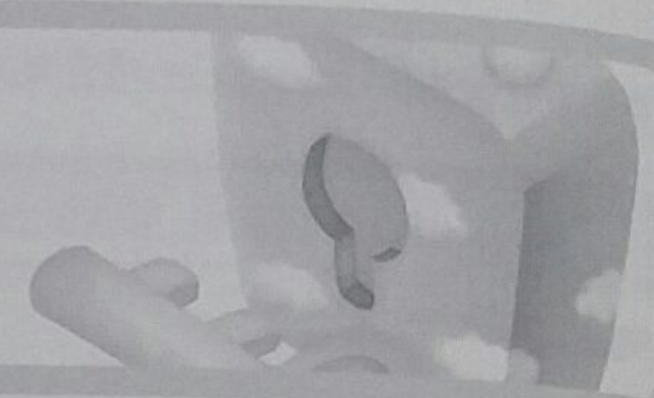
11 페이지와 파일관련



안녕하세요

이번 장에서는 데이터베이스에 저장된 데이터를 리스트로 출력하면서 페이지를 나눠보는 예제, Multipart를 이용한 파일업로드와 다운로드 그리고 이미지 업로드와 활용 예제를 통해 파일 관련된 부분에 대해서 공부한다. 특히 이번 장에서 이미지 업로드를 통해 이미지를 보이는 과정은 실무에서 사용되는 방법으로 많은 도움이 된다.

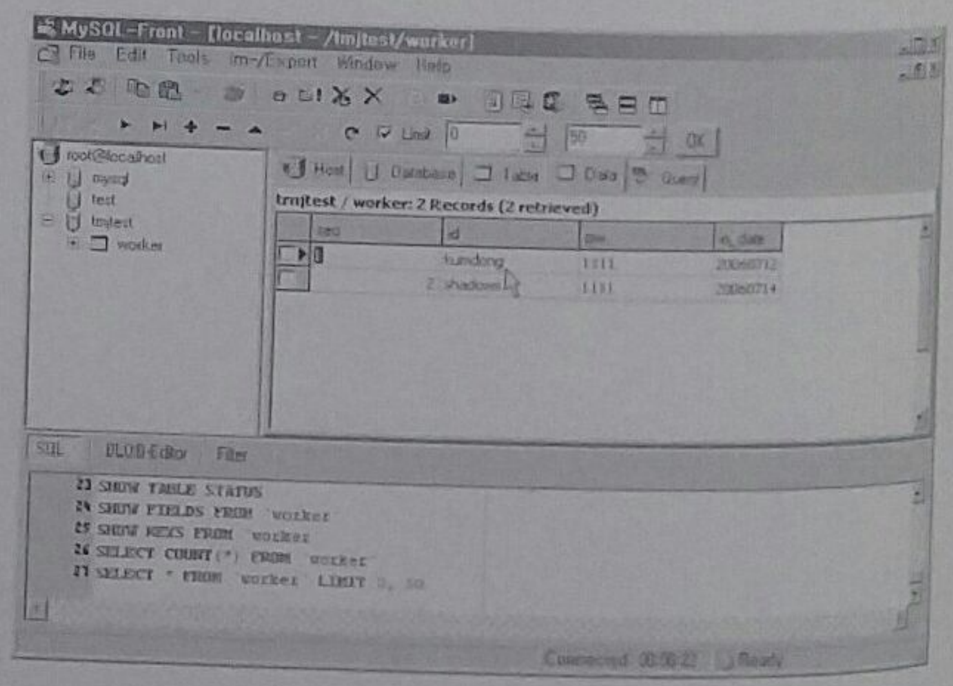
페이지 나누기

앞 장에서 우리는 웹 브라우저를 통해서 데이터베이스인 MySql에 저장, 수정, 삭제하는 것을 배웠다. 이렇게 저장된 데이터가 계속 쌓이다 보면 브라우저 내에서 많은 데이터를 보고자 할 때 데이터베이스의 부하뿐만 아니라 처리속도가 무척이나 늦어질 것이다.

우리는 저장된 데이터를 사용자가 보기 좋게 페이지를 나누어 보여줌으로서 사용자의 편의성을 제공해야 한다. 앞 장에서 만들어진 사용자 등록 예제를 이용하여 리스트를 보여주고 출력된 데이터를 이용하여 페이지 나누기를 공부한다.

따라하기 페이지 나누기

❶ 앞 장에서의 예제를 통해 우리는 MySql에 생성된 데이터베이스 tmjtest라는 데이터베이스에 worker라는 테이블에 그림과 같은 데이터를 저장해 보았다.



▶ 저장된 데이터

11 페이지와 파일관련

안녕하세요

이번 장에서는 데이터베이스에 저장된 데이터를 리스트로 출력하면서 페이지를 나눠보는 예제, Multipart를 이용한 파일업로드와 다운로드 그리고 이미지 업로드와 활용 예제를 통해 파일 관련된 부분에 대해서 공부한다. 특히 이번 장에서 이미지 업로드를 통해 이미지를 보이는 과정은 실무에서 사용되는 방법으로 많은 도움이 된다.

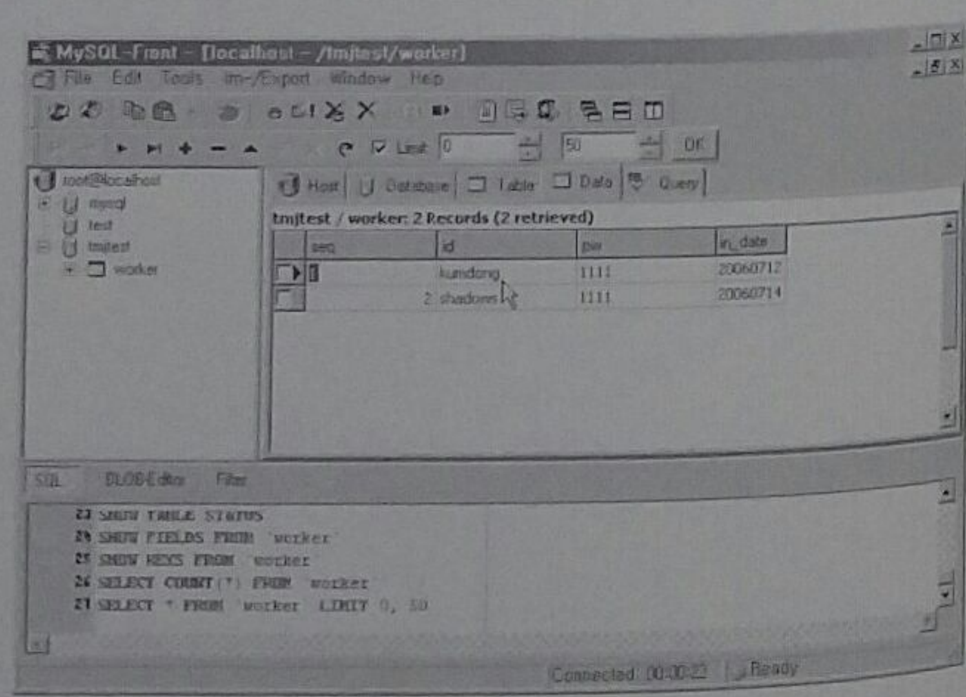
페이지 나누기

앞 장에서 우리는 웹 브라우저를 통해서 데이터베이스인 MySql에 저장, 수정, 삭제하는 것을 배웠다. 이렇게 저장된 데이터가 계속 쌓이다 보면 브라우저 내에서 많은 데이터를 보고자 할 때 데이터베이스의 부하뿐만 아니라 처리속도가 무척이나 늦어질 것이다.

우리는 저장된 데이터를 사용자가 보기 좋게 페이지를 나누어 보여줌으로서 사용자의 편의성을 제공해야 한다. 앞 장에서 만들어진 사용자 등록 예제를 이용하여 리스트를 보여주고 출력된 데이터를 이용하여 페이지 나누기를 공부한다.

따라하기 페이지 나누기

❶ 앞 장에서의 예제를 통해 우리는 MySql에 생성된 데이터베이스 tmjtest라는 데이터베이스에 worker라는 테이블에 그림과 같은 데이터를 저장해 보았다.



▶ 저장된 데이터

이렇게 저장된 데이터를 출력하는 workList.jsp 파일을 작성한다.

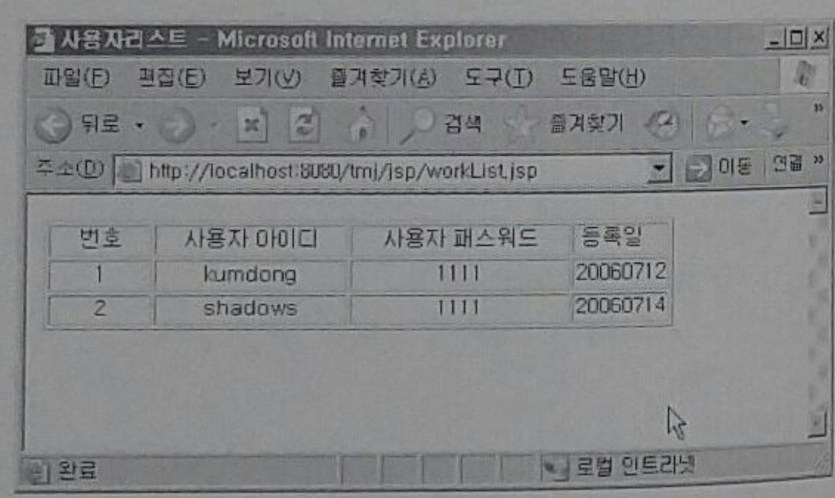
▶ [workList.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                javax.naming.*,javax.sql.DataSource" %>
<%
String listSql="select * from worker";
%>
<%@ include file="../include/try.jsp" %>
<%
  rs = stmt.executeQuery(listSql);
%>
<html>
<head>
<title>사용자리스트</title>
</head>
<body >
<table border="1" width="383">
<tr>
 <td width="60">
   <p align="center">
       <font size="2">번호</font>
   </p>
 </td>
 <td width="115">
   <p align="center">
       <font size="2">사용자 아이디</font>
   </p>
 </td>
 <td width="135">
   <p align="center">
       <font size="2">사용자 패스워드</font>
   </p>
 </td>
 <td width="45">
   <p align="center">
       <font size="2">등록일</font>
   </p>
 </td>
</tr>
<!--while문을 통해 리스트를 출력한다.-->
<%
int num=1;
while(rs.next())
{
%>
<tr>
  <td width="60">
    <p align="center">
        <font size="2"><%=num%></font>
    </p>
  </td>
  <td width="115">
    <p align="center">
       <font size="2"><%=rs.getString("id")%></font>
    </p>
```

다음 페이지에 계속 ➲

```
</td>
<td width="135">
   <p align="center">
      <font size="2"><%=rs.getString("pw")%></font>
   </p>
</td>
<td width="45">
  <p align="center">
     <font size="2"><%=rs.getString("in_date")%></font>
  </p>
</td>
</tr>
<%
//번호에 대한 증가 값
num++;
}
if(num ==1)
{
%>
<tr>
  <td colspan=4 align=center>해당하는 데이터가 없습니다.</td>
</tr>
<%
}
%>
</table>
</body>
</html>
<%@ include file="../include/catch.jsp" %>
```

❷ 브라우저에서 workList.jsp를 호출하면 다음과 같은 화면을 출력한다.



▶ workList.jsp의 호출

❸ 리스트가 호출이 되었다면 리스트의 페이지를 나눌 jsp를 작성한다.

▶ [pageCount.jsp]

```
<hr size=1><br>
<%
//전체보기가 아닐 경우
if (!"ok".equals(request.getParameter("fullview")))
{
  rs=stmt.executeQuery(pageCountQuery);
  if (rs.next())
```

다음 페이지에 계속 ➲

```
{
  if ((rs.getInt(1) % view_rows) == 0)
    total_pages = rs.getInt(1) / view_rows;
  else
  total_pages = (rs.getInt(1) / view_rows) + 1;
}
rs.close();
int start_page = 1;
int end_page = 10;

  if (tpage < 1)
  tpage = 1;
  if (tpage % 10 == 0)
  {
     start_page = tpage - 9;
     end_page = tpage;
  }
  else
  {
    start_page = tpage - (tpage % 10) + 1;
    end_page = start_page + 9;
  }
  if (end_page > total_pages)
  end_page = total_pages;
  if (start_page > 10)
  {
%>
  <a  href="<%=(url+"1")%>">◀◀</a>
    
