

14 Model 2로 만드는 게시판 프로젝트



이 장에서 다룰 내용

1

모델 2 개요

2

MVC 패턴을 사용하여 간단한 게시판 만들기

3

회원관리 시스템과 결합한 답변형 게시판 프로젝트



모델 2 개요

□ 모델 1과 모델 2의 비교

■ 모델 1



[그림 14-1] 모델1 흐름도

- 모델 1은 JSP 페이지에서 화면을 보여주는 기능도 하고 요청이 페이지로 들어올 때 요청에 관한 비즈니스 로직을 수행하는 것도 JSP 페이지에서 직접 이루어진다.
- 개발할 경우 개발 시간은 짧아질 수 있다. 하지만 유지보수 시 코드 분석이 어려워지고 디자이너가 디자인을 변경하려고 할 때 태그 안에 자바 코드들이 섞여 있기 때문에 해석하는 데 문제가 발생하게 된다.

■ 모델 2

- 디자인 부분과 개발(비즈니스 로직) 부분을 분리해서 개발
- 디자이너는 디자인 부분만, 프로그래머는 로직 부분만 개발하는 것이 가능
- 프로젝트 규모가 커지면 오히려 프로젝트 생산성도 좋아진다.
- 유지보수 시 디자인을 변경하거나 로직을 변경할 때 자신이 개발한 부분에만 접근하여 변경할 수 있다.



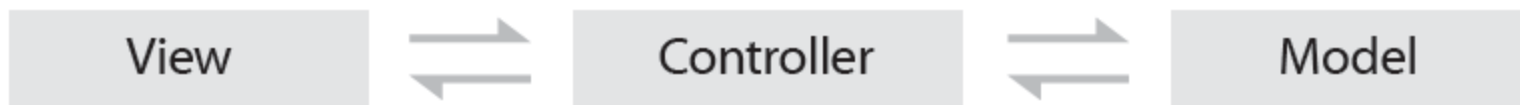
모델 2 개요

□ MVC 패턴의 개요

■ MVC란?

- Model, View, Controller의 약자
- View : 화면 출력 부분(디자인 부분)
- Model : 데이터베이스와 연동하는 등의 로직 부분
- Controller : 이 두 부분을 적절하게 연결시켜주는 역할 수행

□ MVC 패턴의 구성 및 흐름



[그림 14-2] MVC의 패턴 흐름



모델 2 개요

■ Controller (서블릿)

- MVC 패턴에서 View와 Model을 연결시켜주는 핵심 요소
- 초기 진입점이자 View와 Model을 연동하는 역할 수행
- Controller는 서블릿으로 구성

■ View(JSP)

- 화면 출력을 담당
- JSP 페이지로 구성

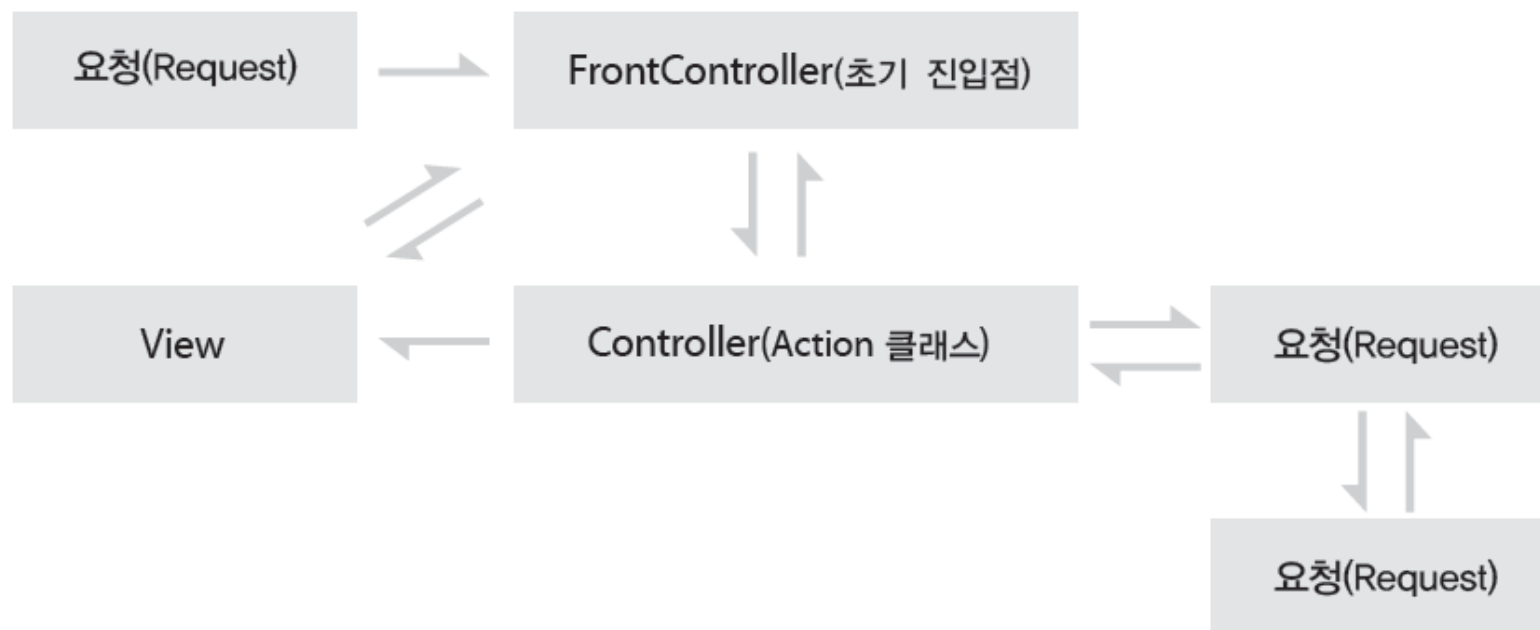
■ Model(JavaBean)

- 비즈니스 로직을 처리하는 부분
- 비즈니스 로직을 처리하고 그 결과를 Controller를 통해 View 페이지로 전달



MVC 패턴을 사용하여 간단한 게시판 만들기

□ 게시판 흐름도



[그림 14-3] 게시판 흐름도



MVC 패턴을 사용하여 간단한 게시판 만들기

□ 게시판 페이지 및 Action 클래스 설계

■ View 페이지 목록 설계하기

페이지 파일	설명
qna_board_list.jsp	게시판의 목록을 보여주는 페이지이다.
qna_board_view.jsp	게시판 글 내용을 보여주는 페이지이다.
qna_board_write.jsp	게시판 글 쓰기 폼 페이지이다.
qna_board_reply.jsp	게시판 글 답변 폼 페이지이다.
qna_board_modify.jsp	게시판 글 수정 폼 페이지이다.



MVC 패턴을 사용하여 간단한 게시판 만들기

■ Action 클래스 설계하기

클래스 파일	설명
BoardAddAction.java	글을 등록할 때 호출되는 Action 클래스이다.
BoardDeleteAction.java	글을 삭제할 때 호출되는 Action 클래스이다.
BoardDetailAction.java	글 내용을 확인할 때 호출되는 Action 클래스이다.
BoardListAction.java	글 목록을 확인할 때 호출되는 Action 클래스이다.
BoardModifyAction.java	수정 처리를 하는 Action 클래스이다.
BoardModifyView.java	수정 폼을 보여주는 Action 클래스이다.
BoardReplyAction.java	답변 처리를 하는 Action 클래스이다.
BoardReplyView.java	답변 폼을 보여주는 Action 클래스이다.



MVC 패턴을 사용하여 간단한 게시판 만들기

□ 데이터베이스 구조 설계

필드명	타입	설명
BOARD_NUM(PK)	INT	글 번호를 의미한다.
BOARD_NAME	VARCHAR2	글 작성자를 의미한다.
BOARD_PASS	VARCHAR2(15)	글의 비밀번호를 의미한다.
BOARD_SUBJECT	VARCHAR2(50)	글의 제목을 의미한다.
BOARD_CONTENT	VARCHAR2(2000)	글의 내용을 의미한다.
BOARD_FILE	VARCHAR2(50)	글에 첨부될 파일의 이름을 의미한다.
BOARD_RE_REF	INT	답변 글 작성 시 참조되는 글의 번호를 의미한다.
BOARD_RE_LEV	INT	답변 글의 깊이를 의미한다.
BOARD_RE_SEQ	INT	답변 글의 순서를 의미한다.
BOARD_READCOUNT	INT	글의 조회수를 의미한다.
BOARD_DATE	DATE	글의 작성 날짜를 의미한다.



MVC 패턴을 사용하여 간단한 게시판 만들기

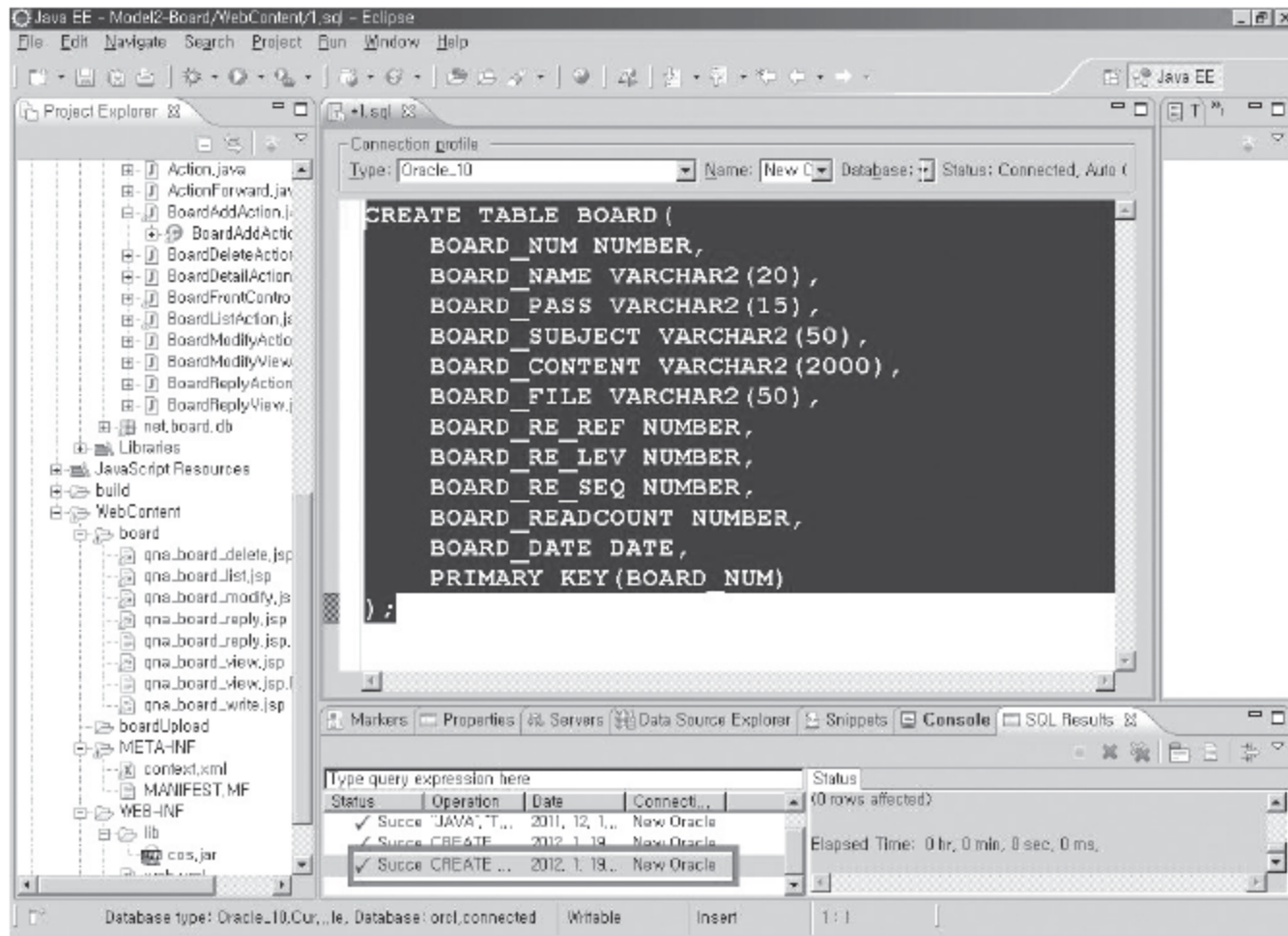
■ 테이블 생성

```
CREATE TABLE BOARD(  
  BOARD_NUM NUMBER,  
  BOARD_NAME VARCHAR2(20),  
  BOARD_PASS VARCHAR2(15),  
  BOARD_SUBJECT VARCHAR2(50),  
  BOARD_CONTENT VARCHAR2(2000),  
  BOARD_FILE VARCHAR2(50),  
  BOARD_RE_REF NUMBER,  
  BOARD_RE_LEV NUMBER,  
  BOARD_RE_SEQ NUMBER,  
  BOARD_READCOUNT NUMBER,  
  BOARD_DATE DATE,  
  PRIMARY KEY(BOARD_NUM)  
);
```



