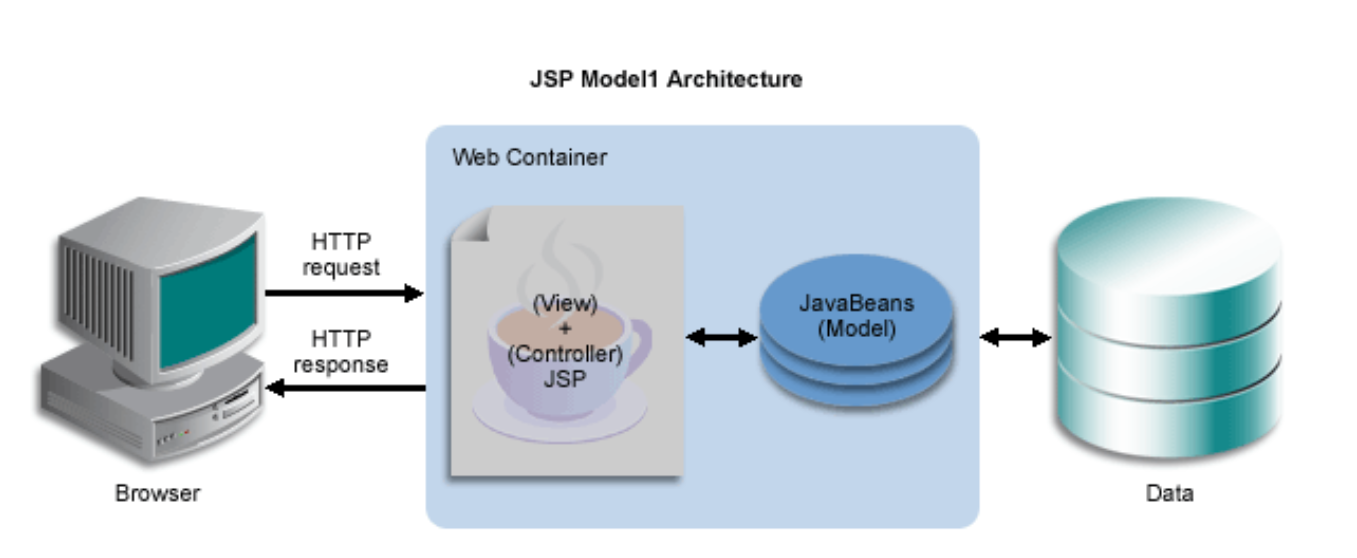
LAB14

URL: <https://lab14teamproject.herokuapp.com>

JSP Model1 Architecture



View와 Logic을 모두 JSP에서 처리하는 구조

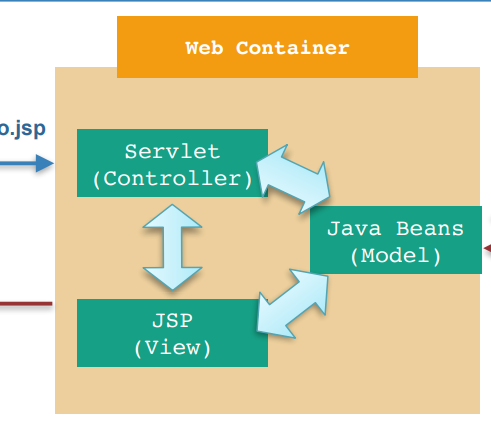
JSP는 Controller 와 View에 해당하는 역할을 하며 Java Bean은 Model에 해당한다.

Controller는 BoardDAO / BoardVO 클래스를 호출하거나 입력정보 추출하는 명령어 String id = request.getParameter(“userid”); 이나 페이지를 이동하는 response.sendRedirect(“list.jsp); 와 같은 명령어로 이루어져 있다.

View는 말 그대로 사용자에 보여지는 UI 와 관련된 부분을 작성하는 코드 파일이 된다.

Model 은 BoardDAO와 BoardVO 처럼 DB를 연동해주는 명령어가 정리 되어있는 파일이다.

