



JSP 포트폴리오

- 렌터카 _ DB
- 게시판 _ MVC2
- 교보문고 관리자 _ MVC2

contents

I . 공통 개요

01 개발 배경

02 개발 환경

II . 프로젝트 _ 렌터카 / 게시판 / 교보문고 관리자

A. 산출물

03 기능 명세서

04 주요 코드

05 테이블 정의서

06 요구사항 정의서

B. 라이브러리

07 화면 소개



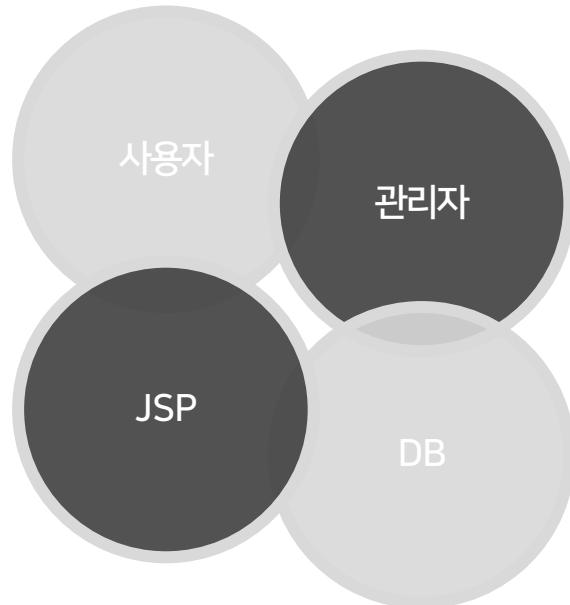
I. 공통 개요

01 개발 배경

02 개발 환경

I . 공통 개요

01 개발 배경

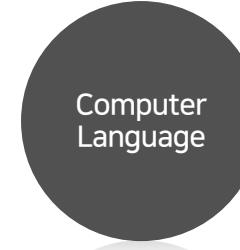
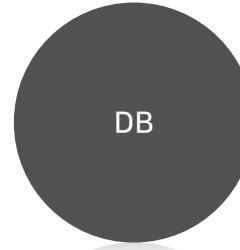
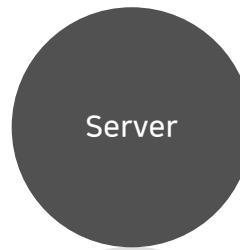
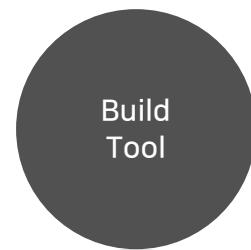


사용자 뿐만 아니라 관리자 화면을 같이
공부함으로써 홈페이지의 전체적인 구성
을 파악해보고 싶었습니다.

특히, 렌터카에서는 게시판 코드를, 관리
자 화면에서는 인터페이스를 활용하여 코
드를 작성하였고 개별적으로 학습을 하여
진행하였습니다.

I . 공통 개요

02 개발 환경



Eclipse

apache-tomcat-8.5.37

MySQL Workbench 8.0 CE

HTML, CSS
JS, JSP
JAVA Servlet



II . 프로젝트 _ 게시판

A. 산출물

03 기능 명세서

04 주요 코드

05 테이블 정의서

06 요구사항 정의서

B. 라이브러리

07 화면 소개



렌터카 _ DB

II. 프로젝트 _ 렌터카

03 기능 명세서



II. 프로젝트 _ 렌터카

04 주요 코드

Java

JSP

▼ _04rentcar

- ▶ CarReserveBean.java
- ▶ CarViewBean.java
- ▶ MemberBean.java
- ▶ RentcarBean.java
- ▶ RentcarDao.java

▶ img

- ▶ 01carMain.jsp
- ▶ 02top.jsp
- ▶ 03bottom.jsp
- ▶ 04center.jsp
- ▶ 05memberLogin.jsp
- ▶ 06memberLoginPro.jsp
- ▶ 07memberLogout.jsp
- ▶ 08carReserveMain.jsp
- ▶ 09carAllList.jsp
- ▶ 10carReserveInfo.jsp
- ▶ 11carOptionSelect.jsp
- ▶ 12carReserveResult.jsp
- ▶ 13carCategoryList.jsp
- ▶ 14carReserveView.jsp
- ▶ 15carReserveDel.jsp