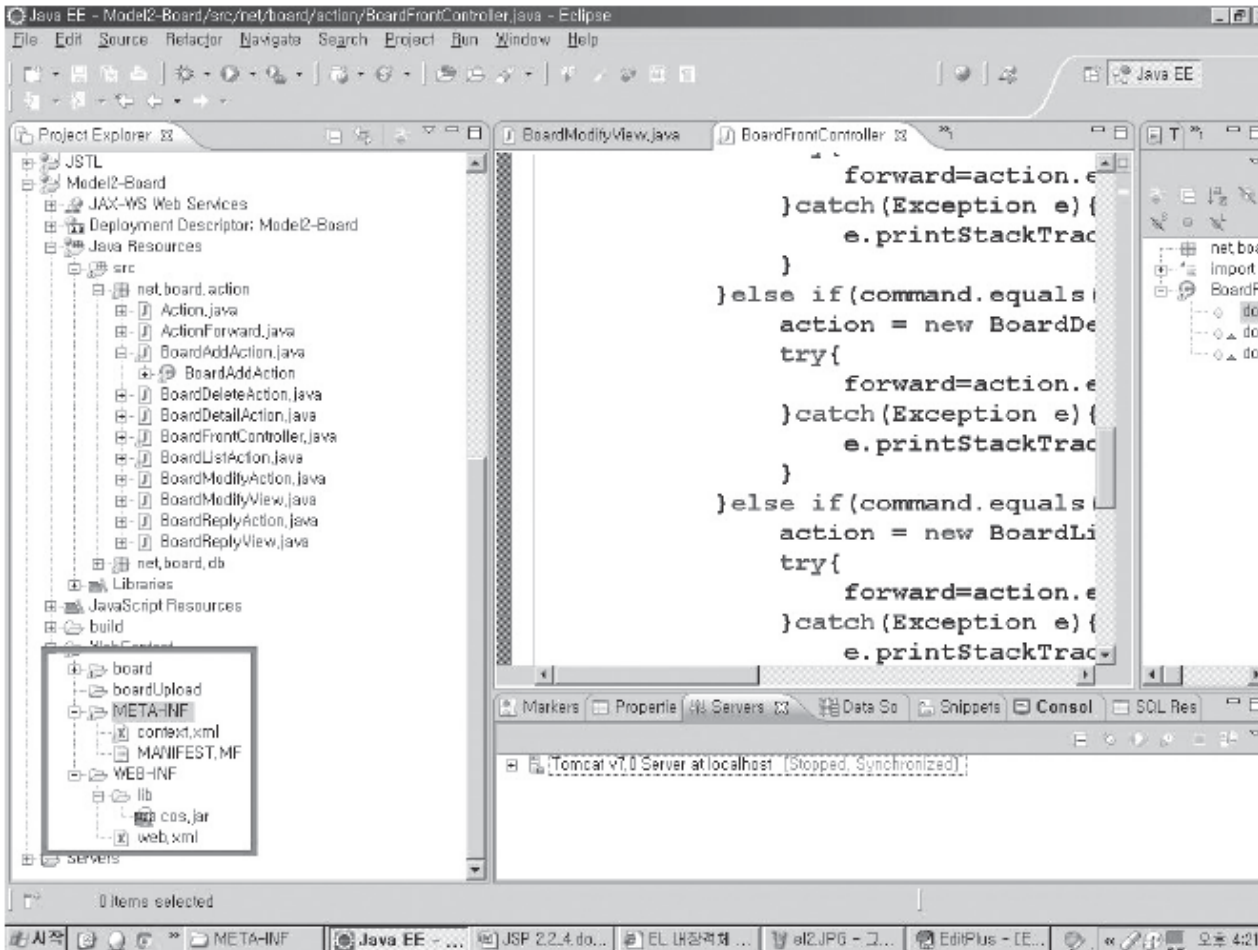
MVC 패턴을 사용하여 간단한 게시판 만들기

- 이클립스에서 실행하여 BOARD 테이블을 생성한다.



MVC 패턴을 사용하여 간단한 게시판 만들기

- DB 연결 시 커넥션 풀 기능을 이용하여 DB 작업을 수행할 것이므로 톨캣의 라이브러리 경로(톨캣홈\lib)에 ojdbc14.jar를 복사해 위치시키고 그림처럼 프로젝트에 필요한 내용들을 추가한다.



MVC 패턴을 사용하여 간단한 게시판 만들기

□ 라이브러리 설정 및 커넥션 풀 작성하기

- ojdbc14.jar 파일과 cos.jar 파일을 이클립스 프로젝트 폴더의 라이브러리 폴더에 추가하도록 한다.
- context.xml 파일을 작성한다.

```
<Context>
```

```
    <Resource name="jdbc/OracleDB"
        auth="Container"
        driverClassName="oracle.jdbc.driver.OracleDriver"
        type="javax.sql.DataSource"
        url="jdbc:oracle:thin:@localhost:1521:orcl"
        username="scott"
        password="tiger"
        maxActive="500"
        maxIdle="100"
```

```
</Context>
```



MVC 패턴을 사용하여 간단한 게시판 만들기

- JNDI 리소스를 web.xml에 등록하는 작업을 해야 한다. 최신버전은 생략해도 된다. 기존의 web.xml 내용은 그대로 두고 JNDI 리소스에 관한 코드만 추가하면 된다.

```
<?xml version="1.0" encoding="UTF-8"?>  
<web-app id="WebApp_ID" version="2.4"  
  xmlns="http://java.sun.com/xml/ns/j2ee"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee  
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
```

```
  <display-name>Model2-Board</display-name>
```

```
  <resource-ref>
```

```
    <description>Connection</description>
```

```
    <res-ref-name>jdbc/OracleDB</res-ref-name>
```

```
    <res-type>javax.sql.DataSource</res-type>
```

```
    <res-auth>Container</res-auth>
```

```
  </resource-ref>
```

```
</web-app>
```



MVC 패턴을 사용하여 간단한 게시판 만들기

□ 데이터 빈(DataBean) 클래스 작성(BoardBean.java)

- 패키지는 'net.board.db' 를 사용하도록 한다.
- TO(Transfer Object)나 VO(Value Object) 형태의 클래스를 작성할 때는 이클립스에서 제공해주는 기능을 이용할 것을 권한다.
- 해당 소스에서 get/set 메소드를 생성할 속성을 선택한 후 Alt + Shift + V 단축키를 누르고 메소드를 자동으로 생성할 속성 체크 박스를 선택한 후 [OK] 버튼을 클릭하면 해당 속성별로 get/set 메소드가 자동으로 생성된다.

□ DAO(Data Access Object) 클래스 작성(BoardDAO.java)

- 실제로 데이터베이스와 연동하여 레코드의 추가, 수정, 삭제 작업이 이루어지는 클래스
- 패키지는 BoardBean.java 파일과 같은 net.board.db 로 설정하도록 한다.



MVC 패턴을 사용하여 간단한 게시판 만들기

■ 답글 관련 설명(198p ~ 199p)

- 현재 작성되어 있는 글이 다음과 같다고 가정하자.

글번호	제목	BOARD_RE_REF	BOARD_RE_LEVEL	BOARD_RE_SEQ
1	Jsp 공부는 정말 재밌다.	1	0	0
2	나도 찬성	1	1	1
3	Java 공부도 재밌다.	1	1	2
4	나도 찬성	1	2	3

- 이 상황에서 1번 글에 대해서 두 번째 답글을 입력하면 두 번째 답글은 1번 글 바로 밑에 출력되어야 할 것이다.

글번호	제목	BOARD_RE_REF	BOARD_RE_LEVEL	BOARD_RE_SEQ
1	Jsp 공부는 정말 재밌다.	1	0	0
5	나도 찬성	1	1	1
2	나도 찬성	1	1	1+1
3	Java 공부도 재밌다.	1	1	2+1
4	나도 찬성	1	2	3+1

MVC 패턴을 사용하여 간단한 게시판 만들기



글번호	제목	BOARD_RE_REF	BOARD_RE_LEVEL	BOARD_RE_SEQ
1	Jsp 공부는 정말 재밌다.	1	0	0
5	나도 찬성	1	1	1
2	나도 찬성	1	1	2
3	Java 공부도 재밌다.	1	1	3
4	나도 찬성	1	2	4

- 5번 글부터 BOARD_RE_SEQ 값을 1씩 증가시켜 주어야 현재 입력한 답글 값을 답글 대상 글의 값보다 1 증가시킬 수 있다.



MVC 패턴을 사용하여 간단한 게시판 만들기

□ FrontController(프론트 컨트롤러) 작성

■ FrontController

- 해당 웹 애플리케이션으로 요청이 전송되었을 때 가장 처음으로 호출되는 컨트롤러 역할을 한다.
 - 웹 애플리케이션에 요청이 전송되어 오면 웹 컨테이너는 해당 요청을 처리하기 위해서 web.xml 파일의
 - web.xml 파일에는 해당 웹 애플리케이션에서 제공하는 서블릿에 관한 설정 내용이 기록되어 있다. 이 서블릿 설정 코드에서 특정한 URL 패턴으로 전송되어 오는 요청들을 특정한 서블릿에서 처리되게 설정하는 역할을 한다. 설정 내용을 참조한다.
- `<url-pattern>*.bo</url-pattern>` 요청은 `<servlet-class>net.board.action.BoardFrontController</servlet-class>`에서 설정한 대로 BoardFrontController에서 처리하도록 설정할 것이다.



MVC 패턴을 사용하여 간단한 게시판 만들기

■ 예제

- BoardFrontController.java (교재 394p 참조)
- web.xml

```
...  
<servlet>  
    <servlet-name>BoardFrontController</servlet-name>  
    <servlet-class>net.board.action.BoardFrontController</servlet-class>  
</servlet>  
<servlet-mapping>  
    <servlet-name>BoardFrontController</servlet-name>  
    <url-pattern>*.bo</url-pattern>  
</servlet-mapping>  
...
```



MVC 패턴을 사용하여 간단한 게시판 만들기

□ Action 인터페이스, ActionForward 클래스의 작성

■ Action 인터페이스

- 특정 비즈니스 요청을 수행하고 결과 값을 ActionForward 타입으로 반환하는 메소드가 정의되어 있다.
- 예제 (Action.java)
 - Action.java 파일은 net.board.action 패키지에 위치시키도록 한다.

```
1 package net.board.action;  
2  
3 import javax.servlet.http.*;  
4  
5 public interface Action {  
6     public ActionForward execute(HttpServletRequest request, HttpServletResponse response)  
7     throws Exception;  
8 }
```



MVC 패턴을 사용하여 간단한 게시판 만들기

■ ActionForward 클래스

- Action 인터페이스에서 명령을 수행하고 결과 값을 가지고 페이지를 포워딩할 때 사용되는 클래스이다
- 예제 (ActionForward.java)

```
1 package net.board.action;
2
3 public class ActionForward {
4     private boolean isRedirect=false;
5     private String path=null;
6
7     public boolean isRedirect(){
8         return isRedirect;
9     }
10
11     public String getPath(){
12         return path;
13     }
14
15     public void setRedirect(boolean b){
16         isRedirect=b;
17     }
18
19     public void setPath(String string){
20         path=string;
21     }
22 }
```



MVC 패턴을 사용하여 간단한 게시판 만들기

□ 게시판 Action 클래스의 작성

■ Action 클래스

- View와 Model 사이의 통로 역할을 하는 컨트롤러라고 할 수 있다.
- View와 Model 사이에 존재하면서 원활하게 요청을 처리해주는 역할 수행
- Action 클래스는 앞에서 작성한 Action 인터페이스나 ActionForward 클래스처럼 net.board.action 패키지에 위치시킨다.