  <a href="<%=(url+Integer.toString(start_page-1))%>">◀</a> 
<%
}
else if (total_pages > 1 && end_page !=0)
{
%>◁◁  ◁ <%
}
   for(int i=start_page;i<=end_page;i++)
   {
     if(i == tpage)
     {
%>
       <font color=red>[<%=tpage%>]</font> 
<%
     }
     else
     {
%>
       <a href="<%=(url+Integer.toString(i))%>"><%=i%> </a>
<%
     }
   }
   if (total_pages > end_page)
   {
%>
   <a href="<%=(url+Integer.toString(end_page+1))%>">▶</a>
      
   <a href="<%=(url+Integer.toString(total_pages))%>">▶▶</a>
    
```

다음 페이지에 계속

```
    <a href="<%=(url+Integer.toString(tpage)+"&fullview=ok")%>">
    [전체보기]</a>
<%
    }
    else if (total_pages > 1 && end_page !=0)
    {
%>
    ▷  ▷▷ 
    <a href="<%=url+Integer.toString(tpage)+"&fullview=ok"%>">[전체보기]</a>
<%
    }
}
else
{
%>
    <a href="<%=url+Integer.toString(tpage)+"&fullview="%>">[페이지보기]</a>
<%
}
%>
```

❹ 작성된 pageCount.jsp 파일을 workList.jsp에 include하고 필요한 변수를 선언한다.

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%
//리스트를 가지고 오는 쿼리변수
String listSql="select * from worker";
//리스트의 전체 레코드수를 가지고 오는 쿼리 변수
String pageCountQuery = "select count(1) from worker ";

// 페이지 설정을 위한 변수
int view_rows =10;
int total_pages =1;          //전체 페이지 (default 1)
int tpage=1;                 //현재 페이지 (default 1)

//페이지 번호 클릭시 이동되는 주소 변수
String url = "";
url = "./workList.jsp?tpage=";

if (!"ok".equals(request.getParameter("fullview")) &&
                 request.getParameter("tpage") != null) //전체보기 확인
tpage = Integer.parseInt(request.getParameter("tpage"));

if (!"ok".equals(request.getParameter("fullview")))
listSql += " limit "+((tpage-1)*view_rows)+","+view_rows;

%>
<%@ include file="../include/try.jsp" %>
<%
   rs = stmt.executeQuery(listSql);
%>
<html>
<head>
```

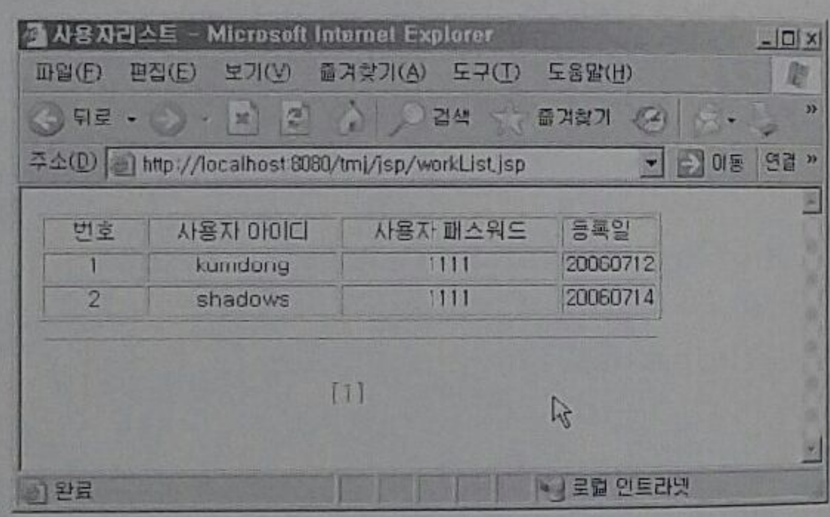
다음 페이지에 계속 ➊

```
<title>사용자 리스트</title>
</head>
<body >
<table border="1" width="383">
<tr>
  <td width="60">
    <p align="center">
      <font size="2">번호</font>
    </p>
  </td>
  <td width="115">
     <p align="center">
       <font size="2">사용자 아이디</font>
     </p>
   </td>
   <td width="135">
      <p align="center">
        <font size="2">사용자 패스워드</font>
      </p>
   </td>
   <td width="45">
      <p align="center">
         <font size="2">등록일</font>
      </p>
   </td>
</tr>
<!--while문을 통해 리스트를 출력한다.-->
<%
int num=1;
if (!"ok".equals(request.getParameter("fullview")))
num = ((tpage-1)*view_rows)+1;
while(rs.next())
{
%>
<tr>
  <td width="60">
     <p align="center">
       <font size="2"><%=num%></font>
     </p>
   </td>
  <td width="115">
      <p align="center">
         <font size="2"><%=rs.getString("id")%></font>
      </p>
  </td>
  <td width="135">
      <p align="center">
        <font size="2"><%=rs.getString("pw")%></font>
      </p>
  </td>
  <td width="45">
      <p align="center">
         <font size="2"><%=rs.getString("in_date")%></font>
     </p>
  </td>
</tr>
<%
//번호에 대한 증가값
```

다음 페이지에 계속

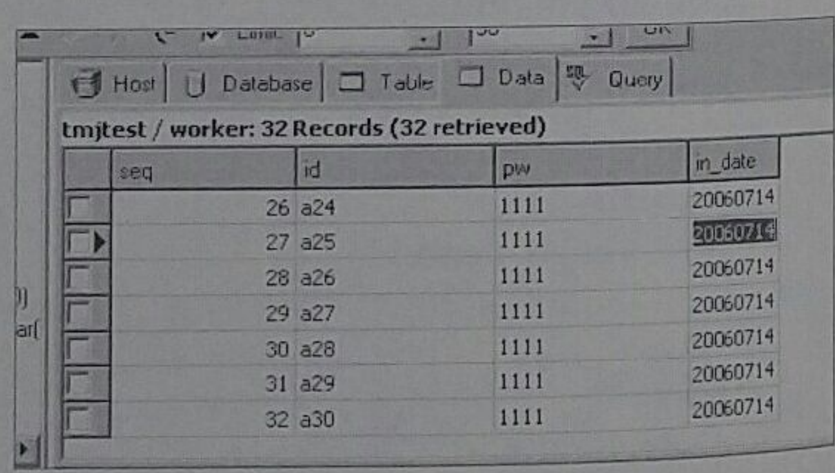
```
num++;
}
if(num ==1)
{
%>
<tr>
    <td colspan=4 align=center>해당하는 데이터가 없습니다.</td>
</tr>
<%
}
%>
</table>
<table border="0" width="383">
 <tr>
    <td align="center">
       <%@ include file="../include/pageCount.jsp"%>
    </td>
  </tr>
</table>
</body>
</html>
<%@ include file="../include/catch.jsp" %>
```

❺ pageCount.jsp에 추가된 workList.jsp 파일을 브라우저에서 호출하면 하단에 페이지가 출력되는 것을 확인한다.



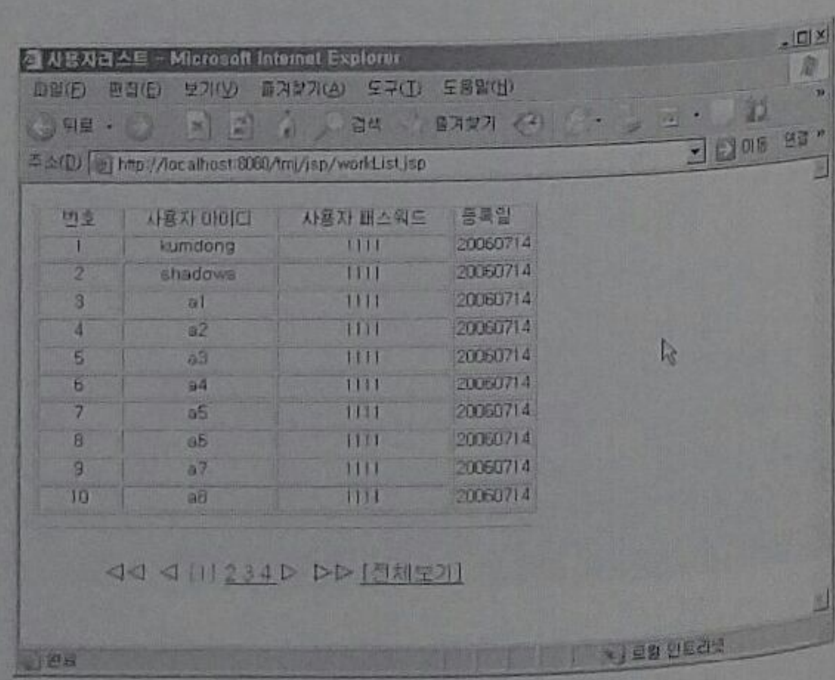
▶ 페이지 인덱스

❻ 작업 확인을 위하여 mySql-Font에서 여러 개의 데이터를 입력하거나 또는 앞 예제를 통해 데이터를 입력해 본다.



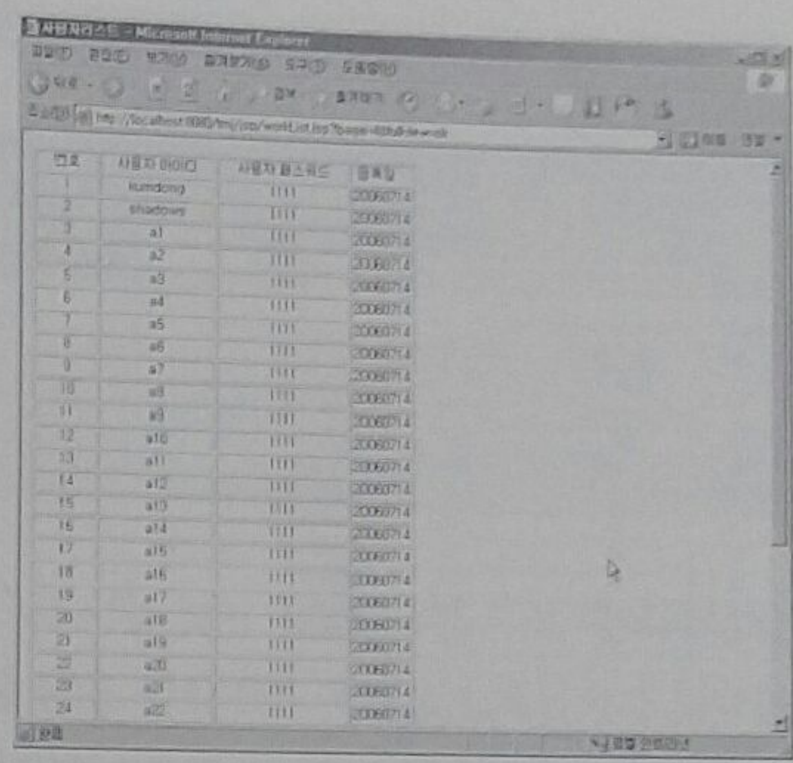
▶ mySqlFont에서 입력된 가상의 데이터

❼ 브라우저에서 workList.jsp를 다시 호출한다.



▶ 페이지가 나누어진 화면

❽ 페이지를 클릭하면 일련번호의 증가와 데이터의 개수가 10개씩 나누어 보이며 전체보기를 클릭하면 페이지 나눔 없이 모든 데이터의 리스트가 출력이 된다. 또한 전체보기에서 페이지 보기를 클릭하면 다시 페이지가 나누어진 화면을 보여준다.



▶ 전체보기

예제를 통해서 우리는 페이지를 나눠 보았다. 이제 페이지 나눔에 대한 상세 부분을 설명하겠다. 처음 workList.jsp 파일을 만든 후에 pageCount.jsp 파일을 따로 작성하여 include하였다. 이는 나중에 또 다른 리스트 페이지가 있다면 공통으로 사용하기 위해 pageCount.jsp를 따로 만든 것이다.

먼저 workList.jsp 파일에서 페이지를 위한 변수 선언을 한다. 소스에서도 나와 있듯이 3개의 변수를 선언하고 다음과 같이 사용하는데 쓰인다.

```
// 페이지 설정을 위한 변수선언
int view_rows =10;  ---------> 페이지에 보일 레코드의 수를 정의
int total_pages =1;  ---------> 전체 페이지
int tpage=1;          ---------> 현재 페이지
```

그런 후 다음을 선언하였다.