▶ META-INF

II. 프로젝트 _ 렌터카

04 주요 코드

JSP

carMain

```
<table>
    <%--top 부분 --%>
    <tr height = "120" align = "center">
        <td align = "center" width = "1000">
            <jsp:include page="02top.jsp"/>
        </td>
    </tr>

    <%--center 부분 --%>
    <tr align = "center">
        <td align = "center" width = "1000">
            <jsp:include page = "<%= center %>" />
        </td>
    </tr>

    <%--bottom 부분 --%>
    <tr height = "100" align = "center">
        <td align = "center" width = "1000">
            <jsp:include page = "03bottom.jsp"/>
        </td>
    </tr>
</table>
```

- location.href='01_carMain.jsp?center=05_memberLogin.jsp';
- 위 아래는 바꿔지 않고 center 부분만 바꿔도록 화면을 구성하기 때문에 이와 같이 작성합니다.
- session.setAttribute 는 보통 회원가입, 로그인에서 사용을 하고 그 외에 static 변수와 같은 경우 이렇게 설정할 수 있습니다.

II. 프로젝트 _ 렌터카

04 주요 코드

Java

CarReserveBean

```
private int reserve_seq;  
private int no;  
private String id;  
private int qty;  
private int dday;  
private String rday;  
private int usein;  
private int usewifi;  
private int usenavi;  
private int useseat;
```

MemberBean

```
private String id;  
private String pw1;  
private String pw2;  
private String email;  
private String tel;  
private String hobby;  
private String job;  
private String age;  
private String info;
```

- getter/setter 메서드를 통해 값을 설정하고 가져옵니다.

II. 프로젝트 _ 렌터카

04 주요 코드

Java

CarViewBean

RentcarBean

```
private String name;  
private int price;  
private String img;  
private int qty;  
private int dday;  
private String rday;  
private int usein;  
private int usewifi;  
private int useseat;  
private int usenavi;
```

```
private int no;  
private String name;  
private int category;  
private int price;  
private int usepeople;  
private String company;  
private String img;  
private String info;
```

- getter/setter 메서드를 통해 값을 설정하고 가져옵니다.

II. 프로젝트 _ 렌터카

04 주요 코드

Java

RentcarDao

```
public Connection getConnection() {
    String dbURL = "jdbc:mysql://localhost:3306/rentcardb04?serverTimezone=UTC";
    String dbID = "root";
    String dbPassword = "root";

    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        conn = DriverManager.getConnection(dbURL, dbID, dbPassword);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return conn;
}
```

- DB를 연동하고 차량을 select하거나 예약을 하는 등의 이벤트를 관리 합니다.

II. 프로젝트 _ 렌터카

04 주요 코드_예약 화면

JSP

carReserveMain

- 차량의 이미지와 함께 이름을 보여줍니다.

```
<tr height = "240">
<%
for (int i=0;i<v.size();i++) {
    RentcarBean bean = v.get(i);
    %>
    <td width = "333" align = "center">
        <a href = "01carMain.jsp?center=10carReserveInfo.jsp?no=<%=bean.getNo() %>">
            <img alt = "" src = "img/<%=bean.getImg() %>" width = "300" height = "220">
        </a>
        <p>
            차량명 : <%=bean.getName() %>
        </td>
    <%
}
%>
</tr>
```

II. 프로젝트 _ 렌터카

04 주요 코드_예약 화면

JSP

carReserveMain

- 소형, 중형, 대형에 따라 분류를 나누어서 보여줄 수 있습니다.

```
<hr size = "3" color = "red">
<p>
<font size = "4" color = "gray"><b>차량 검색 하기</b></font>

<form action = "01carMain.jsp?center=13carCategoryList.jsp" method = "post">
    <font size = "3" color = "gray"><b>차량 검색 하기</b></font>&nbsp;&nbsp;
    <select name = "category">
        <option value = "1">소형</option>
        <option value = "2">중형</option>
        <option value = "3">대형</option>
    </select>
    <input type = "submit" value = "검색" />&nbsp;&nbsp;
</form>

<%--button은 form밖에 위치시키기 --%>
<button onclick = "location.href='01carMain.jsp?center=09carAllList.jsp'">전체 검색</button>
```

II. 프로젝트 _ 렌터카

04 주요 코드_예약 화면

JSP

carCategoryList

- 이와 같이 카테고리 값을 받아오고 list를 보여줍니다.

```
int category = Integer.parseInt(request.getParameter("category"));

String temp = "";
if (category == 1) {
    temp = "소형";
}
else if (category == 2) {
    temp = "중형";
}
else if (category == 3) {
    temp = "대형";
}
```