■ 예제(교재 400p~ 412p 참조)

- 글 등록에 대한 Action 클래스의 소스 작성하기(BoardAddAction.java)
- 글 목록 보기와 관련된 Action 클래스 작성하기(BoardListAction.java)
- 글 내용 보기와 관련된 Action 클래스 작성하기(BoardDetailAction.java)
- 글 삭제에 관련된 Action 클래스 소스 작성하기(BoardDeleteAction.java)
- 글 수정 화면을 출력해주는 Action 클래스 작성하기(BoardModifyView.java)
- 수정 처리를 할 때 호출되는 Action 클래스 작성하기(BoardModifyAction.java)
- 답변하기 폼을 출력하는 Action 클래스 작성하기(BoardReplyView.java)
- 답변글을 테이블에 저장해주는 Action 클래스 작성하기(BoardReplyAction.java)



MVC 패턴을 사용하여 간단한 게시판 만들기

□ View 페이지의 작성

■ View 페이지

- 액션 클래스에서 비즈니스 로직이 수행되고 DB에서 가져온 데이터를 컨트롤러에서 일정 영역에 저장하고, 영역에 저장되어 있는 View 페이지로 포워딩 되었을 때 영역에 저장되어 있는 데이터에 액세스해서 해당 데이터를 요청에 맞게 조작해서 출력해주는 역할을 한다.

■ 예제(교재 412p~ 425p 참조)

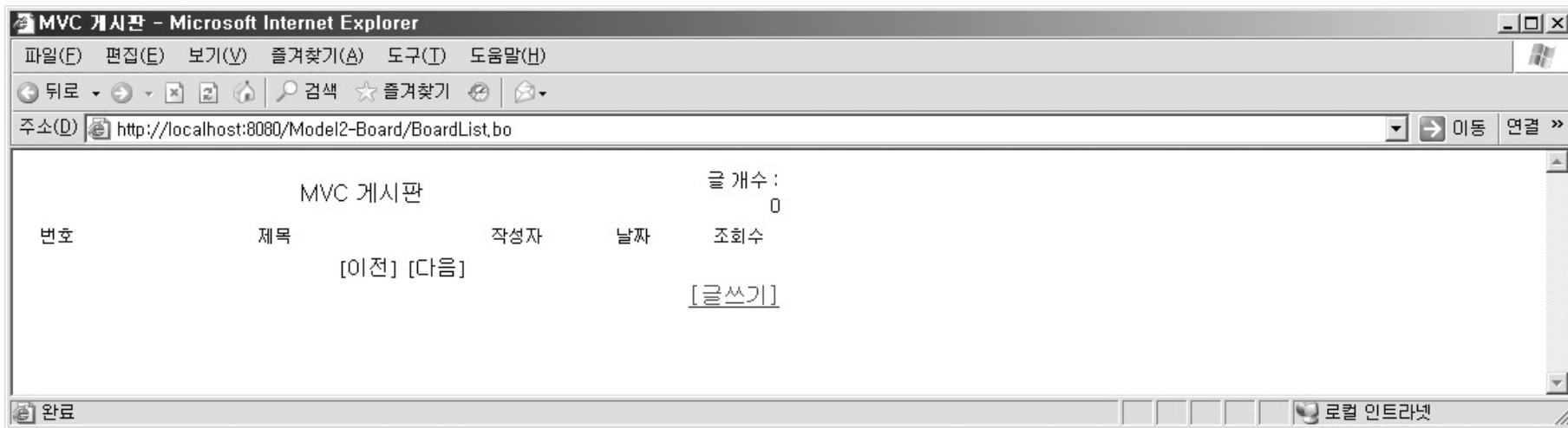
- 글 목록을 보여주는 뷰 페이지 소스 작성하기(qna_board_list.jsp)
- 글 내용을 보여주는 View 페이지 소스 작성하기(qna_board_view.jsp)
- 글 쓰기 폼 페이지 소스 작성하기(qna_board_write.jsp)
- 글 수정 폼 페이지 소스 작성하기(qna_board_modify.jsp)
- 글 답변 폼 페이지 소스 작성하기(qna_board_reply.jsp)
- 글 삭제 폼 페이지 작성하기(qna_board_delete.jsp)



MVC 패턴을 사용하여 간단한 게시판 만들기

□ 게시판의 동작 확인하기

- 01. 다음과 같이 BoardList.bo를 호출한다. 처음에는 데이터베이스에 레코드가 하나도 존재하지 않으므로, 글이 존재하지 않는다. [글쓰기]를 클릭한다.



MVC 패턴을 사용하여 간단한 게시판 만들기

- 02. 다음과 같이 글을 작성하고 [등록]을 클릭한다.

MVC 게시판

이름

비밀번호

제목

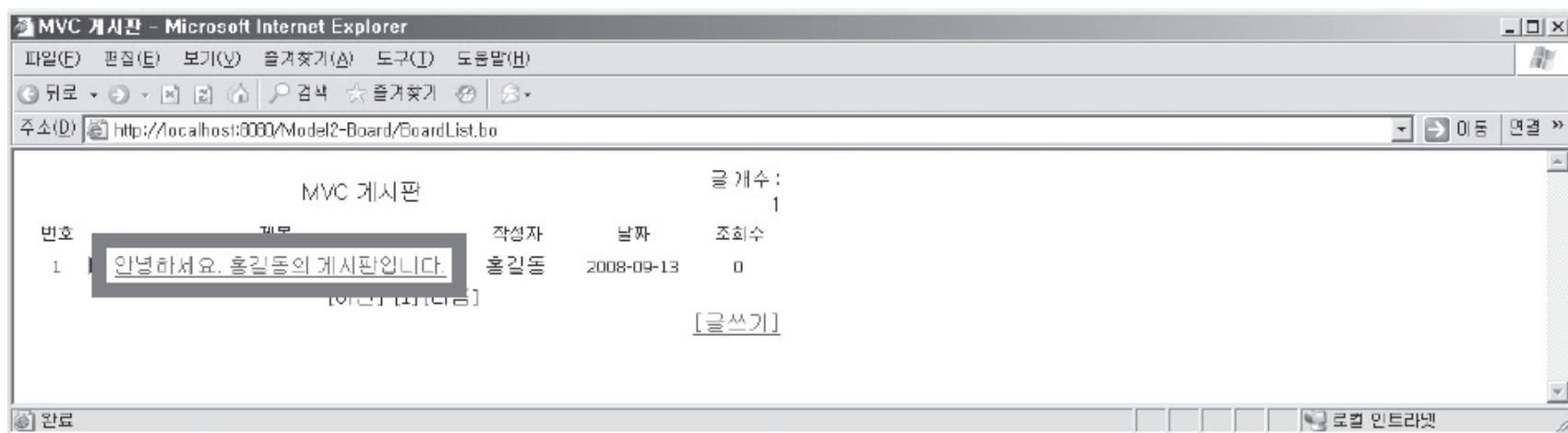
내용

안녕하세요, 잘 부탁드립니다.

파일 첨부

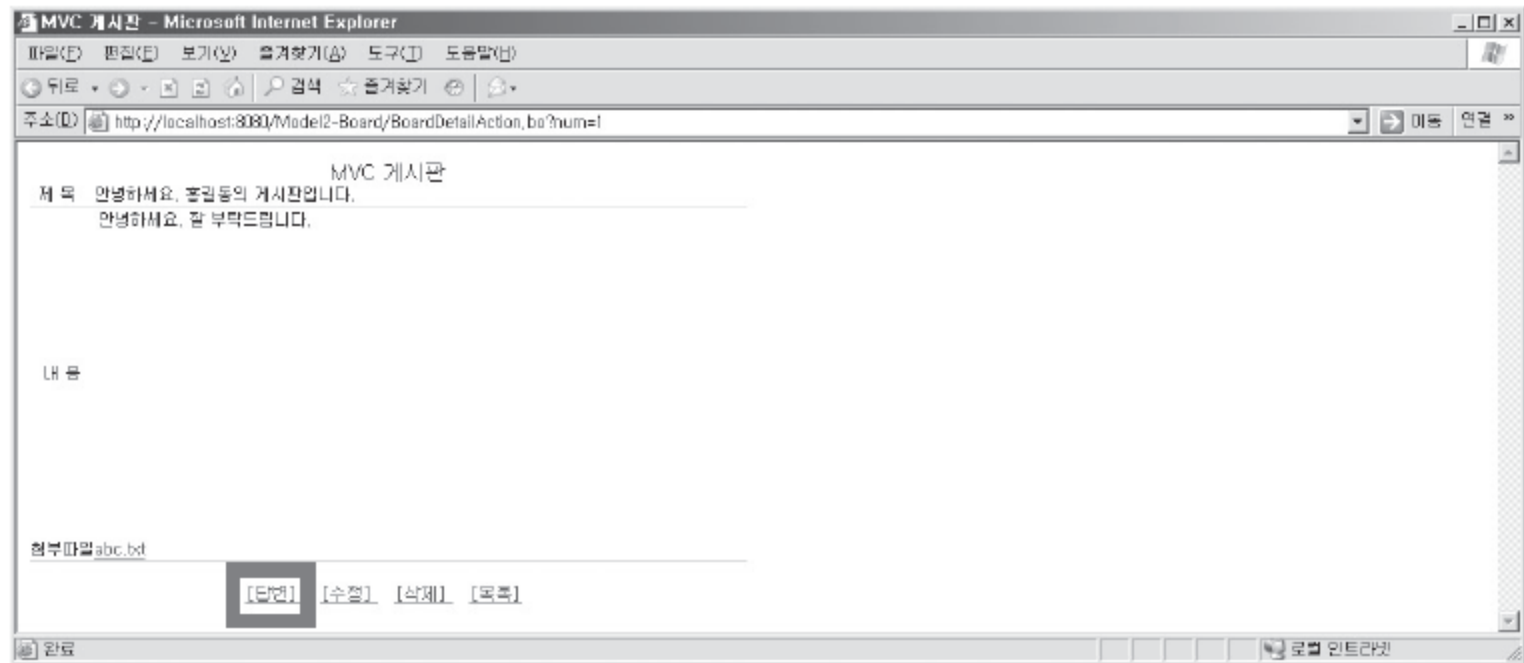
MVC 패턴을 사용하여 간단한 게시판 만들기

- 03. 작성한 글이 데이터베이스에 등록되어 보여지면 글 제목 부분을 클릭한다.



MVC 패턴을 사용하여 간단한 게시판 만들기

- 04. 다음과 같은 화면이 출력된다. 이어서 [답변]을 클릭한다.



