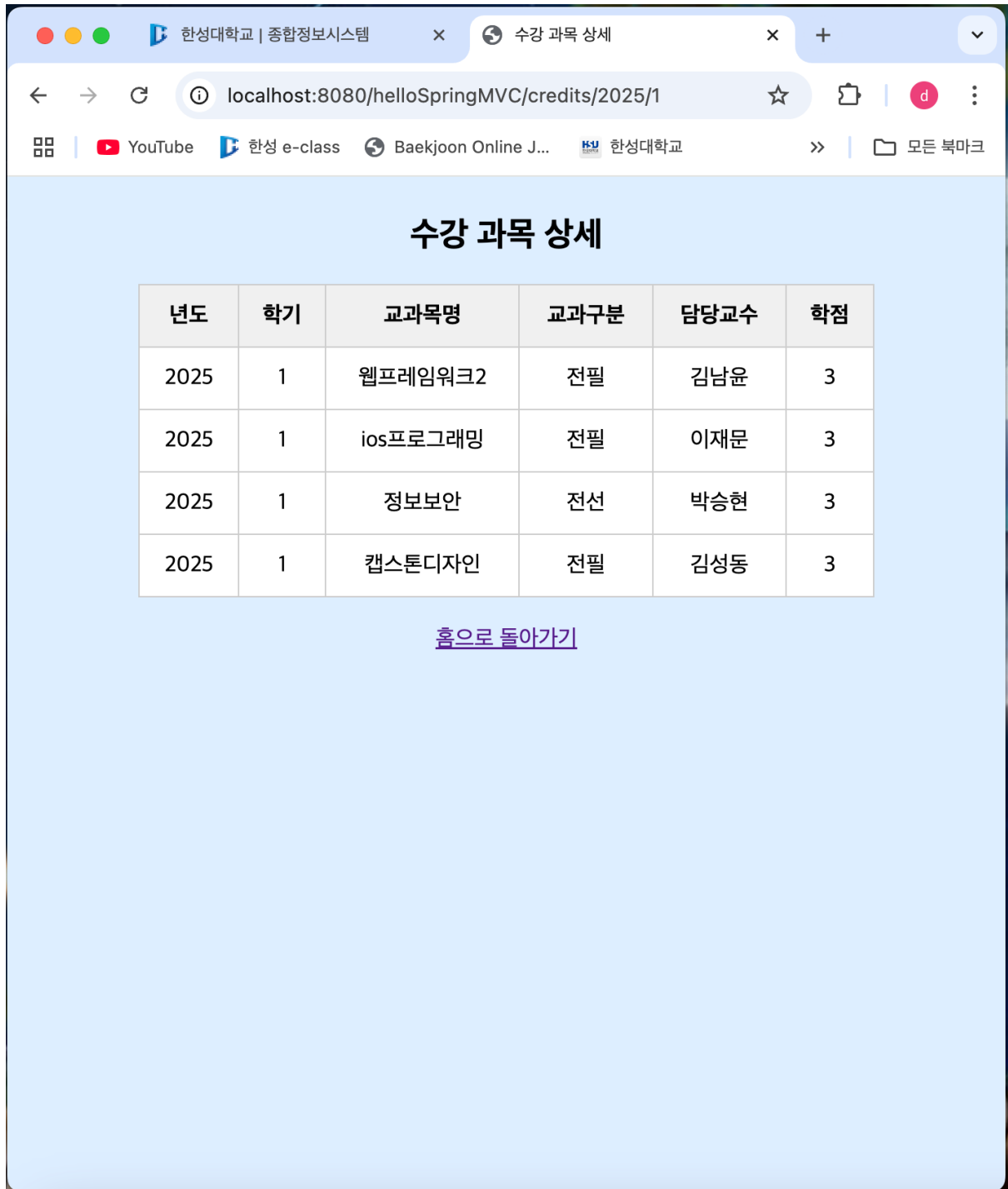
The screenshot shows a web browser window with the address bar displaying 'localhost:8080/helloSpringMVC/credits'. The browser tabs include '한성대학교 | 종합정보시스템' and '학점 요약'. The page content is as follows:

## 학기별 이수 학점

년도	학기	취득 학점	상세보기
2022	1	6	<a href="#">상세보기</a>
2022	2	3	<a href="#">상세보기</a>
2023	1	15	<a href="#">상세보기</a>
2023	2	18	<a href="#">상세보기</a>
2024	1	18	<a href="#">상세보기</a>
2024	2	15	<a href="#">상세보기</a>
2025	1	12	<a href="#">상세보기</a>
총계		87	

[홈으로 돌아가기](#)

#### 4. 상세 보기

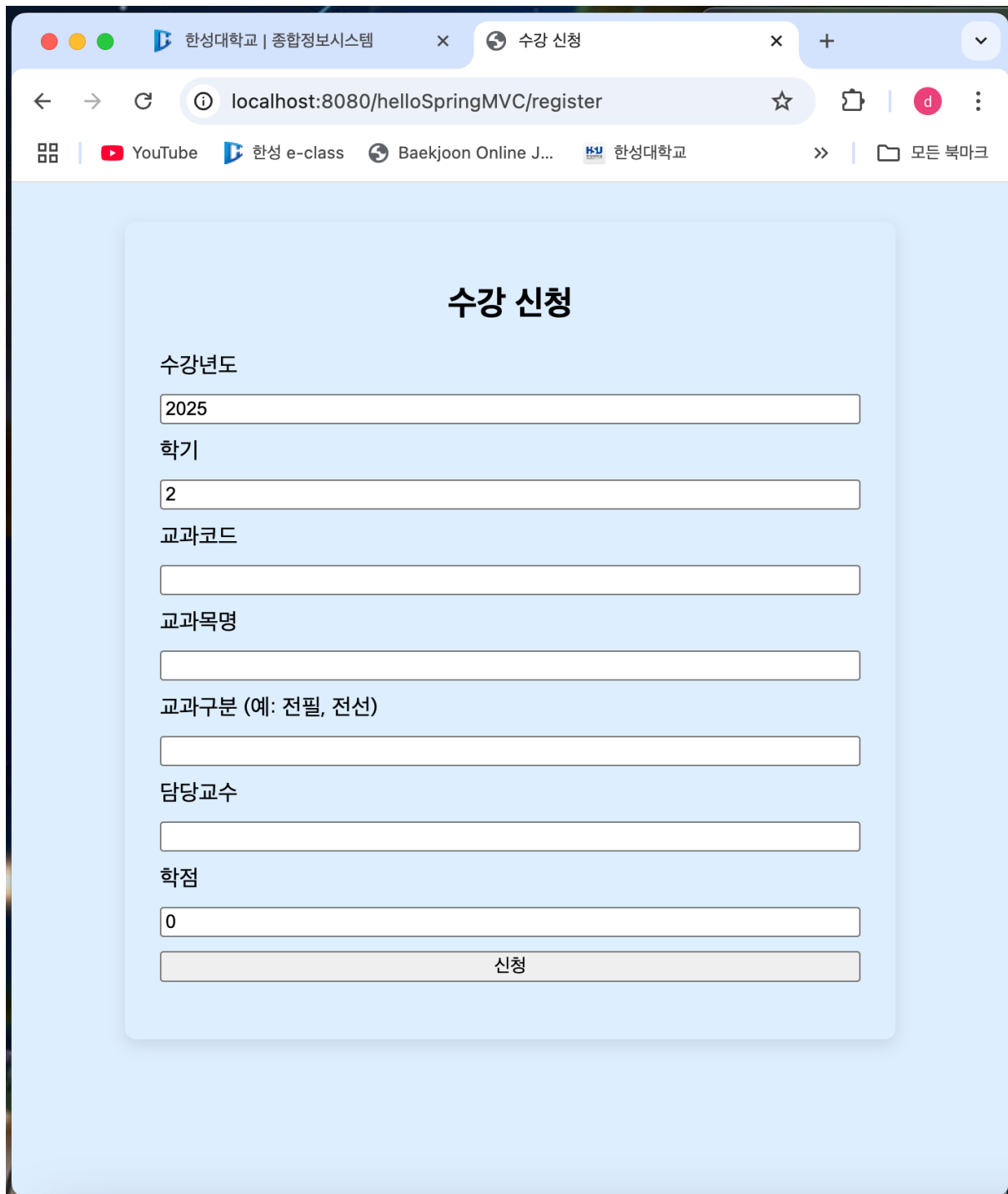


수강 과목 상세

년도	학기	교과목명	교과구분	담당교수	학점
2025	1	웹프레임워크2	전필	김남윤	3
2025	1	ios프로그래밍	전필	이재문	3
2025	1	정보보안	전선	박승현	3
2025	1	캡스톤디자인	전필	김성동	3