II. 프로젝트 _ 렌터카

05 테이블 정의서

rentcar			
테이블명	컬럼_ID	Type	Length
1	no	int	
2	name	varchar	20
3	category	int	
4	price	int	
5	usepeople	int	
6	company	varchar	50
7	img	varchar	50
8	info	varchar	500

member			
테이블명	컬럼_ID	Type	Length
1	id	varchar	20
2	pw1	varchar	20
3	email	varchar	50
4	tel	varchar	20
5	hobby	varchar	60
6	job	varchar	15
7	age	varchar	10
8	info	varchar	500

II. 프로젝트 _ 렌터카

05 테이블 정의서

테이블명 carreserve			
No	컬럼_ID	Type	Length
1	reserve_seq	int	
2	no	int	
3	id	varchar	50
4	qty	int	
5	dday	int	
6	rday	varchar	50
7	usein	int	
8	usewifi	int	
9	usenavi	int	
10	usesseat	int	

II. 프로젝트 _ 렌터카

06 요구사항 정의서

No	대분류	요구사항 명	요구사항 내용
1	메인 화면	로그인	로그인을 해야지만 예약이 가능하다.
		예약하기	차량을 선택하여 예약을 할 수 있다.

No	대분류	요구사항 명	요구사항 내용
2	예약하기	차량 검색	전체 자동차 목록을 보여준다.
		차량 선택	카테고리 및 차량을 선택한다.
		옵션 선택	네비게이션, 베이비 시트, 이용 기간 등을 선택한다.
		예약 완료	최종 결제 금액을 알려주며 예약이 완료됨을 알려준다.

No	대분류	요구사항 명	요구사항 내용
2	예약확인	예약 정보 확인	예약 정보를 확인할 수 있다.

II. 프로젝트 _ 렌터카

07 화면 소개



예약하기 예약확인 자유게시판 이벤트 고객센터

GUEST님 [로그인]

Grandeur Hybrid

로그인

아이디	<input type="text"/>
패스워드	<input type="password"/>
<input type="button" value="로그인"/>	

- 로그아웃 상태에서는 GUEST로 표시하고, 로그인 후 아이디를 넣어줍니다.

II. 프로젝트 _ 렌터카

07 화면 소개



CAR RENTAL

qwer님 [로그아웃](#)

예약하기

예약확인

자유게시판

이벤트

고객센터

최신형 자동차



차량명 : SM5



차량명 : 레이



차량명 : 올라도

- 예약하기, 예약확인 기능

차량 검색하기

차량검색하기

II. 프로젝트 _ 렌터카

07 화면 소개



qwer님 [로그아웃](#)

[예약하기](#) [예약확인](#) [자유게시판](#) [이벤트](#) [고객센터](#)

[전체 렌트카 보기](#)



차량명 : 아반떼



차량명 : BMW



차량명 : 카니발



차량명 : 카렌스



차량명 : 코란도



차량명 : 애쿠스



차량명 : 제네시스



차량명 : 그랜저



차량명 : i30

- 예약하기를 클릭 시,
전체 차량을 보여줍니다.

II. 프로젝트 _ 렌터카

07 화면 소개



qwer님 [로그아웃](#)

예약하기 예약확인 자유게시판 이벤트 고객센터

k5 차량 선택



차량이름	k5
차량수량	1
차량분류	중형
대여가격	5000원
제조회사	기아

[옵션선택 후 구매하기](#)

- 예약 화면이며 옵션을 선택 해야 합니다.

차량 정보 보기

k5 자동차입니다.

II. 프로젝트 _ 렌터카

07 화면 소개



qwer님 [로그아웃]



옵션 선택



대여기간	1일
대여일	2020. 08. 27.
보험적용	적용(1일 1만원)
Wifi 적용	적용(1일 1만원)
네비게이션 적용	적용(무료)
베이비시트 적용	적용(1일 1만원)
차량예약하기	

• 옵션 선택

II. 프로젝트 _ 렌터카

07 화면 소개



qwer'님 [로그아웃]