MVC 패턴을 사용하여 간단한 게시판 만들기

- 05. 다음과 같은 화면이 출력된다. [등록]을 클릭한다

The screenshot shows a Microsoft Internet Explorer window titled "MVC 게시판 - Microsoft Internet Explorer". The address bar displays the URL "http://localhost:8080/Model2-Board/BoardReplyAction.bo?num=1". The main content area is titled "MVC 게시판" and contains a form for replying to a board post. The form fields are as follows:

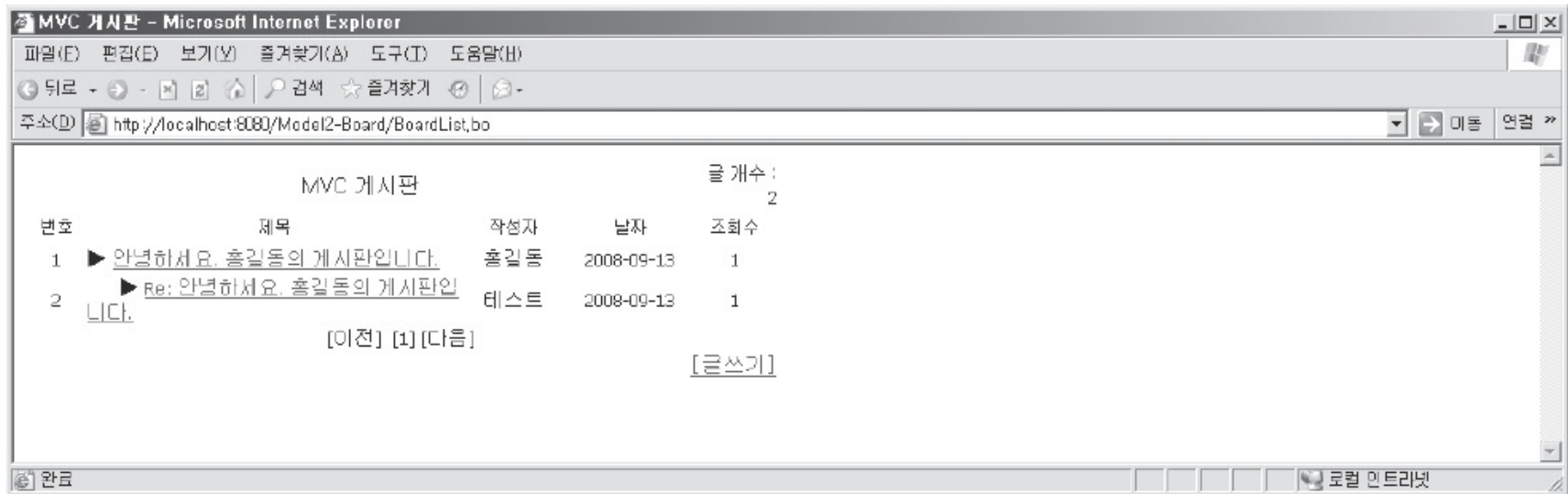
- 글쓴이** (Author): A text input field containing the text "테스트".
- 제 목** (Subject): A text input field containing the text "Re: 안녕하세요. 홍길동의 게시판입니다."
- 내 용** (Content): A large text area containing the text "답변 테스트입니다."
- 비밀번호** (Password): A password input field with four dots (••••).

At the bottom of the form, there are two buttons: "[등록]" (Register) and "[뒤로]" (Back). The "[등록]" button is highlighted with a black border. The status bar at the bottom of the browser window shows "완료" (Done) and "로컬 인트라넷" (Local intranet).



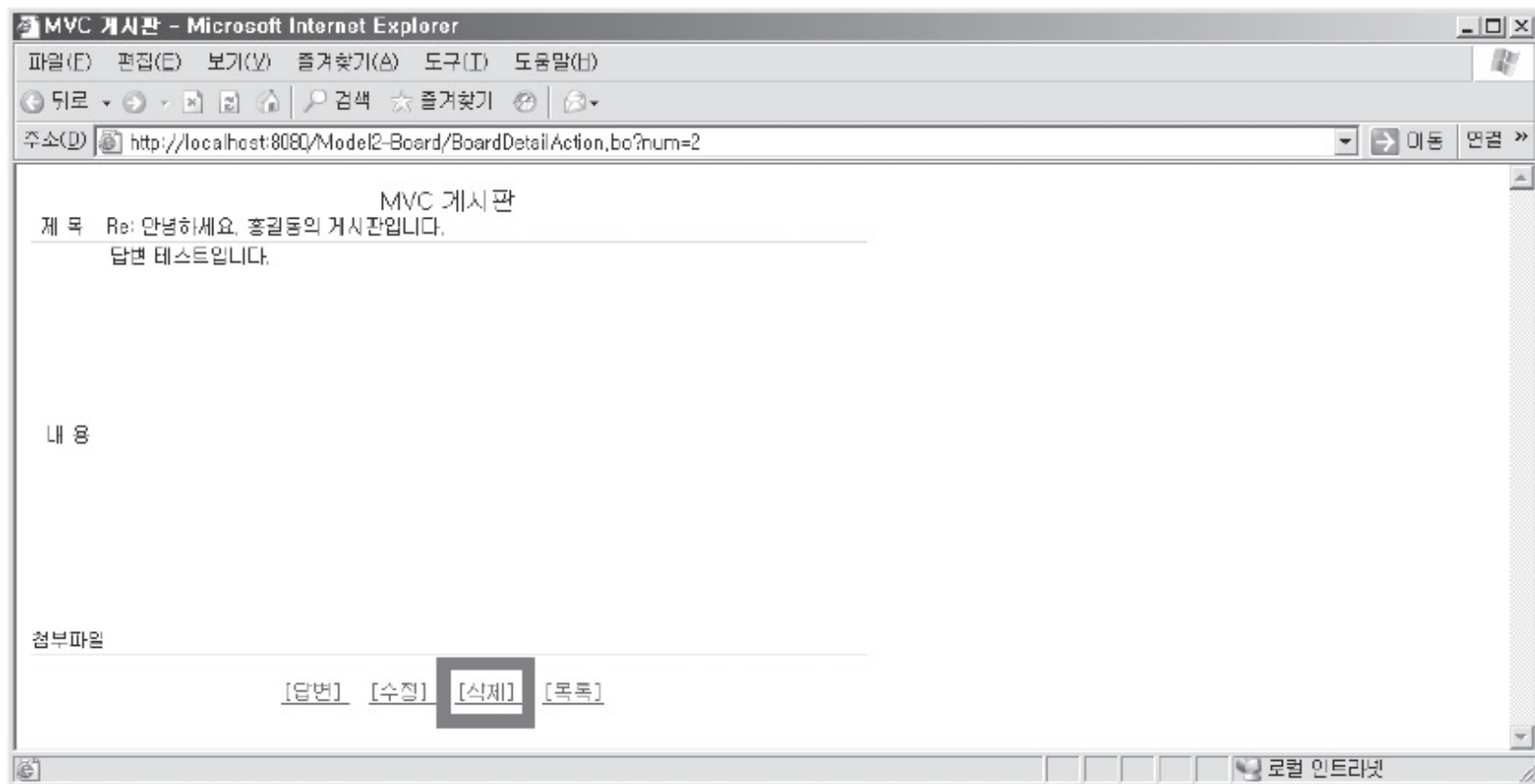
MVC 패턴을 사용하여 간단한 게시판 만들기

- 06. 다음과 같은 화면이 출력된다. 답변 글을 쓰게 되면 화면처럼 답변 글이 시각적으로도 확인할 수 있게 구현되어 있다.



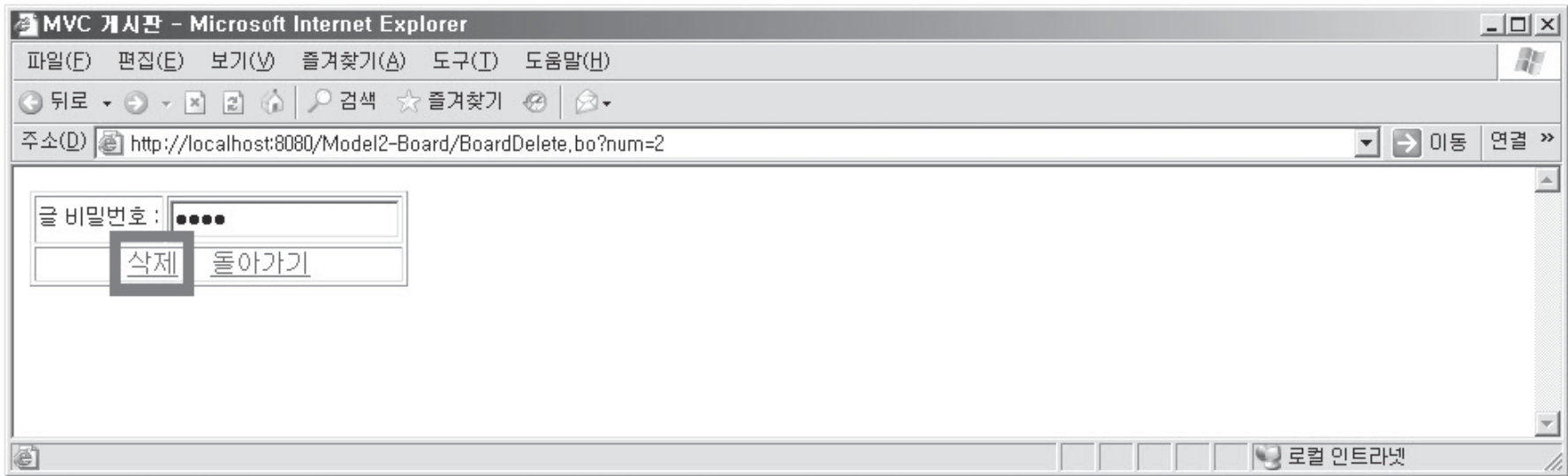
MVC 패턴을 사용하여 간단한 게시판 만들기

- 07. 이번에는 답변 글을 클릭해 들어간 후 [삭제]를 클릭해 답변 글을 삭제해 보도록 한다.



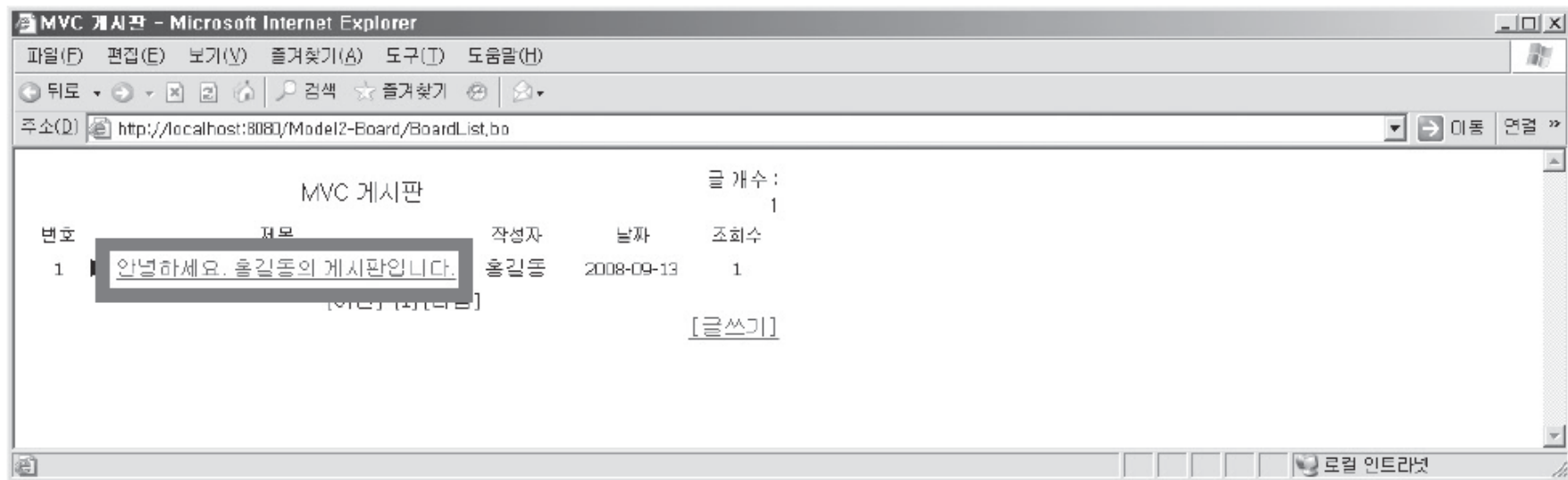
MVC 패턴을 사용하여 간단한 게시판 만들기

- 08. 다음과 같이 본인임을 확인하기 위해 비밀번호를 입력할 수 있는 화면이 출력된다. 비밀번호를 올바르게 입력하고 [삭제]를 클릭한다.