JSP Model2 Architecture



Model1과 다르게 MVC에 해당하는 파일 나누어져 있으며 Controller와 View가 JSP에 의해서 작동하는 것이 아니라 모든 기능들이 따로 분리되어 있다. 즉 동시에 여러 분야의 사람들이 개발을 수행할 수 있다는 장점이 존재한다.

Mode/View/Controller의 역할

Mode: 알고리즘, DB와 상호 작용하는 CRUD 부분과 데이터 처리를 담당한다.

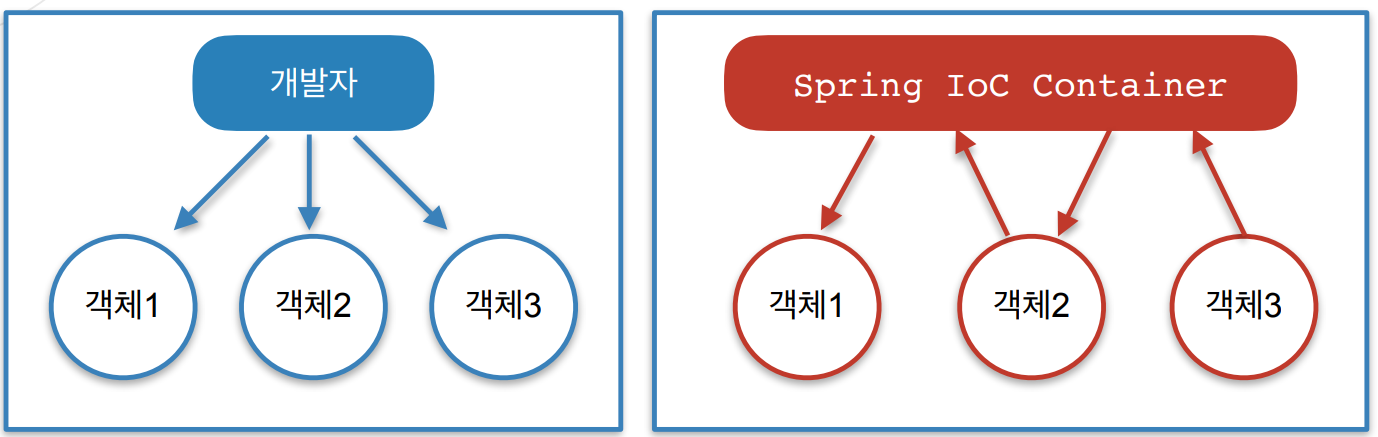
View : 사용자에게 보여주는 UI Interface 역할

Controller: 사용자 요청 처리, 화면의 로직 처리

Spring framework란?

Java를 위한 인기있는 애플리케이션 개발 플랫폼이며 쉬운 테스트와 Reusable한 코드라는 특징이 있다. Spring Framework와 Bean으로 구성되어 있다.

IoC(Inversion Of Control) / POJO



제어의 역전으로 이전에는 개발자가 관리를 하고 해당 요청을 수행했다면 IoC의 경우 객체의 생성부터 소멸 주기까지 컨테이너가 대신 관리하는 것을 말한다.

Spring project의 폴더 구조 설명

Src/main/java

: .java 파일이 모여 있는 곳. 스프링에서 MVC 패턴의 서블릿 구조를 만들어 주기 때문에 따로 서블릿을 만들 필요 없이 스프링 구조에 맞춰 클래스 파일을 작성하면 된다.

Src/main/resources

: 자바 클래스에서 사용하는 리소스를 보관하는 곳. DB 연결을 위한 자원, 의존성 주입(DI)을 위한 xml 파일 등 자바 코드 외 자원은 이곳에 보관하게 된다.

Maven Dependencies

: 메이븐에서 자동으로 관리해주는 라이브러리 폴더이다. Pom.xml 에 작성된 라이브러리를 자동으로 다운받아 관리해준다. 빌드툴을 사용함으로써 직접 관리해두지 않아도 되는 영역

Src  
: Web에 관련된 자원이 담겨있는 루트 폴더이다. 하위 폴더에 웹과 관련된 모든 자원들이 분류되어 있다. Main 폴더 안에 모두 들어가고, test 폴더는 진짜 테스트를 위한 공간이 된다.