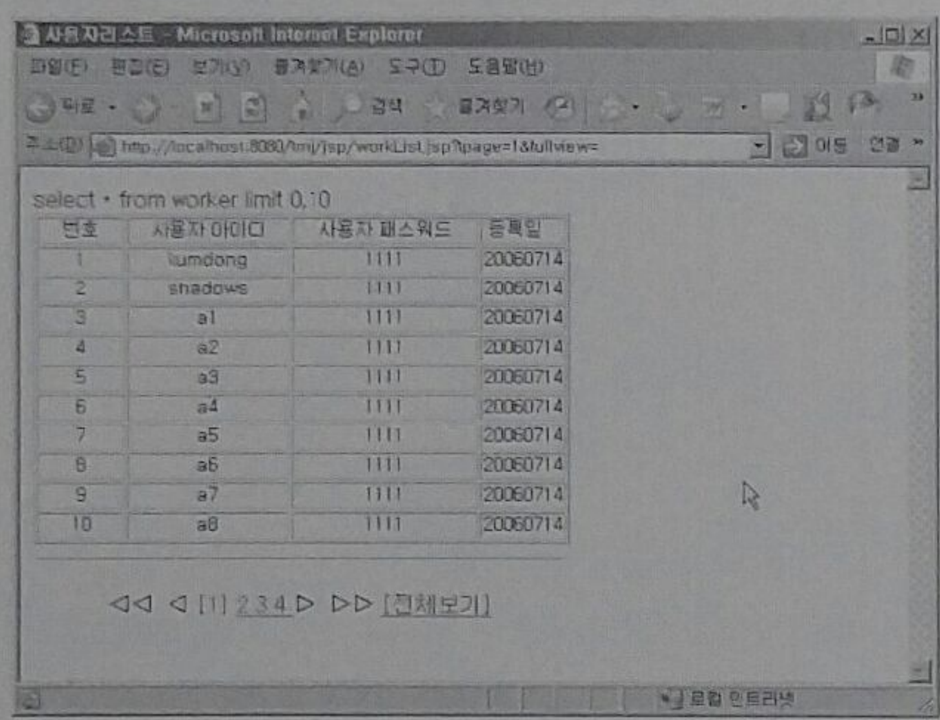
```
if (!"ok".equals(request.getParameter("fullview")) &&
                request.getParameter("tpage") != null)
 tpage = Integer.parseInt(request.getParameter("tpage"));

if (!"ok".equals(request.getParameter("fullview")))
   listSql += " limit "+((tpage-1)*view_rows)+","+view_rows;
```

첫 번째 if문은 전체보기를 클릭하였을 경우 사용하며 파라메타의 값을 1 전달하고 두 번째 if문은 listSql에 limit를 선언 후 limit 함수에 적용되는 값을 listSql에 첨가하는 것이다. 즉 첫 번째 페이지라면 1-1*10은 0이 되고 view_rows의 값은 10이므로 listSql의 쿼리는 select * from worker limit 0,10이 되어 저장된 데이터의 0부터 10개의 데이터만을 출력하게 되는 것이다. 2페이지를 선택했다면 10부터 10개의 데이터를 출력하겠다는 것이다. 쉽게 알기위해 workList.jsp에 listSql를 out.println()으로 출력해 보도록 한다.

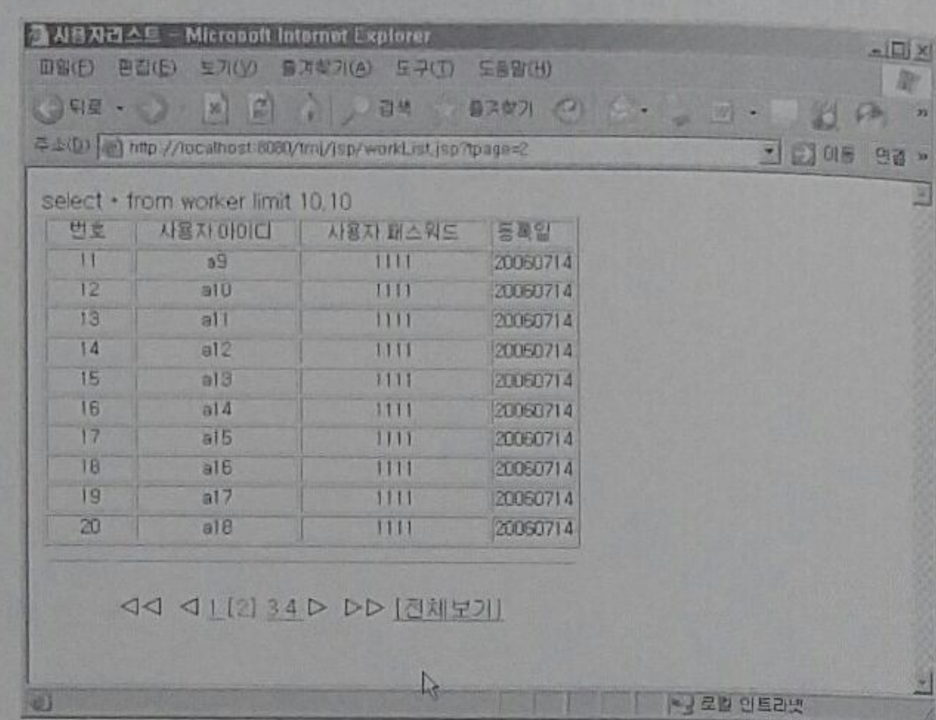
```
if (!"ok".equals(request.getParameter(
   listSql += " limit "+((tpage-1)*vie

out.println(listSql);

<%@ include file="../include/try.jsp"
<%
    rs = stmt.executeQuery(listSql);
```

▶ 소스 내 out.println()의 추가

브라우저를 열고 workList.jsp를 호출하면 listSql문을 출력한다.

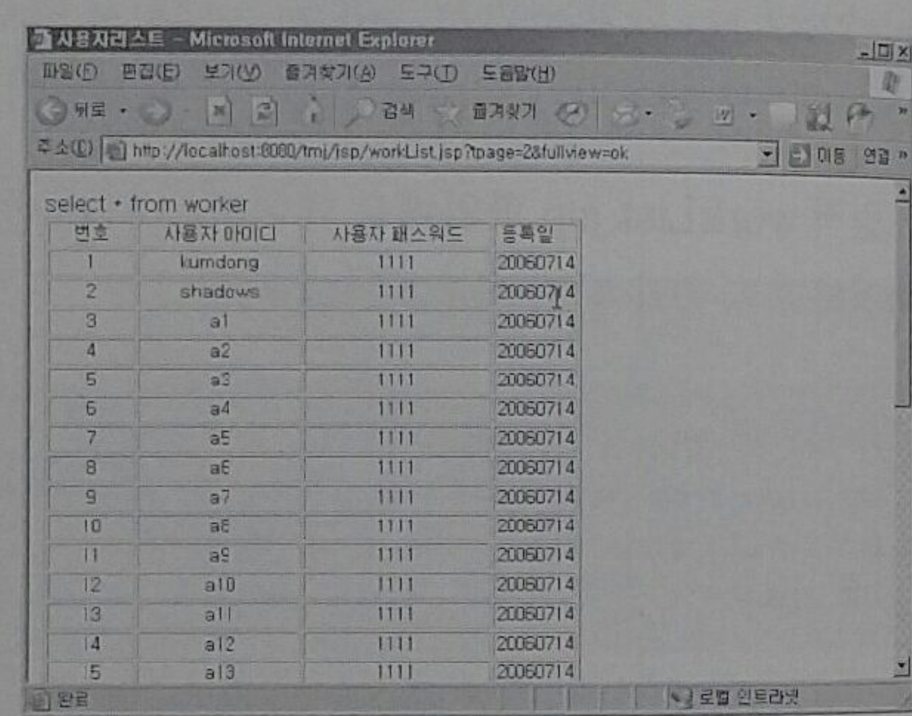


▶ 기본 값이 1일 때의 listSql



▶ 페이지가 2일 때의 listSql

만약 전체 보기를 클릭했다면 listSql은 어떻게 출력이 될까? 당연히 limit가 빠진 쿼리문이 출력된다.



▶ 전체보기를 클릭했을 때의 listSql

이제 페이지를 위한 변수선언과 그 방법에 대해서 이해가 되었다면 pageCount.jsp를 살펴본다.

```
rs=stmt.executeQuery(pageCountQuery);
   if (rs.next()) {
      if ((rs.getInt(1) % view_rows) == 0)
         total_pages = rs.getInt(1) / view_rows;
      else
         total_pages = (rs.getInt(1) / view_rows) + 1;
   }
```

이 문장을 보게 되면 pageCountQuery를 실행하여 나온 결과의 값을 변수에서 선언된 값 10으로 나누어 그것이 0과 같으면 그것을 총 페이지로 하거나 아니면 +1을 하여 총 페이지로 값을 입력한다.

```
if (tpage < 1)
      tpage = 1;
   if (tpage % 10 == 0) {
      start_page = tpage - 9;
      end_page = tpage;
   } else {
      start_page = tpage - (tpage % 10) + 1;
      end_page = start_page + 9;
   }
```

이 문장에서는 선택한 tpage의 값이 1보다 작은 경우 10으로 나눈 것이 0과 같으면 시작 페이지와 끝 페이지에 해당값을 입력한다. 즉 0이면 페이지가 1페이지가 된다. 만약 0이 아니라면 페이지가 있다는 것으로 판단하고 계산된 start_page에 +9를 함으로서 페이지가 나오게 된다. 이러한 값들은 바로 뒤에 나오는 for문에서 페이지 증가 값으로 나타난다.

```
 for(int i=start_page;i<=end_page;i++)
 {
      if(i == tpage)
      {
%>
        <font color=red>[<%=tpage%>]</font> 
<%
      }
      else
      {
%>
      <a href="<%=(url+Integer.toString(i))%>"><%=i%> </a>
<%
      }
    }
```

이 부분이 start_page와 end_page까지 for문을 수행하면서 페이지를 출력하는 부분이다. 또한 다음 10페이지 이전 10 페이지는 List에서 선언된 URL인

```
url = "./workList.jsp?tpage=";
```

tpage의 값에 다음을 링크함으로써 간단하게 해결되었다.

```
<a href="<%=(url+Integer.toString(end_page+1))%>">
```

단 조건이 total_pages 〉 end_page이면 이전의 10개이며 데이터가 있다는 의미로 ▶으로 표시하였고 그렇지 않다면 ▷으로 표시한다. 페이지가 이동이 될 때마다 그 해당 페이지의 번호가 tpage 변수 값에 입력이 되면서 데이터에 따른 시작과 끝이 결정이 되는 것이다.

끝으로 전체 보기를 클릭 시에는 fullview=ok값을 전달하여 앞의 List에서 선언하고 있는

```
if (!"ok".equals(request.getParameter("fullview")))
   listSql += " limit "+((tpage-1)*view_rows)+","+view_rows;
```

부분을 빼고 쿼리를 실행하여 전체의 데이터를 출력하는 것이다. 반대인 경우 즉 페이지를 다시 보기를 한다면 fullview 값에 아무것도 전달하지 않다는 다면 페이지로 나누어져서 보인다. 가능하면 전체 보기는 하지 않는 것이 좋다. 왜냐하면 데이터의 수가 많을 경우, 보통 3000건 이상인 경우에는 속도 저하를 가져올 수 있기 때문이다. 주로 페이지의 view_rows의 값을 변경하여 처리하는 것이 바람직하다.

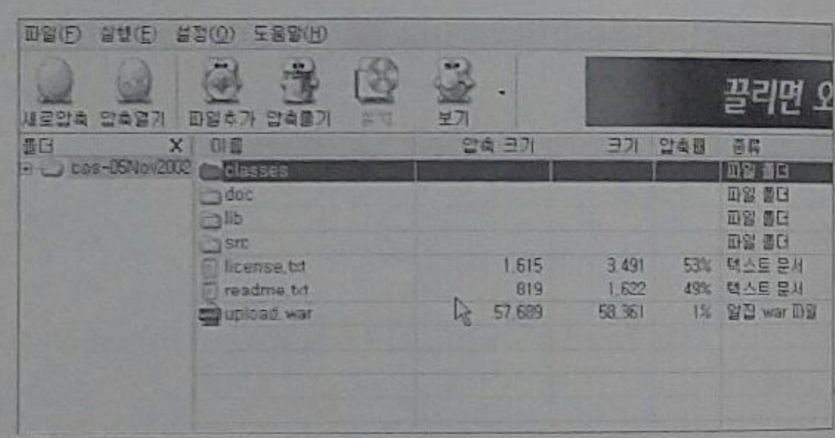
파일 업로드와 다운로드

파일 업로드는 개발 시에 많이 사용되는 방법이다. 많은 양의 데이터를 데이터베이스에 저장할 수 없기 때문에 파일 업로드를 통해 저장의 보완성을 고려한다.