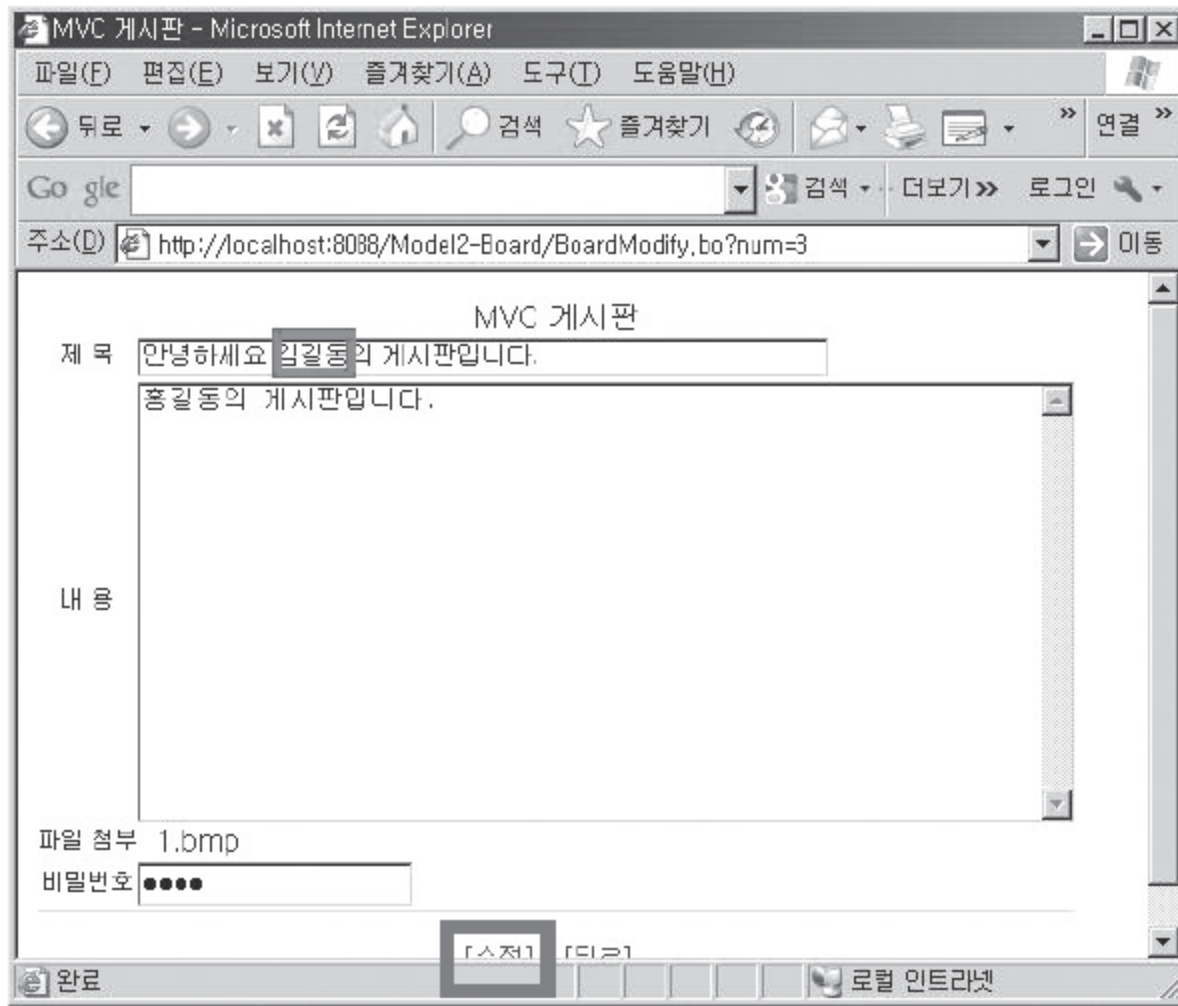
MVC 패턴을 사용하여 간단한 게시판 만들기

- 09. 다음과 같이 답변 글이 삭제된 화면이 출력된다. 이번에는 작성한 1번 게시글을 클릭하여 들어간다.



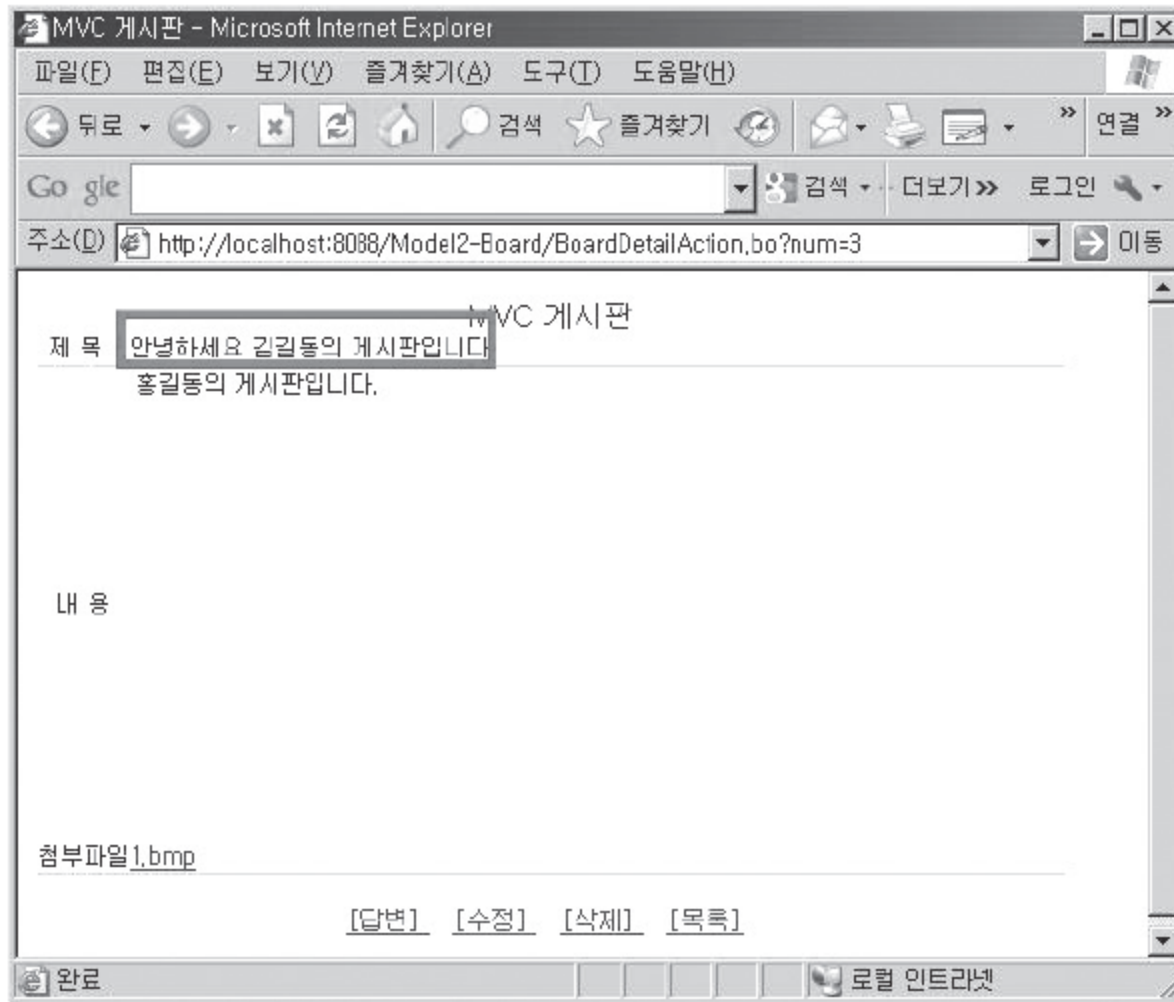
MVC 패턴을 사용하여 간단한 게시판 만들기

- 10. 내용 보기 화면에서 다음 화면처럼 이름을 수정하고 비밀번호를 입력한 후 [수정]을 클릭한다.



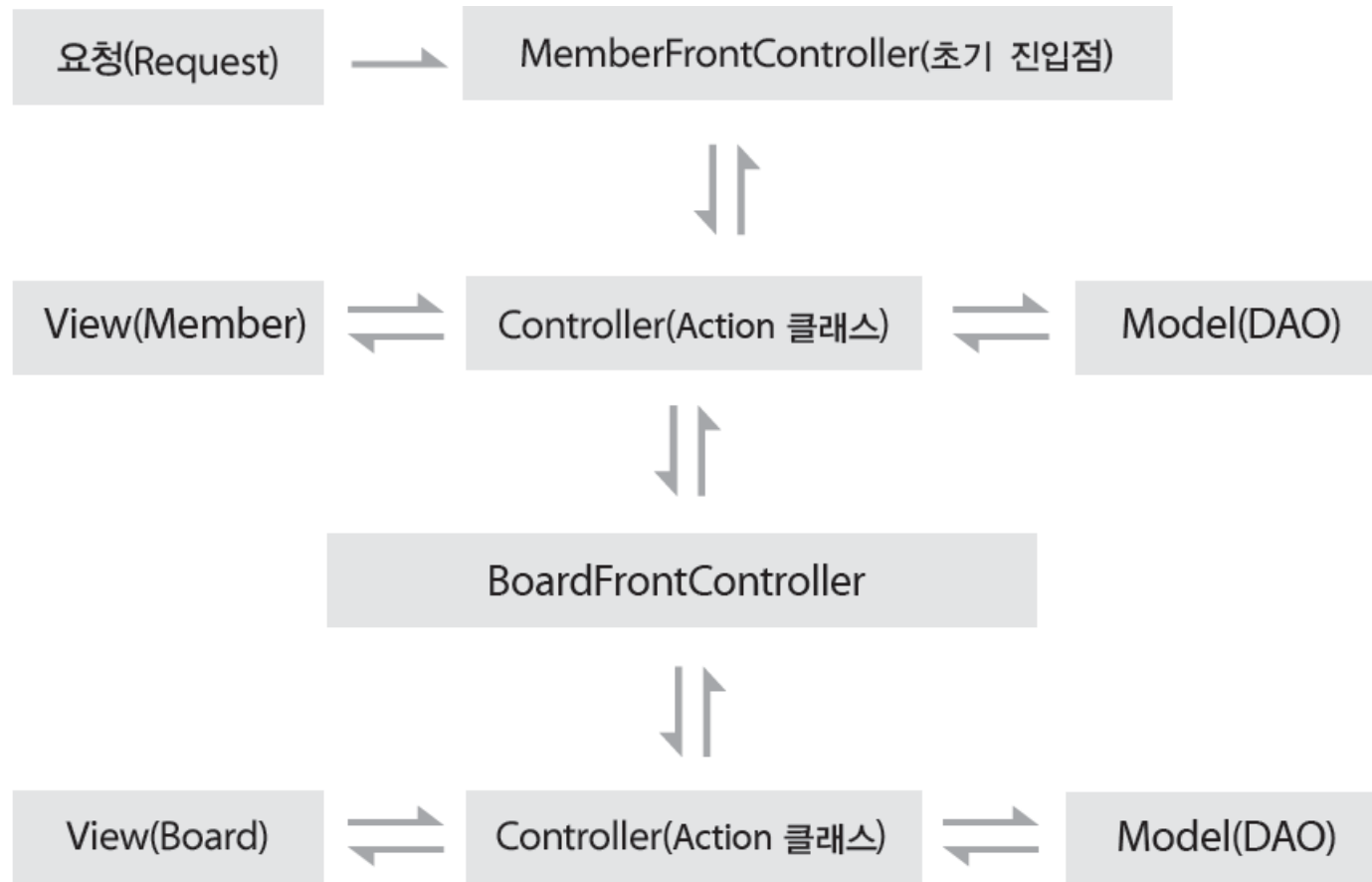
MVC 패턴을 사용하여 간단한 게시판 만들기

- 11. 다음처럼 글 수정이 제대로 이루어지는 것을 확인할 수 있다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