예약 확인

자유게시판

이벤트

고객 센터

차량 예약 완료 화면

차량 총 예약 금액 5000 원

차량 총 옵션 금액 30000 원

차량 총 금액 35000 원



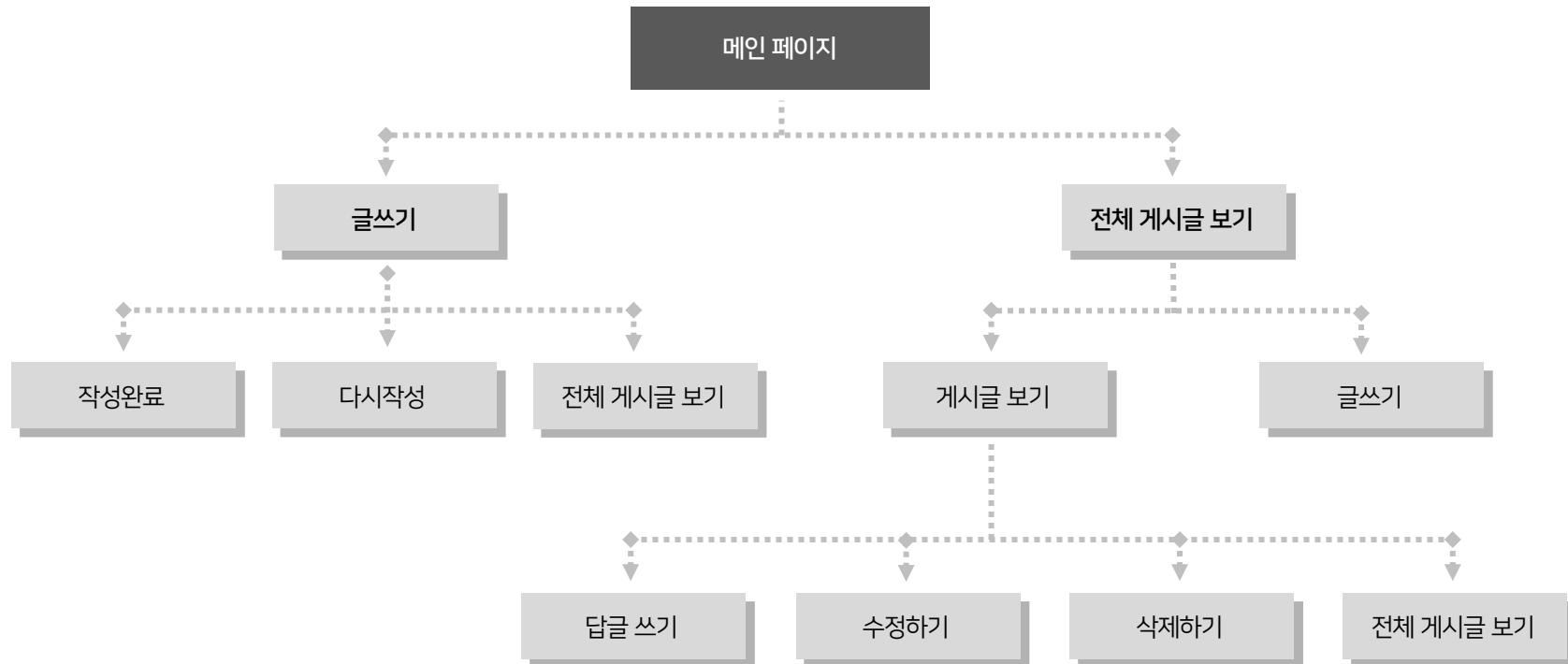
- 예약이 완료된 화면입니다.
- 총 결제금액을 알려줍니다.



게시판_ MVC2

II. 프로젝트 _ 게시판

03 기능 명세서



II. 프로젝트 _ 게시판

04 주요 코드

Java	Java servlet	JSP
<ul style="list-style-type: none">✓ model<ul style="list-style-type: none">> BoardBean.java> BoardDAO.java	<ul style="list-style-type: none">✓ control<ul style="list-style-type: none">> BoardListAction.java> BoardWriteAction.java> DeleteAction.java> InfoAction.java> ReWriteAction.java> ReWriteProAction.java> UpdateAction.java> UpdateActionPro.java	<ul style="list-style-type: none">✓ WebContent<ul style="list-style-type: none">> META-INF> WEB-INF> 01write.jsp> 02boardList.jsp> 03boardInfo.jsp> 04reWrite.jsp> 05boardUpdate.jsp

II. 프로젝트 _ 게시판

04 주요 코드

Java

BoardBean

```
private int num;
private String writer;
private String email;
private String subject;
private String password;
private String reg_date;
private int ref;
private int re_step;
private int re_level;
private int readcount;
private String content;
```

- getter/setter메서드를 통해 private 변수를 가져올 수 있습니다.