JSP에서의 파일 업로드 방법은 jspSmartUpload와 mutipart가 있다. 이 책에서는 개발자들이 많이 사용하는 mutipart를 이용한 파일 업로드를 공부하게 된다.

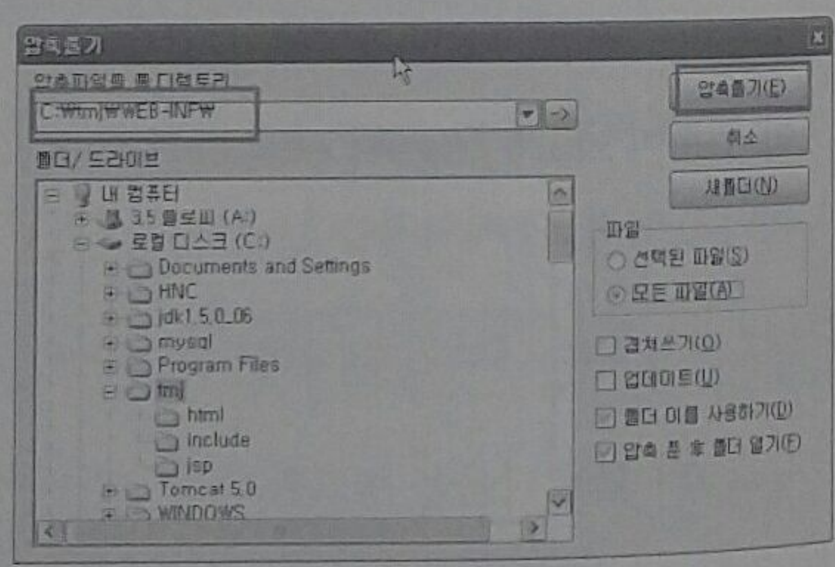
따라하기 파일 업로드

❶ mutilpart를 이용한 파일 업로드를 하기 위해서는 mutilpart에 필요한 class 파일이 필요하다. 도서출판 대림 홈페이지 자료실에서 다운 받은 소스 중에서 cos-05Nov2002.zip 파일의 압축을 푼다.



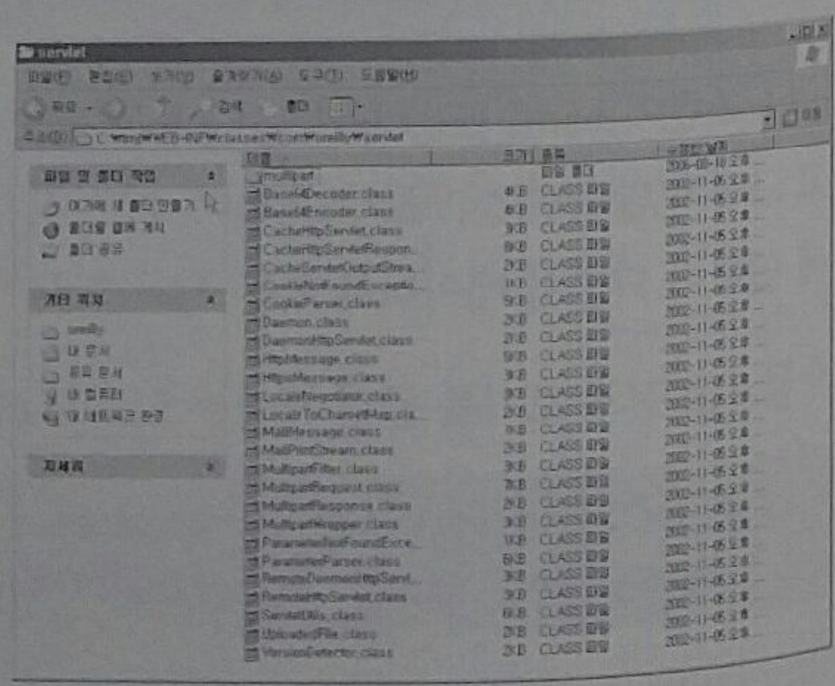
▶ 가상 압축풀기

❷ 화면에서 보는 것처럼 classes 폴더를 우리가 생성한 context 폴더인 tmj에 WEB-INF 폴더를 생성하여 압축을 해지한다.



▶ 압축해지

❸ 압축을 풀고 해당 폴더에 가면 oreilly사의 multipart에 관련된 클래스 파일을 볼 수 있다.



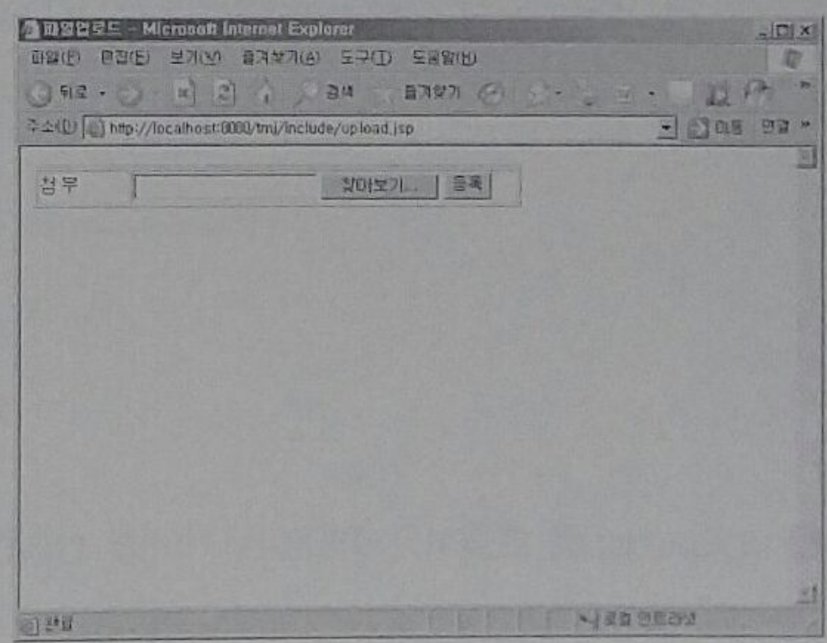
▶ 압축 해지된 결과

❹ mutilpart에 관련된 파일 설치 후 파일을 업로드하는 upload.jsp을 작성한다.

▶ [upload.jsp]

```
<%@ page language="java" contentType="text/html; charset=euc-kr" %>
<html>
<head>
<title>파일업로드</title>
<script language="javascript">
function go_upload()
{
  var TheForm = document.uploadForm;
   if(TheForm.addfile.value == '')
   {
     alert('업로드 할 파일을 선택해 주세요');
   }
   else
   {
     TheForm.action="./uploadSave.jsp";
     TheForm.submit();
   }
}
</script>
</head>
<body>
<form name="uploadForm" method="post" enctype="multipart/form-data">
<table border="1" width="400">
<tr>
 <td width="80">첨부</td>
 <td width="320">
  <input type="file" name="addfile">
  <input type="button" name="up_btn" value="등록"
                                    onClick="javascript:go_upload()">
 </td>
</tr>
</table>
</form>
</body>
</html>
```

여기서 중요한 부분은 enctype="multipart/form-data"이다. 만약 form의 선언이 파일을 업로드할 수 있는 타입으로 선언되지 않았다면 파일 업로드를 할 수 없다. 작성된 upload.jsp 파일을 include 폴더에 저장하고 브라우저를 열고 화면을 확인한다.



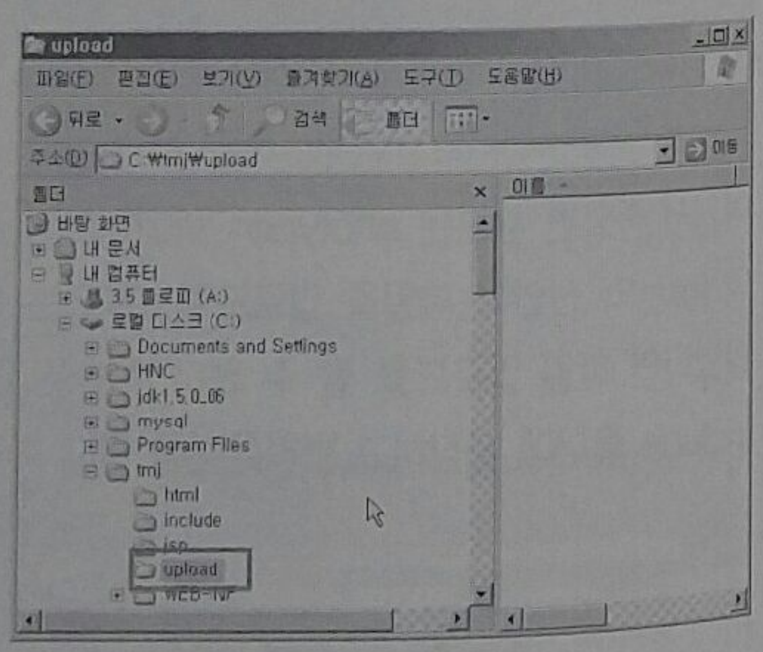
▶ upload.jsp의 호출

❺ 파일을 선택하여 입력받는 upload.jsp가 있다면 이것을 받아 처리하는 uploadSave.jsp 파일을 작성한다.

▶ [uploadSave.jsp]

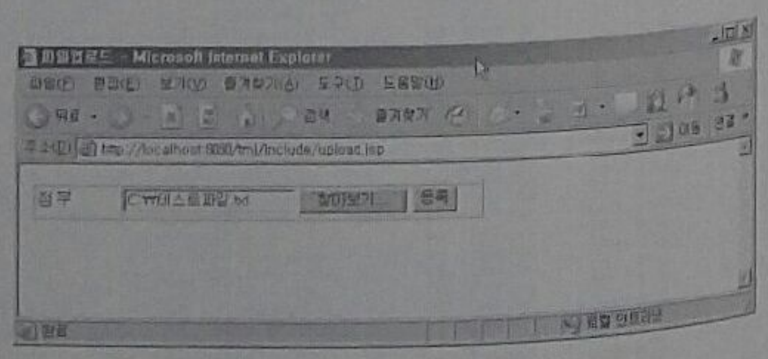
```
<%@ page language="java" contentType="text/html; charset=euc-kr" %>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%
// 저장할 디렉토리 (절대경로)
String savePath="c:/tmj/upload/";
//업로드 할 파일의 이름을 받는 파라메타 변수
String fm = request.getParameter("addfile");
String fileName=" ";
int sizeLimit = 10 * 1024 * 1024 ;      // 파일업로드 용량 제한.. 10Mb
try{

 //만약 같은 이름의 파일이 존재한다면 삭제
 File f = new File ( savePath+fm );
 if (f.exists())
 {
    f.delete();
 }
 else
 {
 //MultipartRequest 객체생성
  MultipartRequest multi =
    new MultipartRequest(request, savePath,sizeLimit,,new DefaultFileRenamePolicy());
    fileName= multi.getFilesystemName("addfile");
    }
}
catch(Exception e) {
 out.print(e.getMessage());
}
%>
```

❻ tmj/upload라는 폴더를 생성한다.



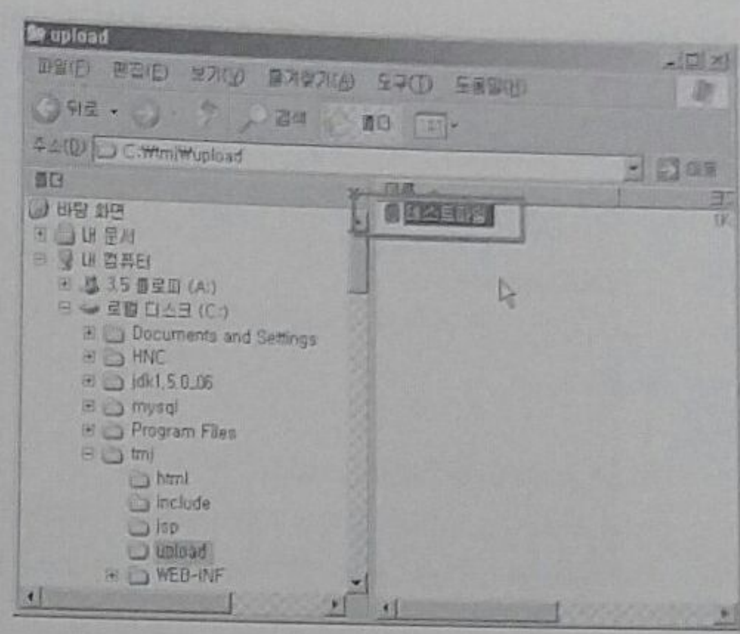
▶ upload 폴더의 생성

❼ upload.jsp를 호출하고 파일(아무거나)을 선택한다.