□ 프로젝트의 흐름도



[그림 14-4] 프로젝트 흐름도



회원관리 시스템과 결합한 답변형 게시판 프로젝트

□ View 페이지와 Action 클래스의 설계

■ 회원 시스템의 View 페이지 목록

페이지 파일	설명
loginForm.jsp	로그인 폼 페이지이다.
joinForm.jsp	회원가입 폼 페이지이다.
member_list.jsp	관리자가 회원 목록을 볼 수 있는 페이지이다.
member_info.jsp	관리자가 회원 정보를 확인할 수 있는 페이지이다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

■ 회원 시스템의 Action 클래스의 목록

클래스 파일	설명
MemberLoginAction.java	사용자가 로그인할 때 호출되는 Action 클래스이다.
MemberJoinAction.java	회원가입 처리를 할 때 호출되는 Action 클래스이다.
MemberListAction.java	회원의 목록을 확인할 때 호출되는 Action 클래스이다.
MemberViewAction.java	회원의 정보를 확인할 때 호출되는 Action 클래스이다.
MemberDeleteAction.java	회원의 정보를 삭제할 때 호출되는 Action 클래스이다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

■ 게시판의 View 페이지 목록

페이지 파일	설명
qna_board_list.jsp	게시판의 목록을 보여주는 페이지이다.
qna_board_view.jsp	게시판 글 내용을 보여주는 페이지이다.
qna_board_write.jsp	게시판 글 쓰기 폼 페이지이다.
qna_board_reply.jsp	게시판 글 답변 폼 페이지이다.
qna_board_modify.jsp	게시판 글 수정 폼 페이지이다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

■ 게시판의 Action 클래스 목록

클래스 파일	설명
BoardAddAction.java	글을 등록할 때 호출되는 Action 클래스이다.
BoardDeleteAction.java	글을 삭제할 때 호출되는 Action 클래스이다.
BoardDetailAction.java	글 내용을 확인할 때 호출되는 Action 클래스이다.
BoardListAction.java	글 목록을 확인할 때 호출되는 Action 클래스이다.
BoardModifyAction.java	수정 처리를 하는 Action 클래스이다.
BoardModifyView.java	수정 폼을 보여주는 Action 클래스이다.
BoardReplyAction.java	답변 처리를 하는 Action 클래스이다.
BoardReplyView.java	답변 폼을 보여주는 Action 클래스이다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

□ 데이터베이스 구조 설계

■ MEMBER 테이블의 구조

필드명	타입	설명
MEMBER_ID	VARCHAR2(15)	회원의 아이디를 의미한다.
MEMBER_PW	VARCHAR2(13)	회원의 비밀번호를 의미한다.
MEMBER_NAME	VARCHAR2(15)	회원의 이름을 의미한다.
MEMBER_AGE	NUMBER	회원의 나이를 의미한다.
MEMBER_GENDER	VARCHAR2(5)	회원의 성별을 의미한다.
MEMBER_EMAIL	VARCHAR2(30)	회원의 이메일 주소를 의미한다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

■ MEMBERBOARD 테이블의 구조

필드명	타입	설명
BOARD_NUM(PK)	NUMBER	글 번호를 의미한다.
BOARD_ID	VARCHAR2(20)	글 작성자의 아이디를 의미한다.
BOARD_SUBJECT	VARCHAR2(50)	글의 제목을 의미한다.
BOARD_CONTENT	VARCHAR2(2000)	글의 내용을 의미한다.
BOARD_FILE	VARCHAR2(50)	글에 첨부될 파일의 이름을 의미한다.
BOARD_RE_REF	NUMBER	글의 그룹 번호를 의미한다.
BOARD_RE_LEV	NUMBER	답변 글의 깊이를 의미한다.
BOARD_RE_SEQ	NUMBER	답변 글의 순서를 의미한다.
BOARD_READCOUNT	NUMBER	글의 조회수를 의미한다.
BOARD_DATE	DATE	글의 작성 날짜를 의미한다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 테이블들을 SQL문으로 변환하여 데이터베이스에 추가하기
 - MEMBER 테이블을 SQL문으로 변환하기

```
CREATE TABLE MEMBER(  
  MEMBER_IDVARCHAR2(15),  
  MEMBER_PWVARCHAR2(13),  
  MEMBER_NAMEVARCHAR2(15),  
  MEMBER_AGENUMBER,  
  MEMBER_GENDERVERCHAR2(5),  
  MEMBER_EMAILVARCHAR2(30),  
  PRIMARY KEY(MEMBER_ID)  
);
```



회원관리 시스템과 결합한 답변형 게시판 프로젝트

- **MEMBERBOARD** 테이블을 SQL문으로 변환하기

```
CREATE TABLE MEMBERBOARD(  
    BOARD_NUMNUMBER,  
    BOARD_IDVARCHAR2(20),  
    BOARD_SUBJECTVARCHAR2(50),  
    BOARD_CONTENTVARCHAR2(2000),  
    BOARD_FILEVARCHAR2(50),  
    BOARD_RE_REFNUMBER,  
    BOARD_RE_LEVNUMBER,  
    BOARD_RE_SEQNUMBER,  
    BOARD_READCOUNTNUMBER,  
    BOARD_DATEDATE,  
    PRIMARY KEY(BOARD_NUM)  
);
```

```
ALTER TABLE MEMBERBOARD  
ADD CONSTRAINT pk_board_id FOREIGN KEY(board_id)  
REFERENCES member(member_id);
```



회원관리 시스템과 결합한 답변형 게시판 프로젝트

□ 데이터 빈(DataBean) 클래스와 DAO 클래스의 작성

- 교재 436p ~ 454p 참조

□ FrontController(프론트 컨트롤러) 작성

- 교재 455p ~ 460p 참조

□ Action 클래스 작성

- 교재 461p ~ 481p 참조

□ View 페이지 작성

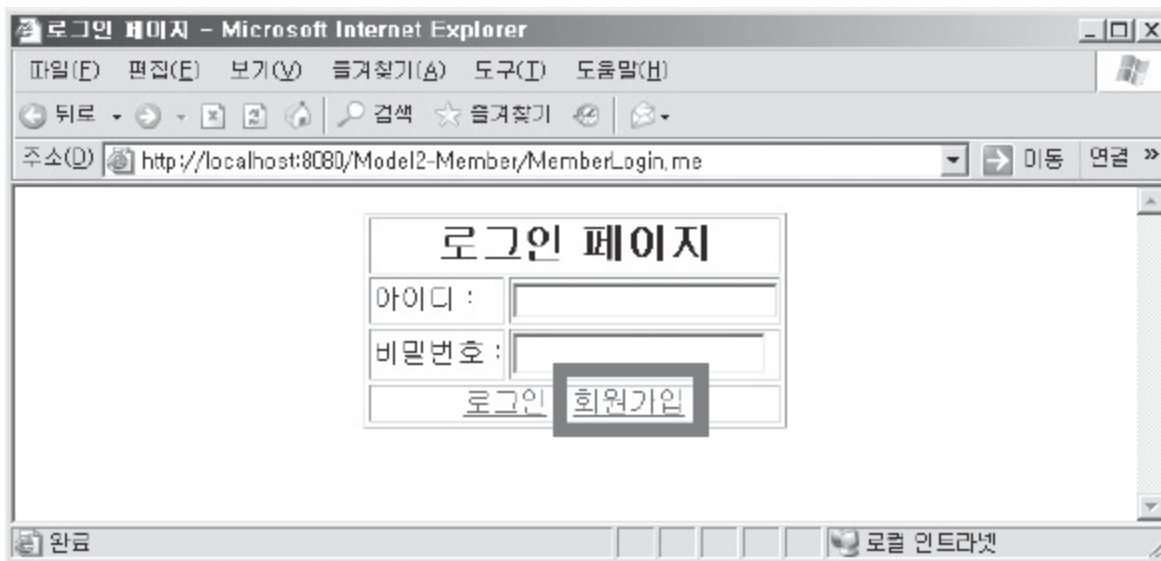
- 교재 481p ~ 498p 참조



회원관리 시스템과 결합한 답변형 게시판 프로젝트

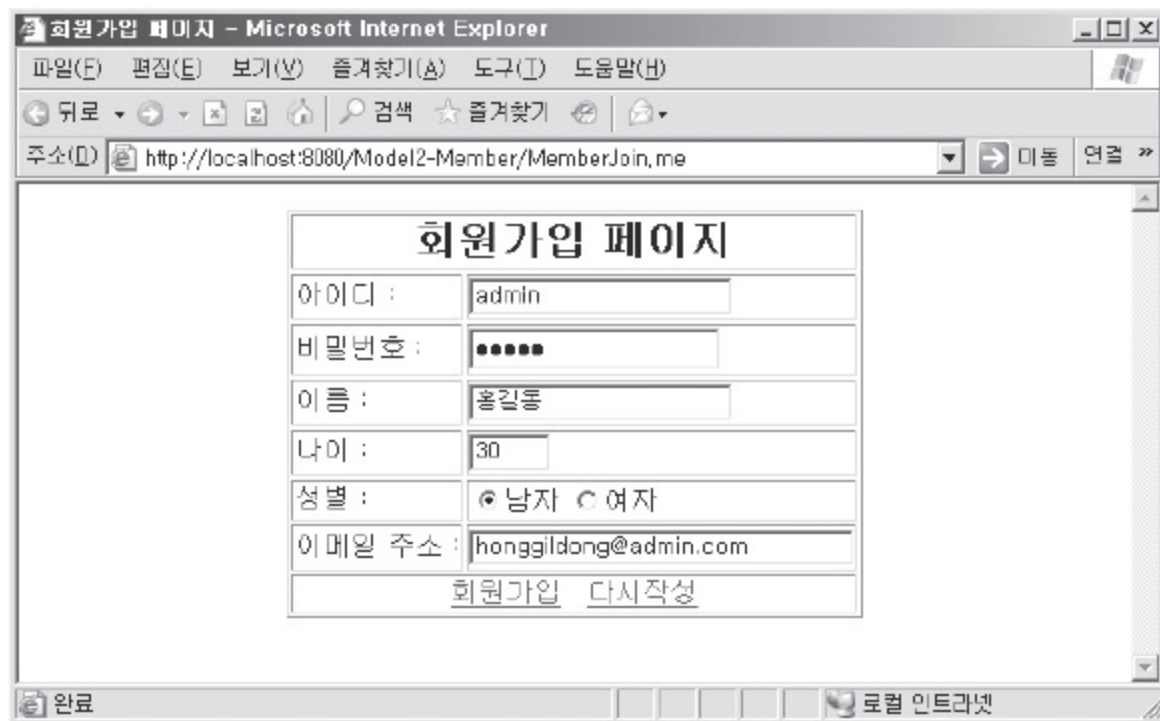
□ 프로젝트의 동작 확인

- 01. 브라우저를 띄운 후 MemberLogin.me를 호출한다. 로그인 페이지가 보이며, 처음에 데이터베이스에는 테이블 구조만 생성되어 있으므로 어떤 레코드도 들어 있지 않을 것이다. 즉, 어떤 아이디도 존재하지 않는 것이므로 ‘회원가입’ 을 클릭하여 아이디를 생성하도록 한다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 02. 테스트를 위하여 아이디는 적당히 3~4개 정도를 생성하도록 한다. 아이디 1개는 반드시 'admin' 으로 생성하여야 한다. 아이디를 모두 생성하였다면, 다시 로그인 페이지로 돌아올 것이다. 생성한 관리자 아이디인 'admin' 으로 접속하도록 한다.