Java

BoardDAO

```
private BoardDAO() {};
private static BoardDAO instance = new BoardDAO();
public static BoardDAO getInstance() {
    return instance;
}
```

- Singleton으로 작성은 하고 db연동을 통해 삽입, 수정, 삭제 등의 작업을 제어하도록 합니다.

II. 프로젝트 _ 게시판

04 주요 코드_답변 작성

Java Servlet

RewriteAction

```
int num = Integer.parseInt(request.getParameter("num"));
BoardDAO bdao = BoardDAO.getInstance();
BoardBean bean = bdao.getOneBoard(num);

int ref = bean.getRef();
int re_step = bean.getRe_step();
int re_level = bean.getRe_level();

request.setAttribute("ref", ref);
request.setAttribute("re_step", re_step);
request.setAttribute("re_level", re_level);
```

- 답변을 적는 글의 ref, re_step, re_level을 가져온 뒤 값을 저장합니다.

- ref : 첫 번째 게시글의 경우 1이며 답변이 아닌 게시글이 작성될 때마다 +1을 해줍니다.
- re_level : 글의 깊이를 의미하며, 게시글과 답변을 구분하지 않고 +1을 해줍니다.
- re_step : 각 게시글에 대한 답변의 깊이를 의미합니다. 클릭한 글의 step에 +1을 해줍니다.

II. 프로젝트 _ 게시판

04 주요 코드_답변 작성

JSP

reWrite

```
<%-- form에서 사용자로부터 데이터를 입력받지 않고 데이터를 넘김 --%>
<td align = "center" colspan = "2">
    <input type = "hidden" name = "ref" value = ${ref }>
    <input type = "hidden" name = "re_step" value = ${re_step }>
    <input type = "hidden" name = "re_level" value = ${re_level }>
    <input type = "submit" value = "답글쓰기완료"/>&nbsp;&nbsp;
    <input type = "reset" value = "취소"/>
    <input type = "button" onclick = "location.href='BoardListAction'" value = "전체글보기",
```

- 데이터를 넘겨서 Servlet에서 처리를 해줍니다.

II. 프로젝트 _ 게시판

04 주요 코드_답변 작성

Java Servlet

RewriteProAction

```
bean.setRef(Integer.parseInt(request.getParameter("ref")));
bean.setRe_step(Integer.parseInt(request.getParameter("re_step")));
bean.setRe_level(Integer.parseInt(request.getParameter("re_level")));

BoardDAO bdao = BoardDAO.getInstance();
bdao.reWriteBoard(bean);
```

- BoardDAO의 reWriteBoard메서드에서 ref, re_step, re_level에 대한 설정이 이루어집니다.

II. 프로젝트 _ 게시판

04 주요 코드_답변 작성

Java

BoardDAO

```
String levelsql = "UPDATE board SET re_level=re_level+1 WHERE ref = ? and re_level > ?";  
  
//쿼리 실행 객체 선언  
stmt = conn.prepareStatement(levelsq);  
stmt.setInt(1, ref);  
stmt.setInt(2, re_level);  
  
//쿼리 실행  
stmt.executeUpdate();
```

stmt.setInt(6, ref); => 부모의 값을 넣어준다.
stmt.setInt(7, re_step + 1); => 답글이기에 +1을 더해준다.
stmt.setInt(8, re_level + 1); => 깊이이므로 +1을 더해준다.

II. 프로젝트 _ 게시판

05 테이블 정의서

테이블명			
No	컬럼_ID	Type	Length
1	num	int	10
2	writer	varchar	20
3	email	varchar	20
4	subject	varchar	50
5	password	varchar	20
6	reg_date	date	
7	ref	int	10
8	re_step	int	10
9	re_level	int	10
10	readcount	int	10
11	content	varchar	20

II. 프로젝트 _ 게시판

06 요구사항 정의서

No	대분류	요구사항 명	요구사항 내용
1	글쓰기	작성 완료	작성이 완료되면 리스트에 올라가야 한다.
		다시 작성	리셋이 되어 다시 작성할 수 있도록 해야 한다.
		전체 게시글 보기	전체 게시글 list가 보여져야 한다.