▶ upload.jsp의 호출

❽ 등록 버튼 클릭 후 tmj/upload 폴더에 파일이 들어갔는지 확인한다.

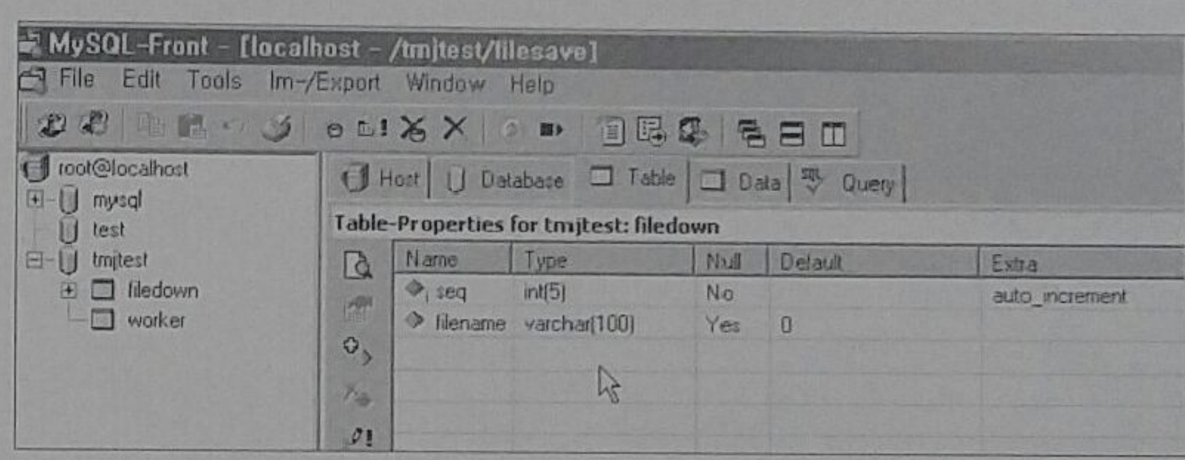


▶ 파일 확인

파일의 업로드가 되었다면 이제 다운로드를 작성해 보자 먼저 download.jsp를 작성하기 전에 업로드 되는 파일의 이름을 데이터베이스에 저장하고 그 이름을 브라우저에 출력하고 그 출력된 파일의 이름을 클릭하였을 때 다운로드되도록 한다.

따라하기 파일다운로드

❶ 먼저 MysqlFornt에서 업로드 파일의 이름을 저장하기 위한 테이블을 만든다.



▶ 테이블 생성

❷ 이 테이블에 업로드 되는 파일의 이름을 저장해야 한다. 위 예제 uploadSave.jsp에 데이터베이스에 저장하는 루틴을 추가한다.

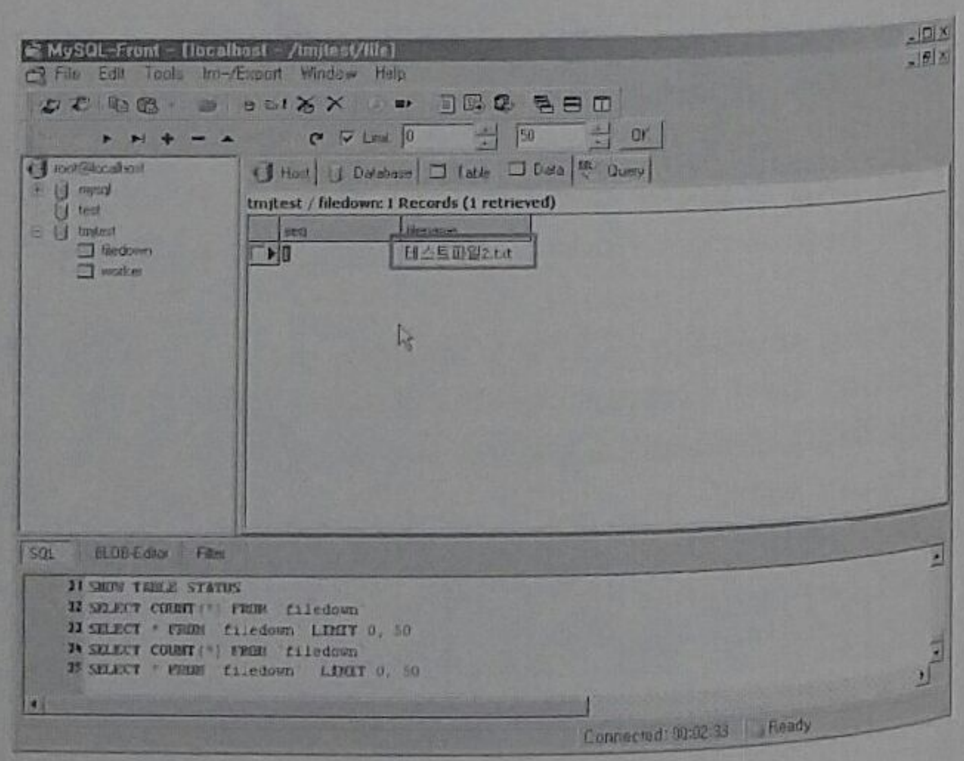
▶ [저장루틴이 추가된 uploadSave.jsp]

```
<%@ page language="java" contentType="text/html; charset=euc-kr" %>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%
String savePath="C:/tmj/upload/";   // 저장할 디렉토리 (절대경로)
String fm = request.getParameter("addfile");
String fileName=" ";
String sql=" ";
//저장결과를 알기 위한 변수 선언
int msg=0;
// 파일업로드 용량 제한.. 10Mb
int sizeLimit = 10 * 1024 * 1024 ;
```

다음 페이지에 계속

```
//만약 같은 이름의 파일이 존재한다면 삭제
 File f = new File ( savePath+fm );
 if (f.exists())
 {
   f.delete();
   }
  else
  {
  MultipartRequest multi =
  new MultipartRequest(request, savePath,sizeLimit,new DefaultFileRenamePolicy());
   fileName= multi.getFilesystemName("addfile");
   }
%>
<%@ include file="../include/try.jsp" %>
<%
     sql="insert into filedown (filename) values( '"+fileName+"' )";
     msg=stmt.executeUpdate(sql);
%>
<%@ include file="../include/catch.jsp" %>
<%
if( msg > 0 )
{
%>
<script language="javascript">
   alert('정상적으로 처리 되었습니다.');
   location.href="./upload.jsp";
</script>
<%
}
else
{
%>
<script language="javascript">
    alert('입력오류 에러입니다.');
    history.go(-1);
</script>
<%
}
%>
```

❸ 브라우저를 열고 파일을 업로드한 후 데이터베이스에 파일의 이름이 제대로 저장이 되어 있는지 확인한다.



▶ 저장확인

저장이 되었다면 이제 업로드된 파일을 다운로드 받을 수 있게 한다. 업로드 된 파일은 단순히 링크만 걸어서 다운로드를 할 수 있다면 쉽겠지만 단순 링크만으로 다운로드 되지 않는다. 따라서 업로드된 파일은 다운로드할 수 있는 프로그램을 필요로 한다.

upload.jsp에 의해 파일을 업로드한 후 업로드 시 저장된 파일의 이름을 불러오는 루틴을 추가한다.

▶ [수정된 upload.jsp]

```
<%@ page language="java" contentType="text/html; charset=euc-kr" %>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%!
 //한글처리
  public static String kor2Db(String sToDb) throws UnsupportedEncodingException{
        if (sToDb == null) return null;
        else return new String(sToDb.getBytes("8859_1"), "KSC5601");
     }
%>
<html>
<head>
<title>파일업로드</title>
<script language="javascript">
        function go_upload()
        {
        var TheForm = document.uploadForm;
        if(TheForm.addfile.value =='')
        {
           alert('업로드 할 파일을 선택해 주세요');
        }
         else
         {
          TheForm.action="./uploadSave.jsp";
          TheForm.submit();
         }
       }
</script>
</head>
<body>
<form name="uploadForm" method="post" enctype="multipart/form-data">
<table border="1" width="400">
    <tr>
        <td width="80">첨부</td>
        <td width="320">
        <input type="file" name="addfile">
         <input type="button" name="up_btn" value="등록"
                                        onClick="javascript:go_upload()">
      </td>
    </tr>
</table>
<%
         //쿼리와 파일이름을 받는 변수선언
        String sql=" ";
         String fileName=" ";
         sql="select filename from filedown";
%>
<%@ include file="../include/try.jsp" %>
```

다음 페이지에 계속 ◎

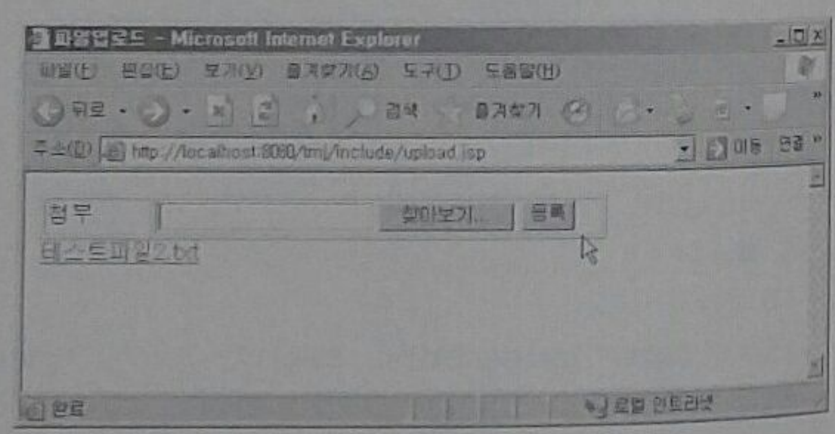
```
<%
    //while문을 통해 저장된 데이터 출력
      rs = stmt.executeQuery(sql);
      while(rs.next())
      {
        fileName=rs.getString("filename");
%>
 <a href="./download.jsp?real_file_nm=<%=kor2Db(fileName)%>
     &save_file_nm=<%=kor2Db(fileName)%>"><%=kor2Db(fileName)%></a>
<%
      }
%>
<%@ include file="../include/catch.jsp" %>
</form>
</body>
</html>
```

여기서 중요부분은 한글처리를 위한 부분이다. 한글이 깨지는 경우 디코딩을 하여 한글이름의 파일명을 처리한다.

```
<%!
  public static String kor2Db(String sToDb) throws UnsupportedEncodingException{
        if (sToDb == null) return null;
        else return new String(sToDb.getBytes("8859_1"), "KSC5601");
    }
%>
```

이런 각종 변환 부분은 부록에 util.java 파일에 나중에 추가하여 사용하게 될 것이다.

브라우저를 열고 upload.jsp를 다시 호출하면 업로드된 파일이 출력된다.



▶ 수정된 upload.jsp

링크된 파일을 다운로드하는 download.jsp를 작성한다. download 루틴들은 웹상에 검색을 통해 많이 제공이 됨으로 굳이 이 책의 download.jsp를 사용하지 않아도 무방하다.