The screenshot shows a Microsoft Internet Explorer window with the address bar displaying `http://localhost:8080/Model2-Member/MemberJoin.me`. The main content area contains a registration form titled "회원가입 페이지". The form fields are as follows:

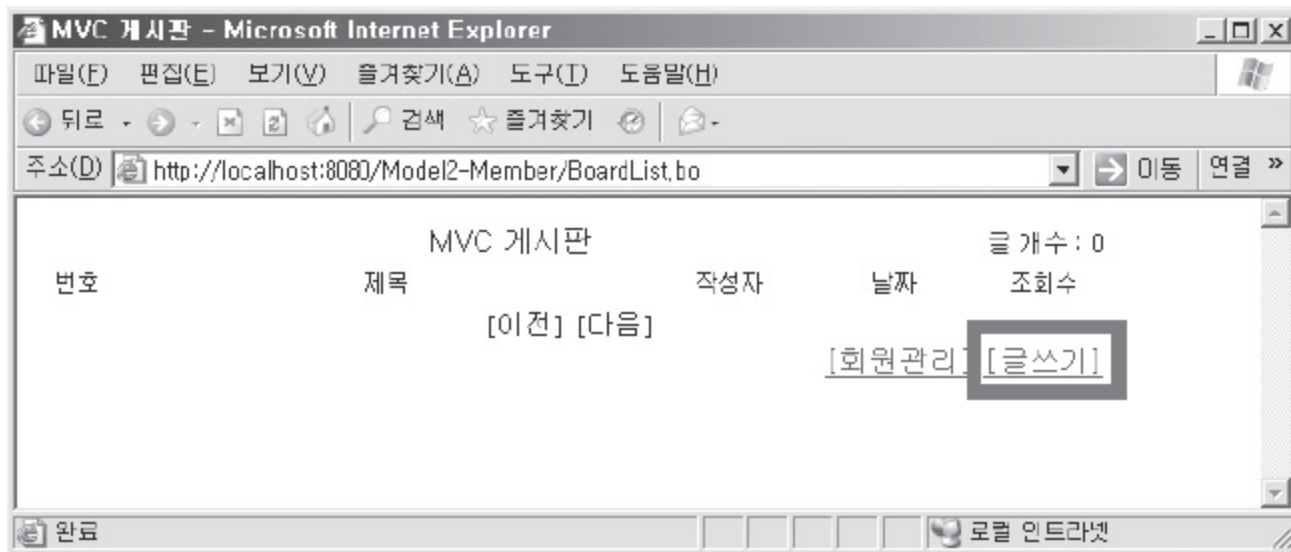
아이디 :	admin
비밀번호 :	*****
이름 :	홍길동
나이 :	30
성별 :	<input checked="" type="radio"/> 남자 <input type="radio"/> 여자
이메일 주소 :	honggildong@admin.com
회원가입 다시작성	

The status bar at the bottom indicates "로컬 인트라넷" (Local Intranet).



회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 03. 로그인이 제대로 이루어지면 게시판 목록 보기 페이지로 화면이 이동된다. 게시판도 마찬가지로 테이블 구조만 생성되어 있을 뿐 레코드는 하나도 존재하지 않으므로 글이 하나도 존재하지 않는다. 글을 작성하기 위해 [글쓰기]를 클릭한다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 04. 글쓴이 셀을 살펴보면 로그인한 아이디가 출력되어 있는 것을 확인할 수 있다. 이것은 로그인 후 로그인한 아이디가 세션에 저장되어 있기 때문에 가능한 것이다. 글을 작성하고 [등록]을 클릭한다.

The screenshot shows a Microsoft Internet Explorer window titled "MVC 게시판 - Microsoft Internet Explorer". The address bar displays "http://localhost:8080/Model2-Member/BoardWrite.bo". The page content includes a form for writing a board post. The form fields are as follows:

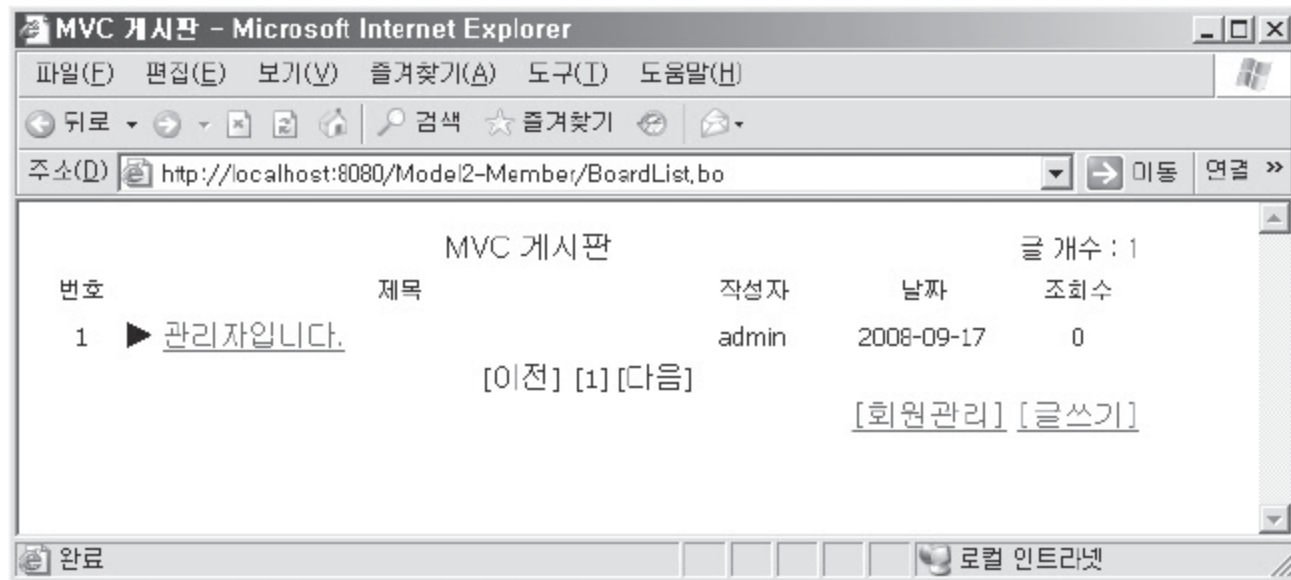
- 글쓴이 (Author):** admin
- 제 목 (Subject):** 관리자입니다.
- 내 용 (Content):** 안녕하세요. 저는 관리자입니다.
- 파일 첨부 (File Attachment):** C:\wabc.txt

At the bottom of the form, there are two buttons: "[등록]" (Register) and "[취소]" (Cancel). The "[등록]" button is highlighted with a red box. The browser's status bar at the bottom indicates "로컬 인트라넷" (Local Intranet).



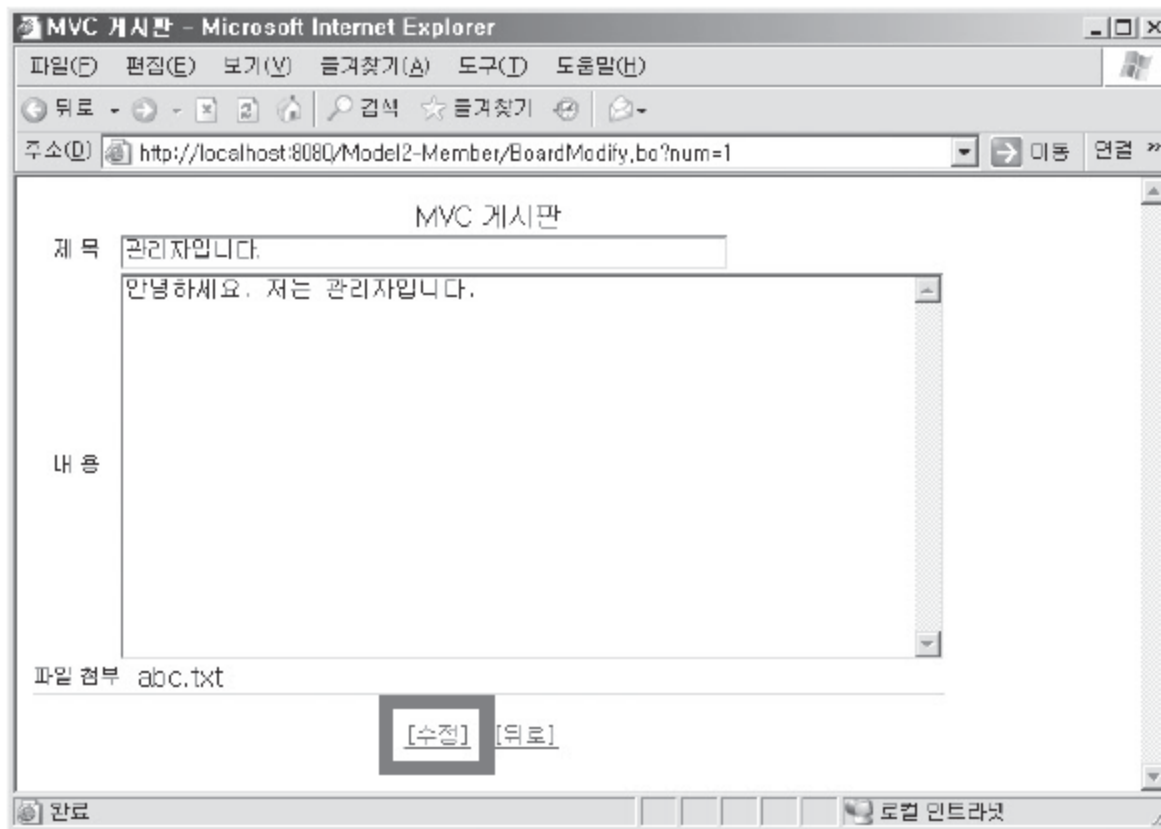
회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 05. 이와 같이 글이 1개 작성되었다.



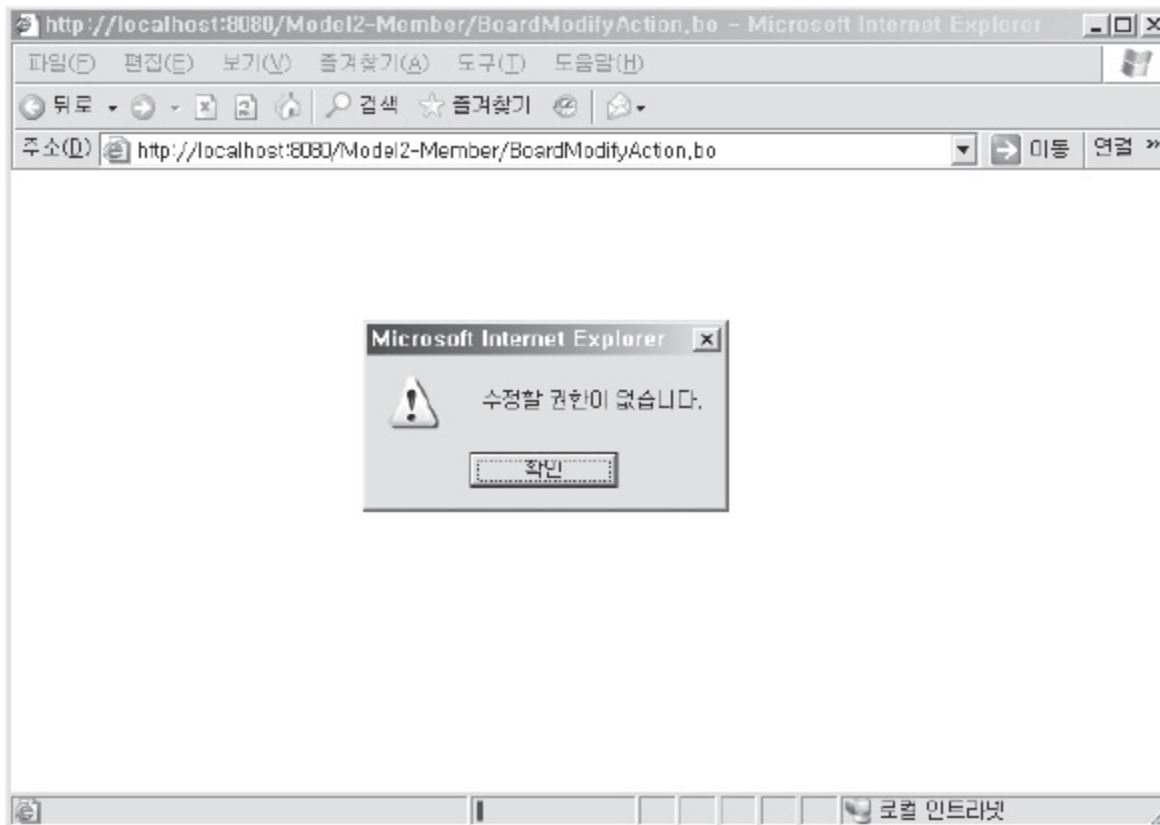
회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 06. 이제 다시 주소 입력창에 MemberLogin.me를 다시 호출하여 다른 아이디로 로그인하고, 글을 수정해 보도록 한다.