No	대분류	요구사항 명	요구사항 내용
2	전체 게시글 보기	게시글 보기	게시글에 대한 info 화면을 보여준 뒤, 답글쓰기, 수정하기, 삭제하기, 전체 게시글 보기 기능을 실행할 수 있어야 한다.
		글쓰기	글쓰기 화면으로 넘어가야 한다.

II. 프로젝트 _ 게시판

07 화면 소개

게시글 쓰기

작성자		
제목		
이메일		
비밀번호		
글내용		
<button>글쓰기</button> <button>다시작성</button>		
전체 게시글 보기		

- 게시글 쓰기 화면입니다.
- 글쓰기와 다시 작성 버튼을 통해 기능을 구현합니다.

II. 프로젝트 _ 게시판

07 화면 소개

전체 게시글 보기

글쓰기				
번호	제목	작성자	작성일	조회수
1	hi	qwer	2020-08-27	0

[1]

- 작성한 글의 list를 보여줍니다.
- 제목을 클릭하게 되면 글에 대한 info를 보여줍니다.

게시글 보기

글 번호	1	조회수	1
작성자	qwer	작성일	2020-08-27
이메일	qwer@1234		
제목	hi		
글 내용	hi		
답글쓰기 수정하기 삭제하기 목록보기			

II. 프로젝트 _ 게시판

07 화면 소개

답변글 입력하기

- 답글 쓰기

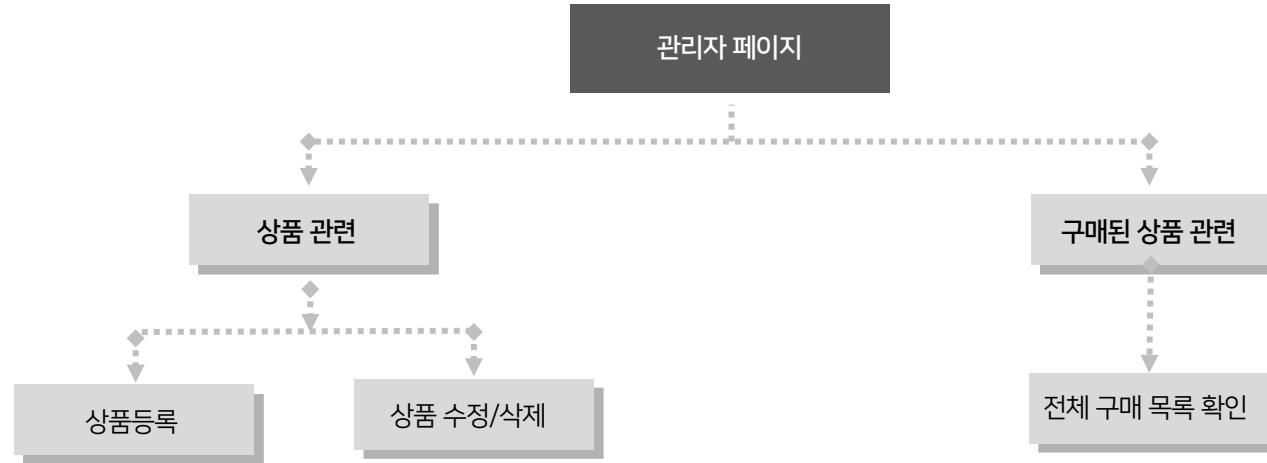
작성자	<input type="text"/>
제목	<input type="text" value="[답변]"/>
이메일	<input type="text"/>
비밀번호	<input type="text"/>
글 내용	<input type="text"/>
<input type="button" value="답글쓰기완료"/> <input type="button" value="취소"/> <input type="button" value="전체글보기"/>	



교보문고 관리자_MVC2

II . 프로젝트 _ 교보문고 관리자

03 기능 명세서



II. 프로젝트 _ 교보문고 관리자

04 주요 코드

Java

- ▼ bookstore.model
 - ▶ ManagerBean.java
 - ▶ ManagerDAO.java
 - ▶ MemberBean.java
 - ▶ MemberDAO.java
- ▼ bookstore.controller
 - ▶ CommandAction.java
 - ▶ Controller.java

- 인터페이스를 활용하여 servlet 연결을 관리합니다.