[홈으로 돌아가기](#)

## 5. 수강 신청



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/helloSpringMVC/register'. The browser's tab bar shows two tabs: '한성대학교 | 종합정보시스템' and '수강 신청'. The browser's bookmark bar includes links to YouTube, 한성 e-class, Baekjoon Online J..., and 한성대학교. The main content area features a light blue background with a white rounded rectangle containing the registration form. The form is titled '수강 신청' and includes the following fields:

- 수강년도: 2025
- 학기: 2
- 교과코드: (empty)
- 교과목명: (empty)
- 교과구분 (예: 전필, 전선): (empty)
- 담당교수: (empty)
- 학점: 0
- 신청: (button)



## 6. 입력 폼 오류 검증 예시 1 (모두 비어있을 경우)

한성대학교 | 종합정보시스템

수강 신청

localhost:8080/helloSpringMVC/register

☆

모든 북마크

YouTube

한성 e-class

Baekjoon Online J...

한성대학교

### 수강 신청

수강년도  
2025

학기  
2

교과코드  

교과코드를 입력하세요.

교과목명  

교과목명을 입력하세요.

교과구분 (예: 전필, 전선)  

교과구분을 입력하세요. (예: 전필, 전선)

담당교수  

담당교수를 입력하세요.

학점  
0  

학점은 1 이상이어야 합니다.

신청

## 7. 입력 폼 오류 검증 예시 1 (교과목명만 입력한 경우)

한성대학교 | 종합정보시스템

수강 신청

localhost:8080/helloSpringMVC/register

YouTube한성 e-classBaekjoon Online J...한성대학교

모든 북마크

### 수강 신청

수강년도

2025

학기

2

교과코드

교과코드를 입력하세요.

교과목명

인공지능

교과구분 (예: 전필, 전선)

교과구분을 입력하세요. (예: 전필, 전선)

담당교수

담당교수를 입력하세요.

학점

0

학점은 1 이상이어야 합니다.

신청

## 8. 신청폼 입력

한성대학교 | 종합정보시스템

수강 신청

localhost:8080/helloSpringMVC/register

YouTube한성 e-classBaekjoon Online J...한성대학교

모든 북마크

### 수강 신청

수강년도  
2025

학기  
2

교과코드  
V020030

교과코드를 입력하세요.

교과목명  
인공지능

교과구분 (예: 전필, 전선)  
전선

교과구분을 입력하세요. (예: 전필, 전선)