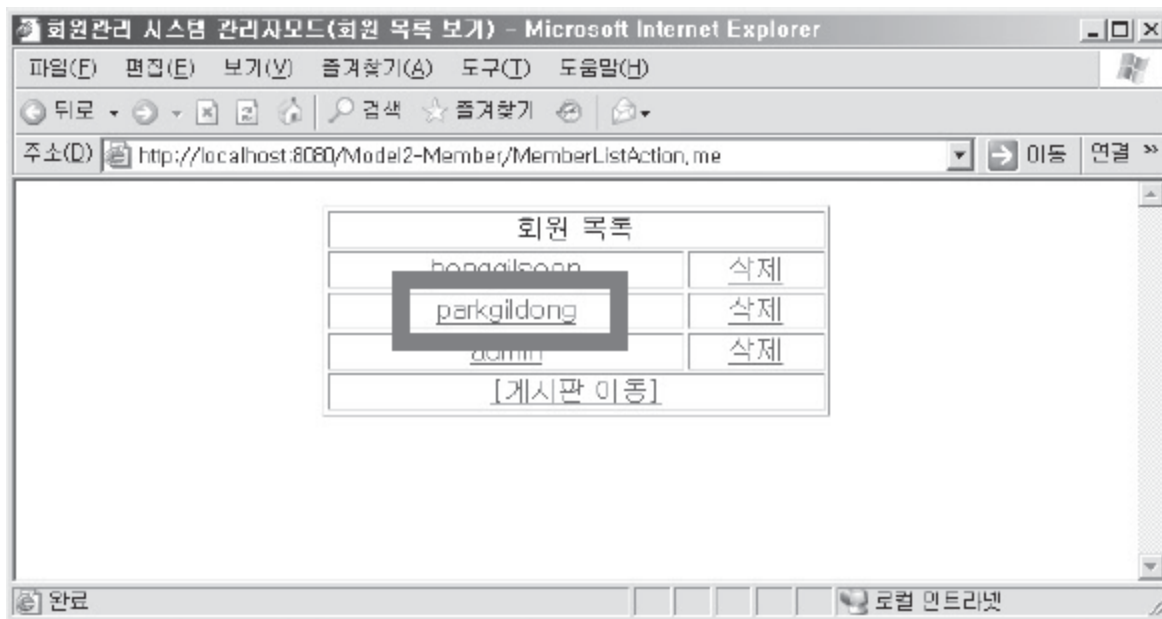
회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 07. 이 같이 글쓴이가 아닌 다른 아이디로 로그인하여 글을 수정하려고 하면 권한이 없다는 오류가 발생한다. 글 작성자와 세션에 설정된 아이디와 비교하여 일치하지 않을 경우에는 수정 처리를 하지 않는 것이다. 삭제 또한 이와 같이 글 작성자와 세션의 아이디를 비교하여 처리하게 된다.



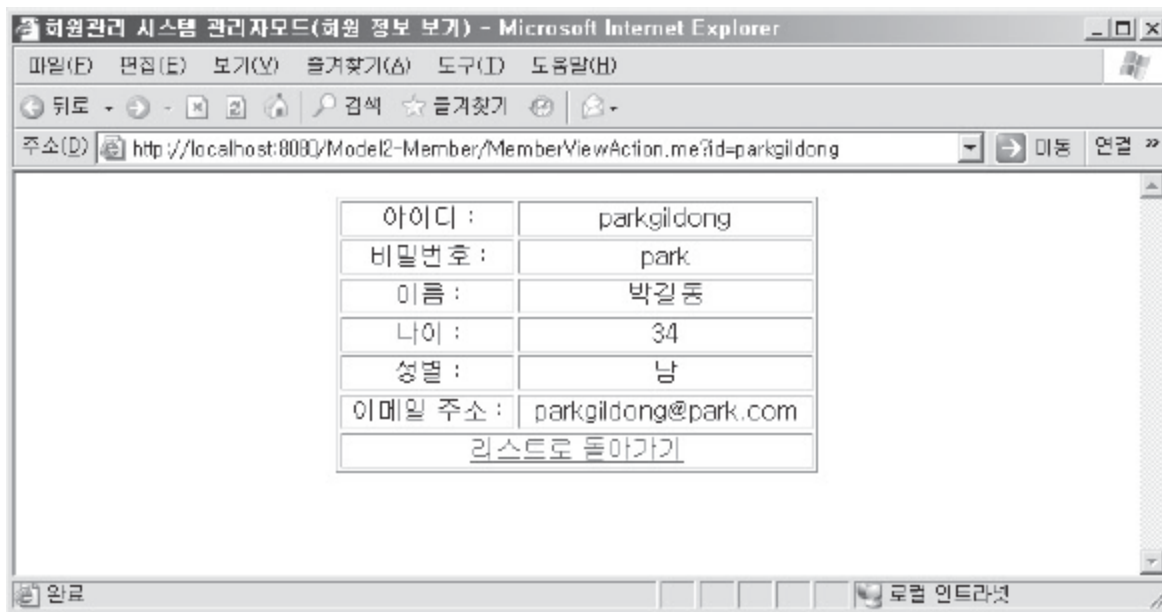
회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 08. 이제 다시 MemberLogin.me를 호출하여 admin 계정으로 로그인 하도록 한다. 그리고 게시판 페이지에서 [회원관리]를 누르도록 한다. [회원관리]로 들어오게 되면 회원 목록이 보여지게 된다. 이 화면에서 회원 아이디를 클릭하여 회원의 정보를 볼 수 있으며, 회원 아이디 옆에 있는 삭제를 클릭하여 회원의 아이디를 삭제할 수 있다. 먼저 회원의 정보를 확인해 보도록 한다.



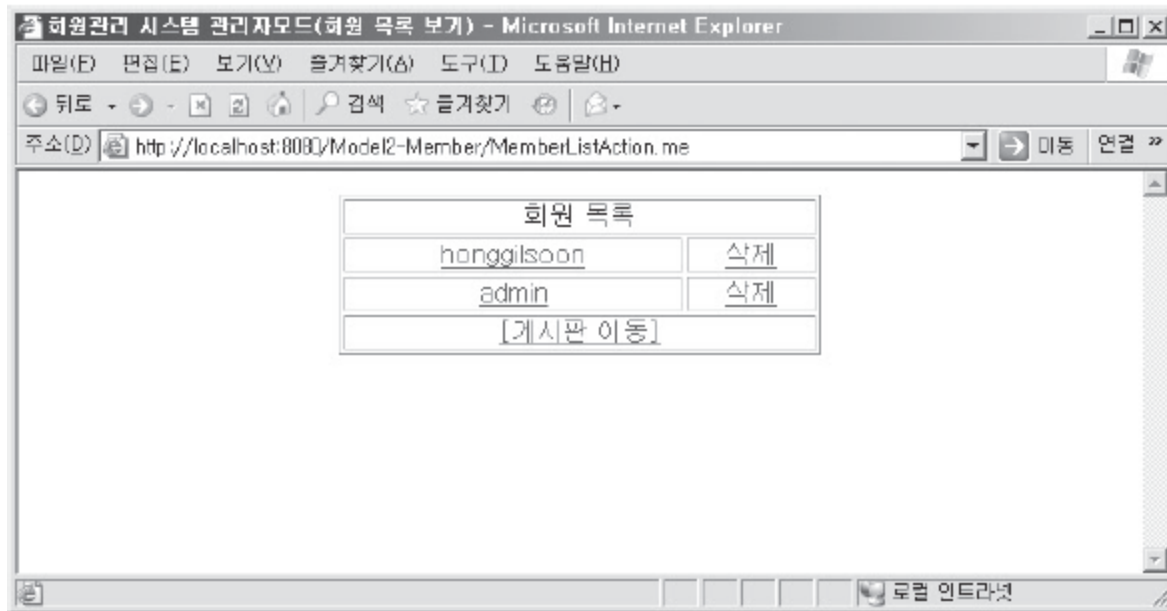
회원관리 시스템과 결합한 답변형 게시판 프로젝트

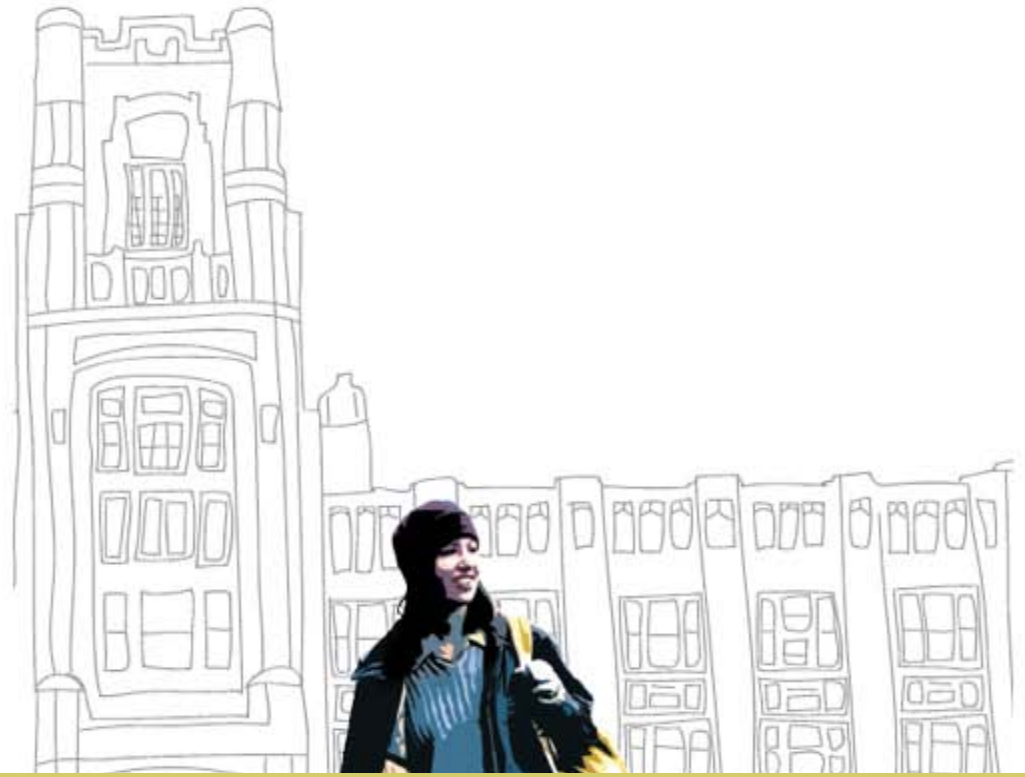
- 09. 주소 표시줄을 보면 id라는 파라미터 이름으로 ‘parkgildong’ 이라는 값이 넘어갔다. 이것은 MemberViewAction 클래스에서 parkgildong 아이디의 정보를 확인하도록 처리해주게 된다. DAO에서는 member 테이블에 접근하여 아이디, 비밀번호, 이름 등 회원의 정보를 가져와서 View 페이지에 보여준다.



회원관리 시스템과 결합한 답변형 게시판 프로젝트

- 10. 이제 다시 회원 목록으로 돌아와 ‘삭제’ 를 클릭해 아이디를 삭제해 보도록 한다.
- 11. 좀 전에 확인했던 parkgildong 이라는 아이디를 삭제하였다. 게시판 목록 보기 페이지를 확인하면 해당 회원이 작성한 글도 모두 삭제되는 것을 확인할 수 있다.





Thank You