Src/main/webapp/resources

: 웹에 필요한 다양한 자원들을 보관하는 곳이자, 사용자가 직접 접근할 수 있는 공간. Js, css, img 파일 등이 보관되어 있고 Controller가 요청을 가로채지 않고 바로 접근할 수 있도록 따로 설정해서 사용하는 곳

Src/main/webapp/WEB-INF

: 웹에 필요한 코드 파일과 컴파일된 파일, 그리고 여러 환경설정 파일들이 보관되는 곳. 보안이 중요하나 파일들이기 때문에 외부 사용자가 직접 접근할 수 없으며 Controller를 통해서 내부적으로만 접근할 수 있는 폴더

Src/main/webapp/WEB-INF/classes

: 컴파일된 파일이 보관되는 곳

Src/main/webapp/WEB-INF/spring

: 스프링 환경설정 파일(context)이 보관되는 곳

Src/main/webapp/WEB-INF/views

: JSP , HTML 파일이 보관되는 곳. 이 폴더가 루트(/)의 기준점. 사용자가 입력하고, Controller 가 받아주는 URL이 이 폴더의 구조를 따라가기 때문에 잘 정리해서 사용할 필요가 있는 곳.

URL : <https://lab14project.herokuapp.com>

Spring Tool3 Add-on 설치

(Marketplace -> Spring 검색 -> Install)

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Build Path 설정 및 Tomcat 추가

텍스트이(가) 표시된 사진

자동 생성된 설명

프로젝트 라이브러리 확인 및 실행

텍스트이(가) 표시된 사진

자동 생성된 설명

새로운 URL 처리하는 Controller 추가

텍스트이(가) 표시된 사진

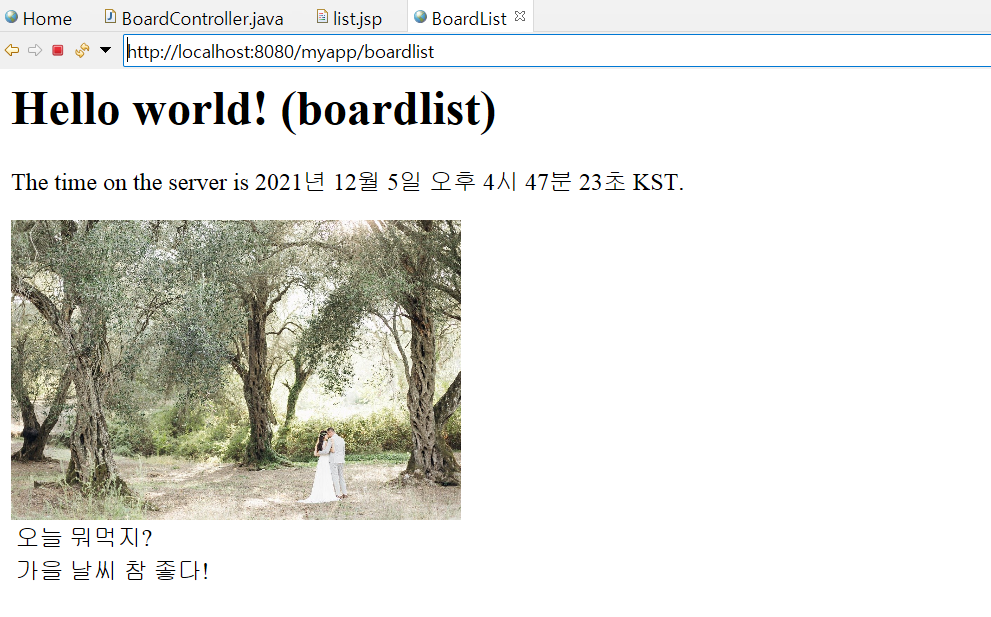
자동 생성된 설명

JSP 페이지 추가

텍스트이(가) 표시된 사진

자동 생성된 설명

Jsp 페이지에 이미지 넣기



Heroku app에 Deploy 성공

텍스트이(가) 표시된 사진

자동 생성된 설명

1. Legacy Project 생성

Spring MVC Project Template으로 생성을 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Top-level package 이름을 생성해주면서 프로젝트 생성을 완료한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 라이브러리 확인 및 Tomcat을 이용해서 정상적으로 돌아가는지 확인

기본 Setting 화면

텍스트이(가) 표시된 사진

자동 생성된 설명

정상적으로 Setting을 마치고 Run을 잘 돌아가는지 확인한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. Web.xml에 UTF-8 인코딩 필터를 추가한다.