담당교수  
강희중

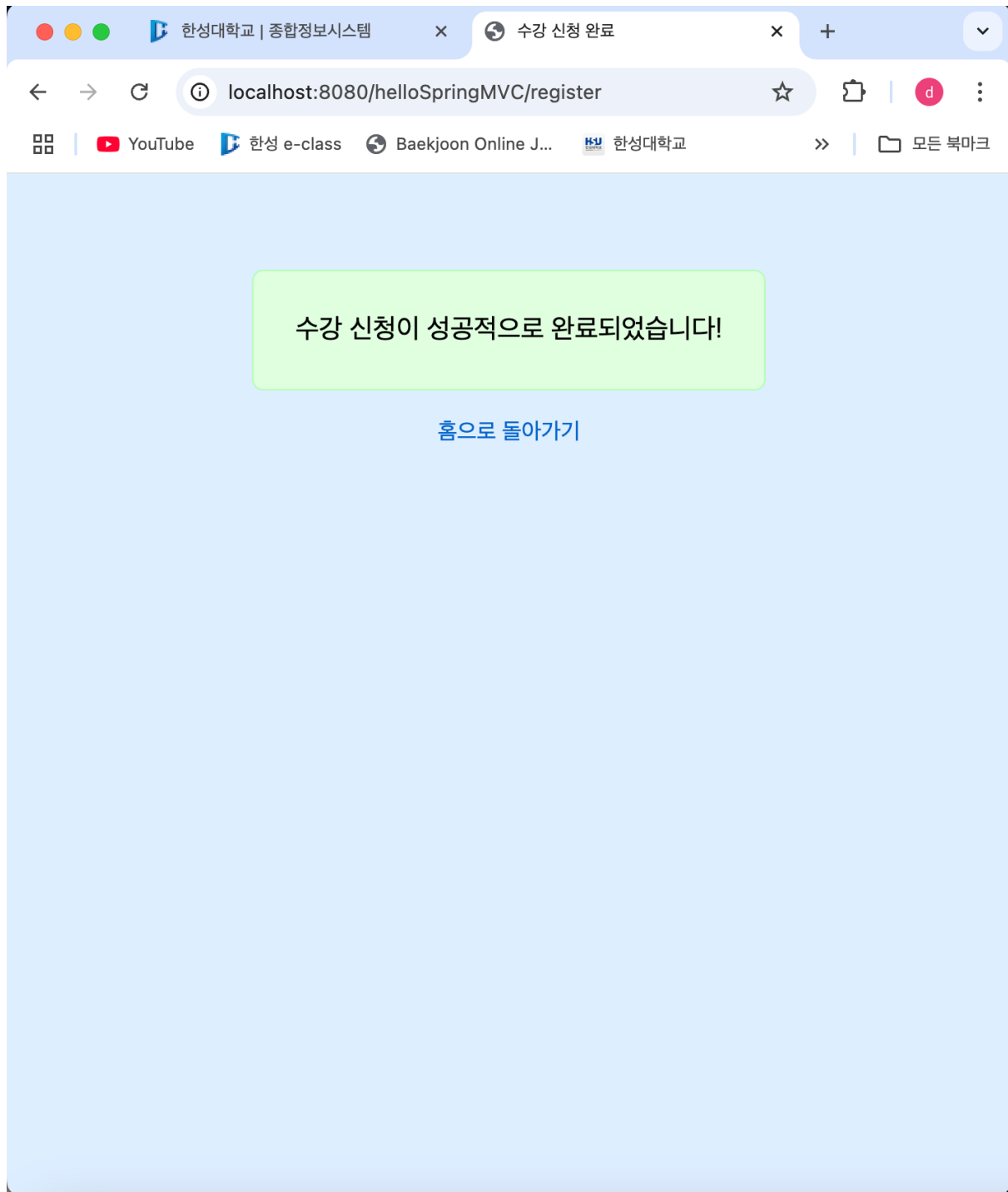
담당교수를 입력하세요.

학점  
3

학점은 1 이상이어야 합니다.

신청

## 9. 수강 신청 완료



## 10. 수강신청 완료 후 DB

[illegible]

## 11. 수강 신청 조회

한성대학교 | 종합정보시스템

2025년 2학기 수강 내역

localhost:8080/helloSpringMVC/registerView

YouTube한성 e-classBaekjoon Online J...한성대학교모든 북마크

### 2025년 2학기 수강 신청 내역

년도	학기	교과목명	교과구분	담당교수	학점
2025	2	인공지능	전선	강희중	3

[홈으로 돌아가기](#)

## 자기평가

- 학년별 이수 학점 조회, 수강 신청하기, 수강 신청 조회 모두 구현에 성공하였습니다.

## 과제 수행 느낀 점

- 이번 과제를 통해 Spring Framework 기반의 MVC 아키텍처를 실제로 구현하면서 처음 배운 Spring에 익숙해지게 되었고 컨트롤러, 서비스, DAO 계층의 각 역할과 어떻게 상호작용하는지를 배우게 되었습니다.  
JPA와 Hibernate를 사용해 데이터베이스와 연동하면서 객체지향과 관계형 데이터의 차이를 이해하게 되었고, 유효성 검증 기능을 통해 사용자 입력 오류를 더 효율적이고 쉽게 처리하는 방법도 알게 되었습니다.  
작은 오타나 경로 실수 하나에 전체 동작이 실패하는 경우가 있어서 오류를 찾기 힘들었지만 하나씩 수정하며 다음 오류를 잡을 때 도움이 되었습니다.