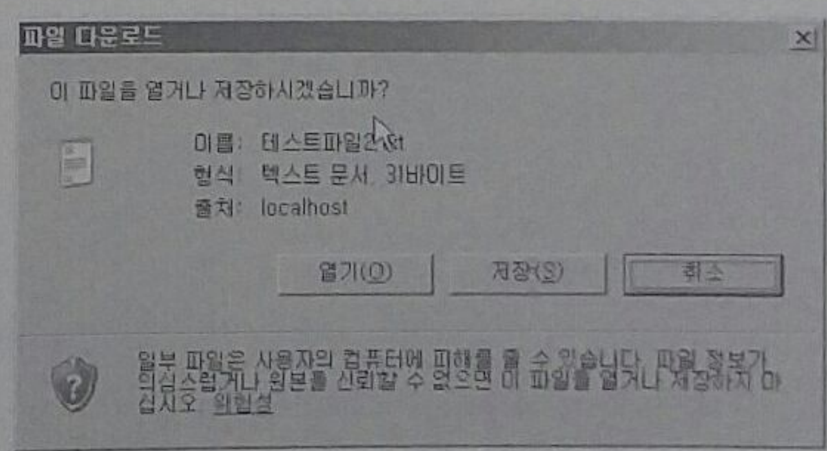
▶ [download.jsp]

```
<%@ page contentType="text/html; charset=MS949"%>
<%@page import="java.io.*,java.util.*,java.net.*" %>
<%
String ref = request.getHeader("REFERER");
//저장된 파일의 실제이름
String real_file_nm = request.getParameter("real_file_nm");
String save_file_nm = request.getParameter("save_file_nm");
//파일이 저장된 경로
```

다음 페이지에 계속

```
String save_dir = "C:/tmj/upload/";
File file = new File(save_dir, save_file_nm);
String strClient = request.getHeader("user-agent");
//파일이 존재하면 헤더와 ContentType을 설정하고
//브라우저 종류에 따라라 Content를 지정한다.
//byte 단위로 BufferedInputStream을 이용하여 읽어 들인다.
//마지막에 if (outs!=null) outs.close();if (fin!=null) fin.close(); 을 반드시 해 줌으로서
//파일을 닫기를 해야 한다.
if (file.exists())
{
if (strClient.indexOf("MSIE 5.5") != -1)
{
response.setContentType("doesn/matter;");
response.setHeader("Content-Disposition","attachment;filename="+real_file_nm+";");
}
else
{
response.setContentType("Content-type:application/octet-stream;");
response.setHeader("Content-Disposition","attachment;filename="+real_file_nm+";");
}
response.setHeader("Content-Transfer-Encoding", "binary;");
response.setHeader ("Content-Length", " "+file.length());
response.setHeader("Pragma", "no-cache;");
response.setHeader("Expires", "-1;");
byte b[] = new byte[1024];
BufferedInputStream fin = new BufferedInputStream(new FileInputStream(file));
BufferedOutputStream outs =
new BufferedOutputStream(response.getOutputStream());
try{
        int read = 0;
        while ((read = fin.read(b)) != -1)
       {
           outs.write(b,0,read);
          }
  }catch (Exception e){}
  finally
  {
    if (outs!=null) outs.close();
    if (fin!=null) fin.close();
  }
}
%>
```

브라우저를 열고 파일을 클릭하면 다운로드 창이 출력된다.



▶ 파일 다운로드

이미지 업로드와 활용예제

우리는 파일을 어떻게 업로드 하는지에 대해서 공부하였다. 이미지 업로드는 파일 업로드와 같은 방법으로 업로드 된다. 여기서의 예제는 앞서 배운 파일 업로드를 통해 이미지 업로드시 이미지 resize의 방법 그리고 이미지 파일만을 업로드할 수 있도록 하는 스크립트의 작성 방법과 실무에서 많이 사용되는 방법이다.

화면 6개의 이미지 파일을 업로드 하여 상세보기를 통해 마우스오버 스크립트를 이용하여 메인 화면에 큰 화면과 서브 화면에 6개를 보여주고 마우스가 이미지를 가리키면 그 이미지가 메인 화면에 보이게 할 것이다.

따라하기 이미지 업로드

❶ 제목과 내용 입력하고 이미지를 업로드 할 imageUpload.jsp을 작성한다.

▶ [imageUpload.jsp]

```
<%@ page language="java" contentType="text/html; charset=euc-kr" %>
<html>
<head>
<title>이미지 업로드</title>
<script language="javascript">
   function go_save()
   {
     var TheForm = document.formm;

   if( TheForm.subject.value==“”)
   {
        alert( '제목을 입력하세요' );
        TheForm.subject.focus();
   }
   else if( TheForm.contexts.value ==“”)
   {
       alert( '내용을 입력하세요' );
       TheForm.contexts.focus();
   }
   else
   {
       TheForm.action="./imageUploadSave.jsp";
       TheForm.submit();
   }
}
</script>
</head>
<body leftmargin="20" topmargin="25">
<form name="formm" method="post" enctype="multipart/form-data">
<table cellpadding="0" cellspacing="0" width="545">
    <tr>
        <td width="545" bgcolor="white" align="center" height=22>
           <font color="blue" size="3"><b>이미지 업로드</b></font>
```

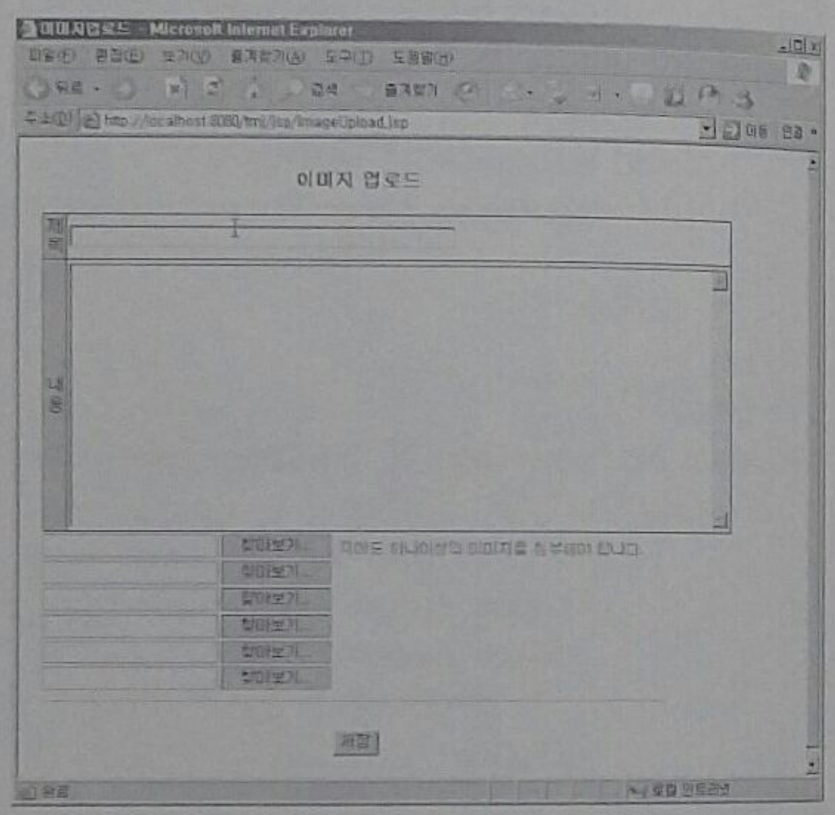
다음 페이지에 계속 ➲

```
        </td>
    </tr>
</table>
<br>
<table border="1" cellspacing="0" width="548" bgcolor="white"
                        bordercolordark="white" bordercolorlight="black">
    <tr>
        <td width="68" bgcolor="#CCCCCC" height=22 align="center">
            <p>제목</p>
        </td>
        <td width="451" bgcolor="white" align="center">
         <p align="left">
          <input type="text" name="subject" size="46" maxlength="100">
       </td>
    </tr>
   <tr>
        <td width="68" bgcolor="#CCCCCC" align="center" height=22>
           내용
        </td>
       <td colspan="2" width="470">
          <textarea name="contents" class="textarea" style="width:580;
                        height:230;" rows="1" cols="20"></textarea>
       </td>
</tr>
</table>
<table cellpadding="0" cellspacing="0" width="548">
    <tr>
        <td width="548" align="left">
          <input type="file" name="add_file1" value="" style="border-style:groove;">
           <font size="2" color="red">
              적어도 하나이상의 이미지를 첨부해야 합니다.
          </font>
       </td>
</tr>
 <tr>
       <td width="548" align="left">
          <input type="file" name="add_file2" value="" style="border-style:groove;">
       </td>
</tr>
      <tr>
       <td width="548" align="left">
          <input type="file" name="add_file3" value="" style="border-style:groove;">
       </td>
</tr>
      <tr>
       <td width="548" align="left">
          <input type="file" name="add_file4" value="" style="border-style:groove;">
       </td>
</tr>
      <tr>
       <td width="548" align="left">
          <input type="file" name="add_file5" value="" style="border-style:groove;">
       </td>
</tr>
      <tr>
       <td width="548" align="left">
          <input type="file" name="add_file6" value="" style="border-style:groove;">
       </td>
```

다음 페이지에 계속

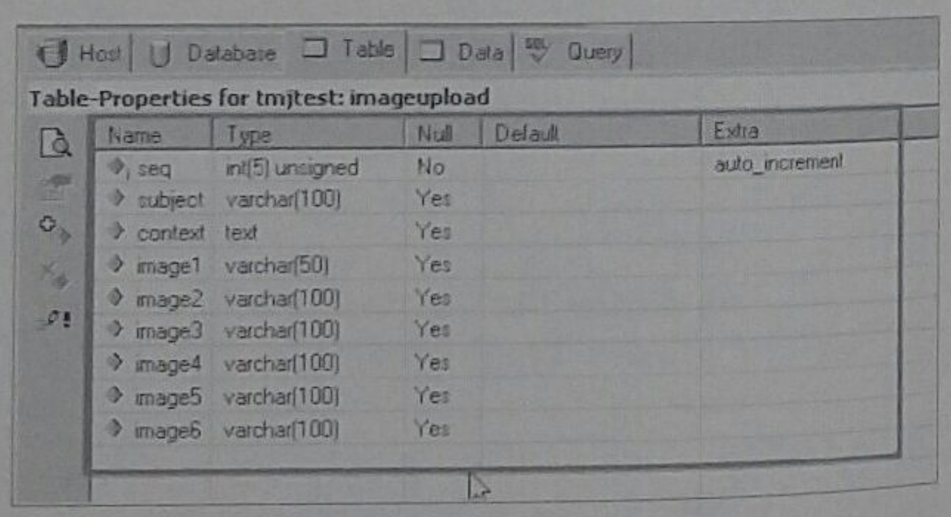
```
    </tr>
</table>
<table cellpadding="0" cellspacing="0" width="542">
    <tr>
        <td width="542" align="center">
            <hr size=1><br>
             <input type="button" name="save" value="저장"
                                        onClick="javascript:go_save()">
        </td>
    </tr>
</table>
</form>
</body>
</html>
```

❷ 브라우저를 열고 imageUpload.jsp 파일을 호출한다.



▶ imageUpload.jsp의 호출

❸ 제목과 내용 그리고 업로드 되는 이미지의 이름이 저장되는 테이블을 생성한다.



Name	Type	Null	Default	Extra
seq	int(5) unsigned	No		auto_increment
subject	varchar(100)	Yes		
context	text	Yes		
image1	varchar(50)	Yes		
image2	varchar(100)	Yes		
image3	varchar(100)	Yes		
image4	varchar(100)	Yes		
image5	varchar(100)	Yes		
image6	varchar(100)	Yes		

▶ imageupload 테이블 생성

❹ 제목과 내용, 업로드 되는 이미지 파일의 이름을 저장하는 imageUploadSave.jsp를 작성한다.