Java servlet

- ▼ bookstore.command
 - ▶ _01_ManagerMain.java
 - ▶ _02_ManagerLoginPro.java
 - ▶ _03_ManagerLogout.java
 - ▶ _04_BookRegister.java
 - ▶ _05_BookList.java
 - ▶ _06_BookRegisterPro.java
 - ▶ _07_BookUpdate.java
 - ▶ _08_BookUpdatePro.java
 - ▶ _09_BookDeletePro.java
 - ▶ _10_Login.java
 - ▶ _11_Main.java

JSP

- 00_index.jsp
- 01_managerLogin.jsp
- 02_managerMain.jsp
- 03_managerLoginPro.jsp
- 05_managerLogout.jsp
- 06_bookRegister.jsp
- 07_bookList.jsp
- 08_bookRegisterPro.jsp
- 09_bookUpdate.jsp
- 10_bookUpdatePro.jsp
- 11_bookDeletePro.jsp
- 12_login.jsp
- 13_shopMain.jsp

II. 프로젝트 _ 교보문고 관리자

04 주요 코드

Java

CommandAction

```
import javax.servlet.http.HttpServletRequest;[]

public interface CommandAction {
    public String requestPro(HttpServletRequest request, HttpServletResponse response)
        throws Exception;
}
```

- 인터페이스로 구현된 코드이며 각 Servlet에서 상속을 받도록 합니다.
- 예시) BookRegister Servlet

```
public class _04_BookRegister implements CommandAction {
```

II . 프로젝트 _ 교보문고 관리자

04 주요 코드

Java

CommandAction

```
private Map<String, Object> commandMap = new HashMap<String, Object>();  
  
public void init() throws ServletException {  
    commandMap.put("/mg/main.do", new _01_ManagerMain());  
    commandMap.put("/mg/loginPro.do", new _02_ManagerLoginPro());  
    commandMap.put("/mg/logout.do", new _03_ManagerLogout());  
    commandMap.put("/mg/bookRegister.do", new _04_BookRegister());  
    commandMap.put("/mg/bookList.do", new _05_BookList());  
    commandMap.put("/mg/bookRegisterPro.do", new _06_BookRegisterPro());  
    commandMap.put("/mg/bookUpdate.do", new _07_BookUpdate());  
    commandMap.put("/mg/bookUpdatePro.do", new _08_BookUpdatePro());  
    commandMap.put("/mg/bookDeletePro.do", new _09_BookDeletePro());  
    commandMap.put("/login.do", new _10_Login());  
    commandMap.put("/main.do", new _11_Main());  
}
```

II. 프로젝트 _ 교보문고 관리자

04 주요 코드_상품 등록하기

- _04_BookRegister
request.setAttribute("type", 0);
return "/06_bookRegister.jsp";

- 06_bookRegister
form 태그로 상품 등록에 필요한 값들을
bookRegisterPro.do 로 넘긴다.

```
<%-- 책 정보가 중복되지 않는다면 --%>
<c:if test="${ check == 1 }">
    <script type="text/javascript">
        alert('책이 등록되었습니다.');
        location.href='/_jsp_06_bookstore_mvc/mg/bookList.do?book_kind=${ book_kind }';
    </script>
</c:if>
<%-- 책 정보가 중복된다면 --%>
<c:if test="${ check == -1 }">
    <script type="text/javascript">
        alert('이미 등록된 책입니다.');
        history.go(-1);
    </script>
</c:if>
```

II . 프로젝트 _ 교보문고 관리자

05 테이블 정의서

테이블명			
No	컬럼_ID	Type	Length
1	managerId	varchar	50
2	managerPw	varchar	16

테이블명	book		
No	컬럼_ID	Type	Length
1	book_id	int	
2	book_kind	varchar	3
3	book_title	varchar	100
4	book_price	int	
5	book_count	int	
6	author	varchar	40
7	publishing_com	varchar	30
8	publishing_date	varchar	15
9	book_image	varchar	16
10	book_content	varchar	500
11	discount_rate	int	
12	reg_date	varchar	50

II . 프로젝트 _ 교보문고 관리자

05 테이블 정의서

bank			
테이블명	컬럼_ID	Type	Length
1	account	varchar	30
2	bank	varchar	10
3	name	varchar	10

cart			
테이블명	컬럼_ID	Type	Length
1	cart_id	int	
2	buyer	varchar	50
3	book_id	int	
4	book_title	varchar	100
5	book_price	int	
6	book_count	int	
7	book_image	varchar	16