텍스트이(가) 표시된 사진

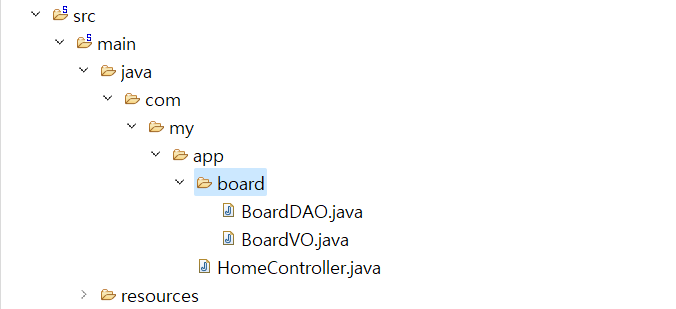
자동 생성된 설명

1. Pom.xml에 Maven Dependency를 추가한다.

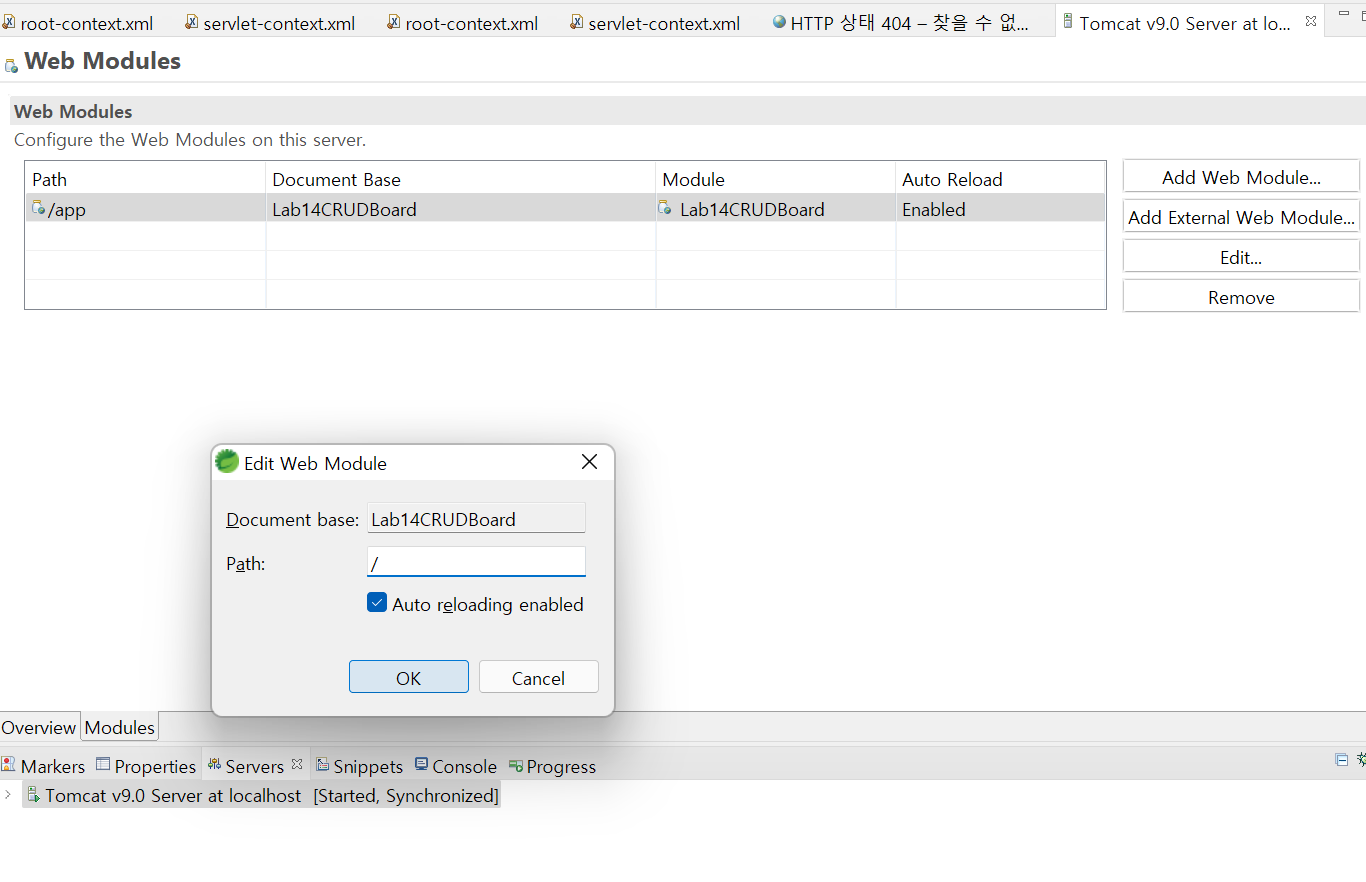
텍스트이(가) 표시된 사진

자동 생성된 설명

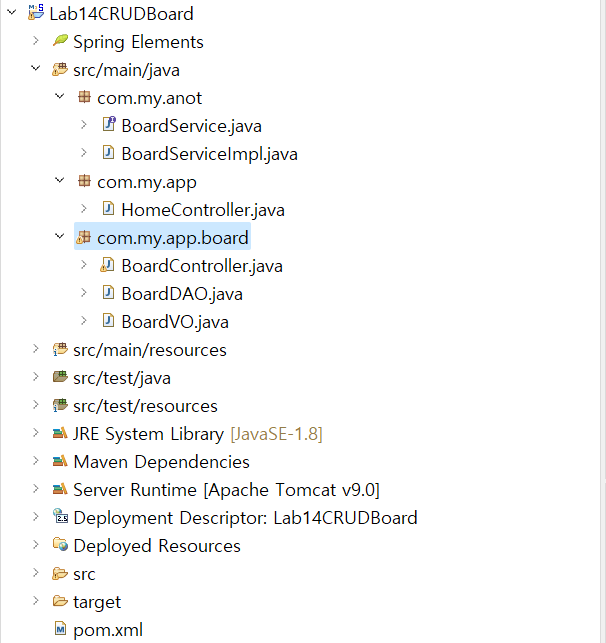
1. BoardDAO와 BoardVO 클래스 파일을 생성하고 내용 넣기



Web module path 수정



폴더 구조



BoardController

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

BoardDAO