▶ [imageUploadSave.jsp]

```
<%@page contentType="text/html;charset=euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
```

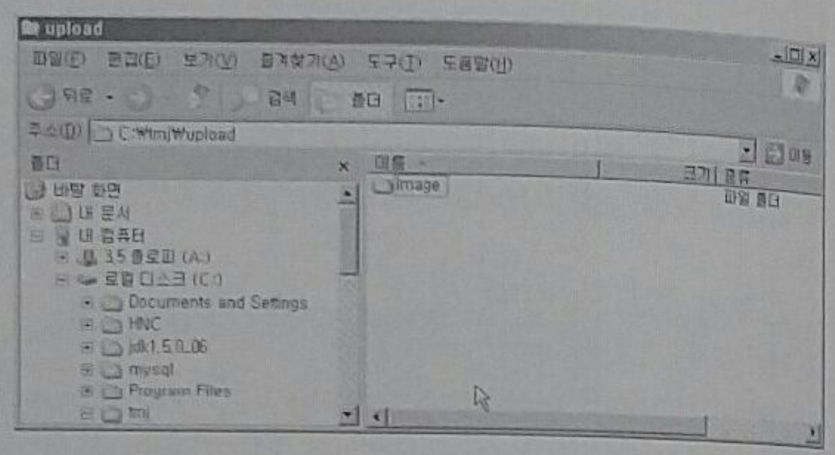
다음 페이지에 계속 ❶

```
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%!
 //null 체크를 위한 function으로 nullpoint 에러를 잡는다.
 public static String nullcheck(String str) throws Exception {
        if (str == null){
            return "";
        }else{
            return str;
              }
      }
%>
<%
//이미지의 저장경로
String savePath="C:/tmj/upload/image";
int sizeLimit=1024*1024*5; //용량제한 5메가

//파일 업로드 객체 생성
MultipartRequest multi =
new MultipartRequest(request,savePath,sizeLimit,new DefaultFileRenamePolicy());

String subject=nullcheck(multi.getParameter("subject"));
String contexts=nullcheck(multi.getParameter("contexts"));
String fileName1 = nullcheck(multi.getFilesystemName("add_file1"));
String fileName2 = nullcheck(multi.getFilesystemName("add_file2"));
String fileName3 = nullcheck(multi.getFilesystemName("add_file3"));
String fileName4 = nullcheck(multi.getFilesystemName("add_file4"));
String fileName5 = nullcheck(multi.getFilesystemName("add_file5"));
String fileName6 = nullcheck(multi.getFilesystemName("add_file6"));

int message=0;
//제목, 내용, 업로드 되는 이미지 파일의 이름을 저장하는 쿼리
String sql="";

sql="insert into imageupload ";
sql+="(subject,context,image1,image2,image3,image4,image5,image6 ) ";
sql+="value( '"+subject+"' ,'"+contexts+"' ,'"+fileName1+"' , ";
sql+="'"+fileName2+"' ,'"+fileName3+"' ,'"+fileName4+"' , ";
sql+="'"+fileName5+"' ,'"+fileName6+"' )";
%>
<%@ include file="../include/try.jsp" %>
<%
  message=stmt.executeUpdate(sql);
   stmt.close();
%>
<%@ include file="../include/catch.jsp" %>
<%
if( message > 0 )
{
%>
<script language="javascript">
   alert('정상적으로 처리 되었습니다.');
   location.href="./imageUpload.jsp";
</script>
<%
}
else
{
```

다음 페이지에 계속

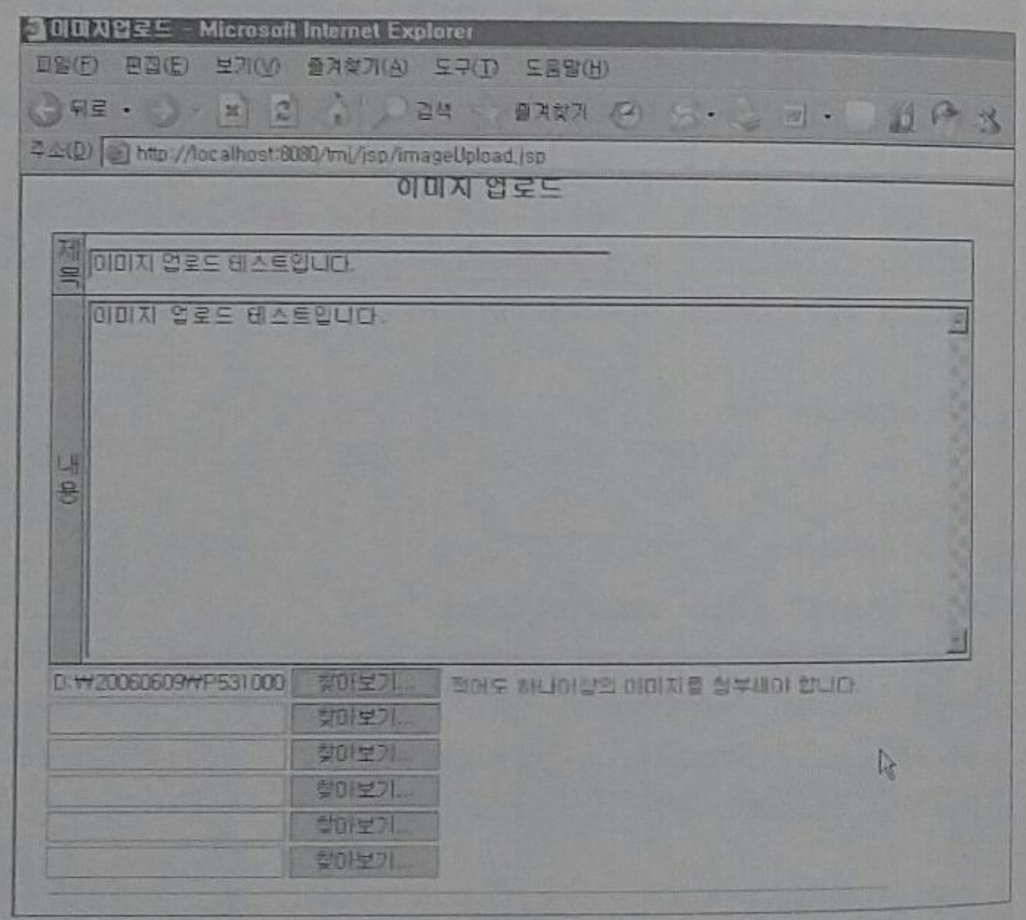
```
%>
   <script language="javascript">
    alert('입력오류 에러입니다.');
    history.go(-1);
</script>
<%
}
%>
```

❺ c:/tmj/upload/image 폴더를 생성한다.



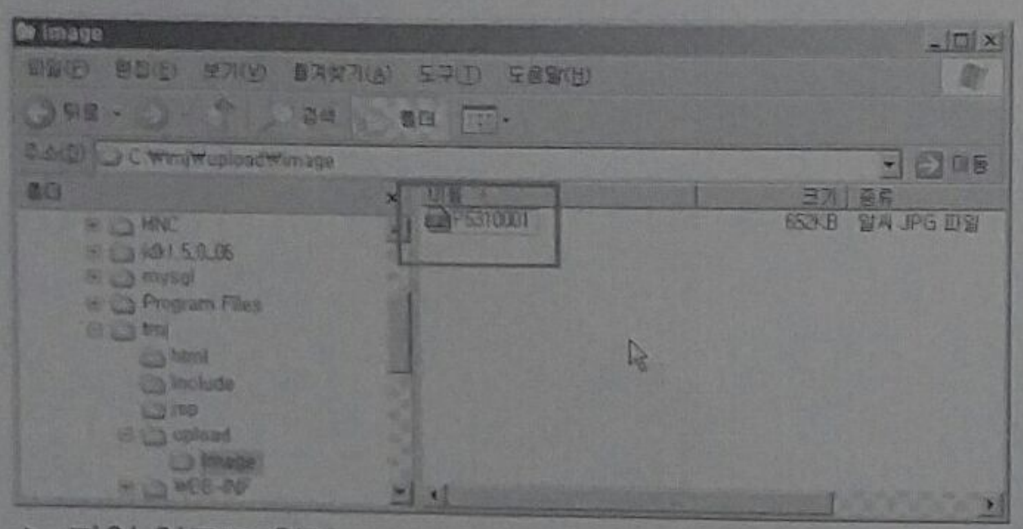
▶ 저장 폴더 생성

❻ 브라우저를 열고 imageUpload.jsp 호출하여 제목과 내용을 입력하고 업로드될 이미지를 찾아보기를 통해 입력하고 저장버튼을 클릭한다.

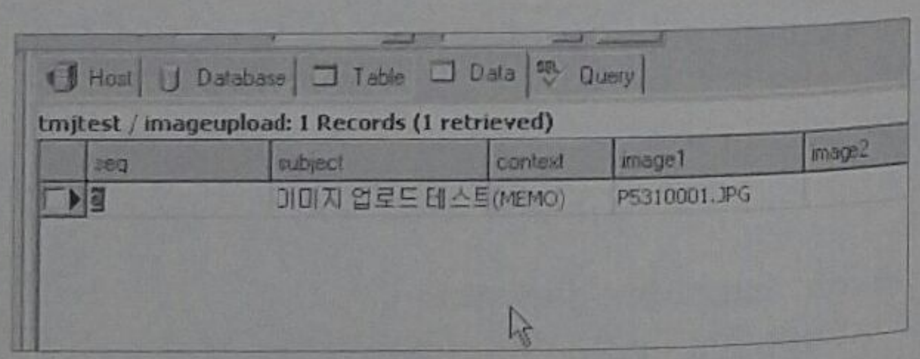


▶ imageUpload.jsp 입력값

❼ 데이터베이스와 upload/image 폴더를 확인한다.



▶ 파일 업로드 확인



▶ 데이터베이스 저장확인

❽ 저장된 데이터의 리스트 화면을 작성한다.

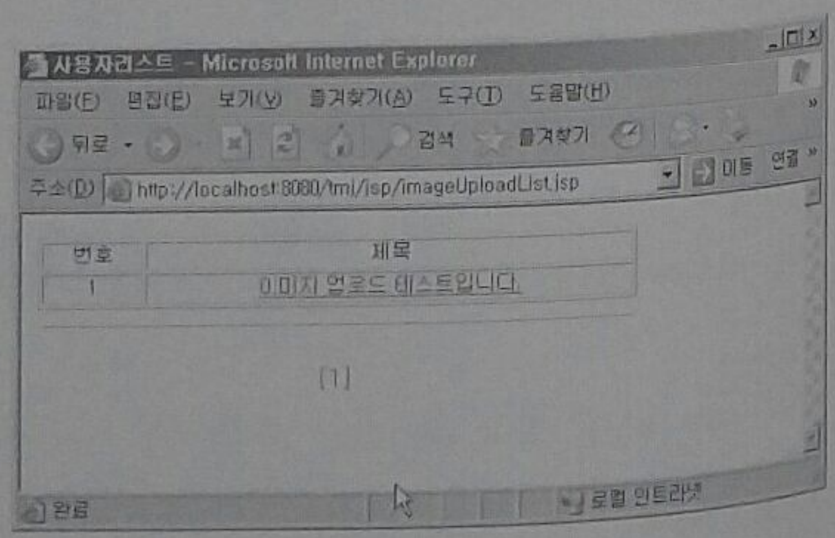
▶ [imageUploadList.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%
String listSql="select * from imageupload";
String pageCountQuery = "select count(1) from imageupload ";

// 페이지 설정을 위한 변수
int view_rows =10;
int total_pages =1;            //전체 페이지 (default 1)
int tpage=1;                   //현재 페이지 (default 1)

String url = "";
url = "./imageUploadList.jsp?tpage=";

if (!"ok".equals(request.getParameter("fullview")) &&
                    request.getParameter("tpage") != null) //전체보기확인
tpage = Integer.parseInt(request.getParameter("tpage"));

if (!"ok".equals(request.getParameter("fullview")))
listSql += " limit "+((tpage-1)*view_rows)+","+view_rows;

%>
<%@ include file="../include/try.jsp" %>
<%
   rs = stmt.executeQuery(listSql);
%>
<html>
<head>
<title>사용자리스트</title>
<script language="javascript">
function go_view(seq)
{
var TheForm=document.imageForm;
//레코드의 고유증가값 seq를 파라메타로 넘겨준다.
TheForm.action="./imageView.jsp?seq="+seq;
TheForm.submit();
}
</script>
</head>
<body >
<form name="imageForm" method="post">
<table border="1" width="383">
<tr>
<td width="60">
<p align="center"><font size="2">번호</font></p>
</td>
<td>
<p align="center"><font size="2">제목</font></p>
</td>
</tr>
<!--while문을 통해 리스트를 출력한다.-->
<%
int num=1;
```

다음 페이지에 계속 ➲

```
if (!"ok".equals(request.getParameter("fullview")))
num = ((tpage-1)*view_rows)+1;

while(rs.next())
{
%>
<tr>
<td width="60">
   <p align="center"><font size="2"><%=num%></font></p>
</td>
<td>
<p align="center"><font size="2">
<!-- seq: 데이터의 PK(키값) 값으로 파라메타값으로 전달한다.-->
<a href="#" onClick="go_view('<%=rs.getString("seq")%>')">
   <%=rs.getString("subject")%>
</a>
</font></p>
</td>
</tr>
<%
//번호에 대한 증가 값
num++;
}
if(num ==1)
{
%>
<tr>
   <td colspan=4 align=center>해당하는 데이터가 없습니다.</td>
</tr>
<%
}
%>
</table>
<table border="0" width="383">
<tr>
  <td align="center">
    <%@ include file="../include/pageCount.jsp"%>
  </td>
</tr>
</table>
</form>
</body>
</html>
<%@ include file="../include/catch.jsp" %>
```

❽ 브라우저를 열고 imageUploadList.jsp를 호출한다.



▶ imageUploadList.jsp

❾ 출력된 리스트에서 제목을 클릭하면 해당 seq를 파마메타로 전달받아 해당 데이터의 상세를 출력하는 imageView.jsp를 작성한다.

▶ [imageView.jsp]