II. 프로젝트 _ 교보문고 관리자

05 테이블 정의서

테이블명				member			
No	컬럼_ID	Type	Length	No	컬럼_ID	Type	Length
1	id	varchar	20	1	buy_id	int	
2	passwd	varchar	16	2	buyer	varchar	50
3	name	varchar	20	3	book_id	varchar	12
4	reg_date	varchar	30	4	book_title	varchar	100
5	address	varchar	100	5	book_price	int	
6	tel	varchar	20	6	buy_count	int	

테이블명				buy			
No	컬럼_ID	Type	Length	No	컬럼_ID	Type	Length
1	buy_id	int		1	buy_id	int	
2	buyer	varchar	50	2	buyer	varchar	50
3	book_id	varchar	12	3	book_id	varchar	12
4	book_title	varchar	100	4	book_title	varchar	100
5	book_price	int		5	book_price	int	
6	buy_count	int		6	buy_count	int	
7	book_image	varchar	16	7	book_image	varchar	16
8	buy_date	varchar	50	8	buy_date	varchar	50
9	account	varchar	50	9	account	varchar	50
10	deliveryName	varchar	10	10	deliveryName	varchar	10
11	deliveryTel	varchar	20	11	deliveryTel	varchar	20
12	deliveryAddress	varchar	10	12	deliveryAddress	varchar	10
13	sanction	varchar	10	13	sanction	varchar	10

II . 프로젝트 _ 교보문고 관리자

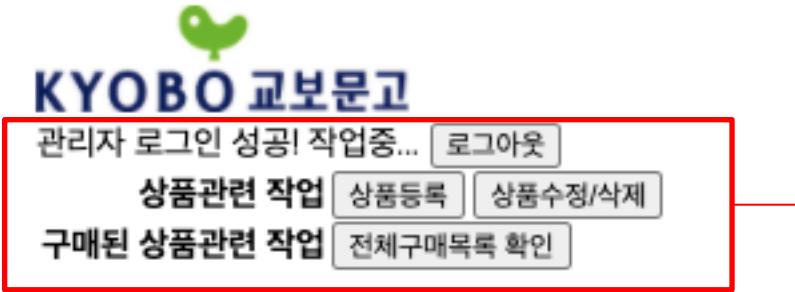
06 요구사항 정의서

No	대분류	요구사항 명	요구사항 내용
1	상품 관련	상품등록	새로운 상품을 등록한다.
		상품 수정/삭제	상품을 수정하거나 삭제한다.

No	대분류	요구사항 명	요구사항 내용
2	구매된 상품 관련	전체 구매 목록 확인	전체 상품의 목록을 확인하고 관리한다.

II. 프로젝트 _ 교보문고 관리자

07 화면 소개



KYOBO 교보문고

관리자 로그인 성공! 작업중... [로그아웃]

관리자 메인으로 목록으로

분류선택	문학
제목	제목
가격	가격 원
수량	수량 권
저자	저자
출판사	출판사
출판일	2010년 1월 1일
책 이미지	파일 선택 선택된 파일 없음
내용	
할인율	10
책등록	

- 관리할 목록을 보여주고, 상품을 등록하거나 수정 및 삭제를 할 수 있습니다.

II. 프로젝트 _ 교보문고 관리자

07 화면 소개



- 책을 등록한 후에는 전체 목록을 보여주고 수정 및 삭제가 가능합니다.



등록된 상품 목록(전체 상품 : 13개)

번호	책분류	제목	가격	수량	저자	출판사	출판일	책이미지	할인율	등록일	수정	삭제
1	100	안녕	10,000원	1개	qwer	qwer	2020-08-27	nothing.jpg	10%	2020-08-27 15:23:32	[수정]	[삭제]
2	100	인간 실격	20,000원	50개	다자이 오사무	민음사	2016-11-20	nothing.jpg	15%	2020-08-25 08:16:33	[수정]	[삭제]
3	100	기묘한 신혼여행	12,000원	30개	정태원	문학의 문학	2017-04-20	nothing.jpg	20%	2020-08-25 08:16:33	[수정]	[삭제]
4	100	연필로 쓰기	18,000원	80개	김훈	문학동네	2018-02-14	nothing.jpg	5%	2020-08-25 08:16:31	[수정]	[삭제]
5	100	나미야 잡화점의 기적	15,000원	100개	히가시노 게이고	현대문학	2019-01-11	nothing.jpg	10%	2020-08-25 08:16:30	[수정]	[삭제]

감사합니다.

지원자 최지완 드림