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

BoardVO

텍스트이(가) 표시된 사진

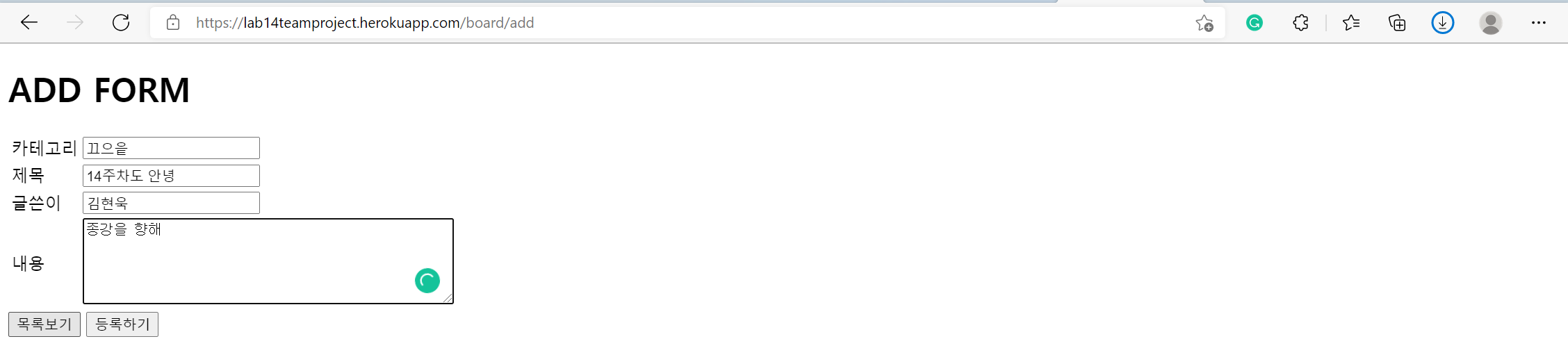
자동 생성된 설명

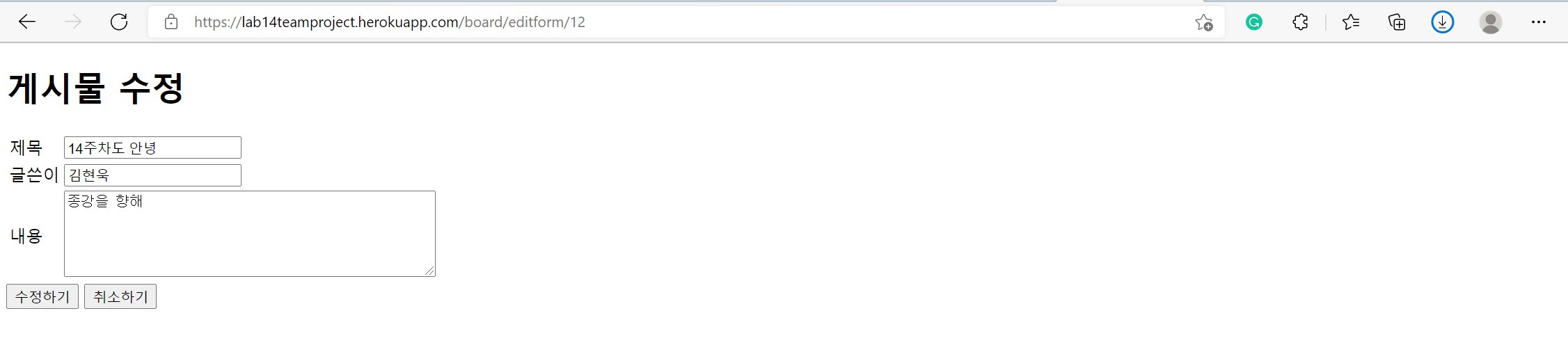
텍스트이(가) 표시된 사진

자동 생성된 설명

Deploy 화면







팀원과 협업

자신이 듣고 있는 수업에 대한 성적 관리 화면을 CRUD 기능을 추가하여 만들어보고자 합니다.

패스워드 (숫자) /수업 이름 1~10? (글자) / 수업 성적 1~10 (글자) /수업 상세 - 과제, 시험 점수등 (글자) / 수정 날짜 (날짜)

C: 자신의 페이지를 패스워드를 입력해 새롭게 생성한다.

R: 자신의 페이지로 들어가면 내가 듣는 수업과 성적, 과제 점수 등 상세 정보들을 볼 수 있다.

U: 자신의 페이지에서 수업에 대한 내용들을 수정하고 업데이트 할 수 있다.

D: 페이지 삭제 혹은 하나의 수업에 대한 내용을 모두 삭제할 수 있다.

느낀점

확실히 실습을 하기 전 이론 설명을 들을 때는 아 그냥 그렇구나 했는데 하면서 Controller와 DAO VO JSP 파일을 작성하고 오류 수정하면서 왜 안되지… 한 참을 하다가 아 이게 이렇게 MVC가 나뉘어지는 구나를 얻어갑니다. 오류를 고치고 에러가 떠도 이게 무슨 에러인지 몰라서 정말 힘들고 짜증나기도 한 순간이 더 많지만 이렇게 정상적으로 작동시키기까지 재미있었다? 이렇게 생각이 듭니다. 마지막 팀 프로젝트까지 잘 따라가겠습니다.