```
<%@page contentType="text/html;charset=euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%!
//null 처리를 위한 function
public static String nullcheck(String str) throws Exception {
   if (str == null)
   {
        return "";
   }
   else
   {
      return str;
   }
}
%
<%!
//한글 처리를 위한 function
  public static String kor2Db(String sToDb) throws UnsupportedEncodingException{
        if (sToDb == null) return null;
        else return new String(sToDb.getBytes("8859_1"), "KSC5601");
     }
%>
<%
//파라메타를 받는 변수선언
String vseq= nullcheck(request.getParameter("seq"));
String view_sql="";
//전달된 파라메타 seq를 가지고 쿼리를 생성한다.
view_sql="select * from imageupload where seq='"+vseq+"'";

String r_seqn="";
String r_context="";
String r_subject="";
String r_file1="";
String r_file2="";
String r_file3="";
String r_file4="";
String r_file5="";
String r_file6="";
String r_file_path="c:/tmj/upload/image";
%>
<%@ include file="../include/try.jsp" %>
<%
rs=stmt.executeQuery(view_sql);
if(rs.next())
{
r_subject=nullcheck(rs.getString("subject"));
r_context=nullcheck(rs.getString("context"));
r_seqn=nullcheck(rs.getString("seq"));
r_file1=nullcheck(rs.getString("image1"));
r_file2=nullcheck(rs.getString("image2"));
r_file3=nullcheck(rs.getString("image3"));
r_file4=nullcheck(rs.getString("image4"));
```

다음 페이지에 계속 ➲

```
r_file5=nullcheck(rs.getString("image5"));
r_file6=nullcheck(rs.getString("image6"));
}
rs.close();
%>
<%@ include file="../include/catch.jsp" %>
<html>
<head>
<title>이미지보기</title>
<script language="javascript">

function go_move()
{
 location.href="./imageUploadList.jsp";
}

//마우스 오버에 따른 script function
function changeLayerMain(num)
{
 if(num == "1")
 {
  document.getElementById("f1").style.display = "block";
  document.getElementById("f2").style.display = "none";
  document.getElementById("f3").style.display = "none";
  document.getElementById("f4").style.display = "none";
  document.getElementById("f5").style.display = "none";
  document.getElementById("f6").style.display = "none";
 }
 else if(num == "2")
 {
  document.getElementById("f1").style.display = "none";
  document.getElementById("f2").style.display = "block";
  document.getElementById("f3").style.display = "none";
  document.getElementById("f4").style.display = "none";
  document.getElementById("f5").style.display = "none";
  document.getElementById("f6").style.display = "none";
 }
 else if(num == "3")
 {
  document.getElementById("f1").style.display = "none";
  document.getElementById("f2").style.display = "none";
  document.getElementById("f3").style.display = "block";
  document.getElementById("f4").style.display = "none";
  document.getElementById("f5").style.display = "none";
  document.getElementById("f6").style.display = "none";
 }
 else if(num == "4")
 {
  document.getElementById("f1").style.display = "none";
  document.getElementById("f2").style.display = "none";
  document.getElementById("f3").style.display = "none";
  document.getElementById("f4").style.display = "block";
  document.getElementById("f5").style.display = "none";
  document.getElementById("f6").style.display = "none";
 }
 else if(num == "5")
 {
```

다음 페이지에 계속

```
 document.getElementById("f1").style.display = "none";
 document.getElementById("f2").style.display = "none";
 document.getElementById("f3").style.display = "none";
 document.getElementById("f4").style.display = "none";
 document.getElementById("f5").style.display = "block";
 document.getElementById("f6").style.display = "none";
}
else if(num == "6")
{
 document.getElementById("f1").style.display = "none";
  document.getElementById("f2").style.display = "none";
  document.getElementById("f3").style.display = "none";
  document.getElementById("f4").style.display = "none";
  document.getElementById("f5").style.display = "none";
  document.getElementById("f6").style.display = "block";
 }
}
</script>
</head>
<body leftmargin="20" topmargin="25">
<form name="formm" method="post">
<table cellpadding="0" cellspacing="0" width="670">
<tr>
 <td width="670" bgcolor="white" align="center" height=22>
    <font color="blue" size="3"><b>이미지보기</b></font>
 </td>
</tr>
</table>
<br>
<table border="1" cellspacing="0" width="670" bgcolor="white"
                                 bordercolordark="white" bordercolorlight="black">
<tr >
   <td width="67" bgcolor="#CCCCCC" height=22 align="center">
      <p>제목</p>
    </td>
<td width="451">
<%=kor2Db(r_subject) %>
</tr>
<tr>
  <table cellpadding="0" cellspacing="0" width="529">
   <tr>
     <td width="320" rowspan="3">
<%
      if(!"".equals(r_file1))
      {
%>
     <img src="<%=r_file_path%>/<%=r_file1%>" width="303" height="204"
                 border="0" id="f1"style="padding-top:2px;display:block;">
     <img src="<%=r_file_path%>/<%=r_file2%>" width="303" height="204"
                 border="0" id="f2"style="padding-top:2px;display:none;">
     <img src="<%=r_file_path%>/<%=r_file3%>" width="303" height="204"
                 border="0" id="f3"style="padding-top:2px;display:none;">
     <img src="<%=r_file_path%>/<%=r_file4%>" width="303" height="204"
                 border="0" id="f4"style="padding-top:2px;display:none;">
     <img src="<%=r_file_path%>/<%=r_file5%>" width="303" height="204"
                 border="0" id="f5"style="padding-top:2px;display:none;">
     <img src="<%=r_file_path%>/<%=r_file6%>" width="303" height="204"
                 border="0" id="f6"style="padding-top:2px;display:none;">
```

다음 페이지에 계속

```
<%
}
else
{
%>
이미지 없음.
<%
}
%>
</td>
<td width="121">
<%
if(!"".equals(r_file1))
{
%>
<a href="#" onMouseOver="javascript:changeLayerMain('1');" >
    <img src="<%=r_file_path%>/<%=java.net.URLEncoder.encode(r_file1)%>"
                                        width="85" height="54" border="0">
</a>
<%
}
else
{
%>
이미지 없음.
<%
}
%>
</td>
<td width="115">
<%
if(!"".equals(r_file2))
{
%>
<a href="#" onMouseOver="javascript:changeLayerMain('2');" >
  <img src="<%=r_file_path%>/<%=java.net.URLEncoder.encode(r_file2)%>"
                width="85" height="54" border="0">
</a>
<%
}
else
{
%>
이미지 없음.
<%
}
%>
</td>
</tr>
<tr>
<td width="121">
<%
if(!"".equals(r_file3))
{
%>
<a href="#" onMouseOver="javascript:changeLayerMain('3');" >
    <img src="<%=r_file_path%>/<%=java.net.URLEncoder.encode(r_file3)%>"
                                        width="85" height="54" border="0">
```

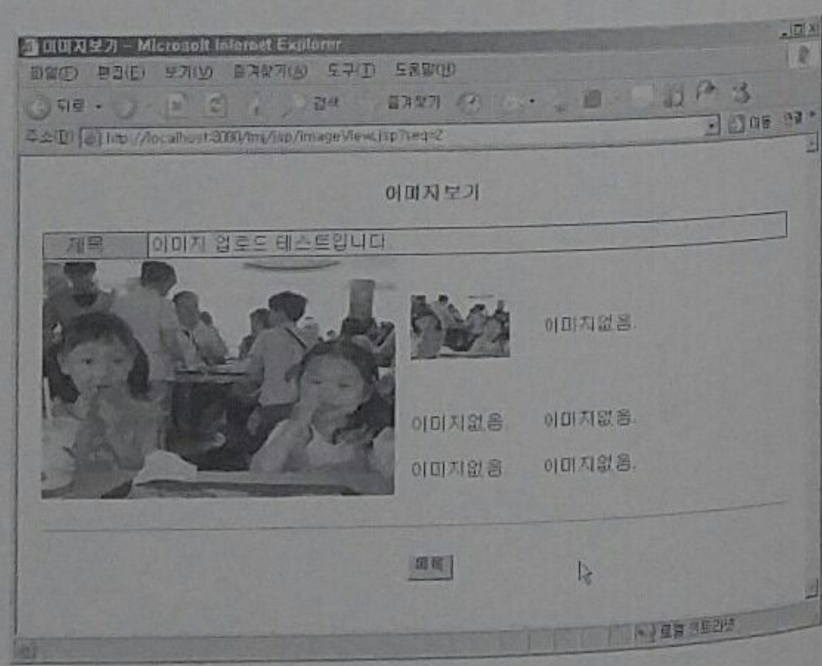
다음 페이지에 계속

```
</a>
<%
}
else
{
%>
이미지 없음.
<%
}
%>
</td>
<td width="115">
<%
if(!"".equals(r_file4))
{
%>
<a href="#" onMouseOver="javascript:changeLayerMain('4');" >
    <img src="<%=r_file_path%>/<%=java.net.URLEncoder.encode(r_file4)%>"
                                        width="85" height="54" border="0">
</a>
<%
}
else
{
%>
이미지 없음.
<%
}
%>
</td>
</tr>
<tr>
<td width="121">
<%
if(!"".equals(r_file5))
{
%>
<a href="#" onMouseOver="javascript:changeLayerMain('5');" >
    <img src="<%=r_file_path%>/<%=java.net.URLEncoder.encode(r_file5)%>"
                                        width="85" height="54" border="0">
</a>
<%
}
else
{
%>
이미지 없음.
<%
}
%>
</td>
<td width="115">
<%
if(!"".equals(r_file6))
{
%>
<a href="#" onMouseOver="javascript:changeLayerMain('6');" >
   <img src="<%=r_file_path%>/<%=java.net.URLEncoder.encode(r_file6)%>"
                                        width="85" height="54" border="0">
```

다음 페이지에 계속

```
</a>
<%
}
else
{
%>
이미지 없음.
<%
}
%>
</td>
</tr>
</table>
<br>
</tr>
<tr>
<table width="670" border="0" cellspacing="0" cellpadding="0">
<tr>
  <td valign="top">
   <div align="justify">
    <%=kor2Db(r_context)%>
   </div>
  </td>
</tr>
</table>
<table cellpadding="0" cellspacing="0" width="670">
<tr>
<td width="670" align="center">
<hr size=1><br>
<input type="button" name="save" value="목록" onClick="javascript:go_move()">
</td>
</tr>
</table>
</form>
</body>
</html>
```

⑩ 브라우저를 열고 imageUploadList.jsp를 호출, 리스트의 제목을 클릭하면 imageView.jsp가 화면에 출력이 된다.



▶ imageView.jsp

⓫ 목록으로 이동하면 다시 리스트가 보인다. 하지만 새로운 데이터 입력의 이미지 업로드하는 imageUpload.jsp로 이동 버튼이 없다. imageUploadList.jsp에 imageUpload.jsp로 이동되는 버튼과 스크립트를 추가한다.

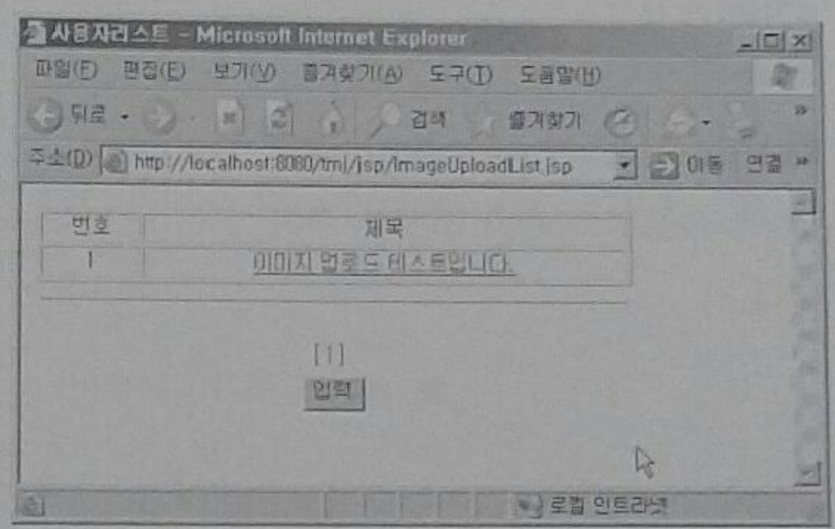
```
</table>
<table border="0" width="383">
<tr><td align="center">
<input type="button" name="move" value="입력" onClick="go_move()">
</table>
</form>
</body>
```

▶ 버튼추가

```
//입력으로 이동되는 function
function go_move()
{
    var TheForm=document.imageForm;

    TheForm.action="./imageUpload.jsp";
    TheForm.submit();

}
```

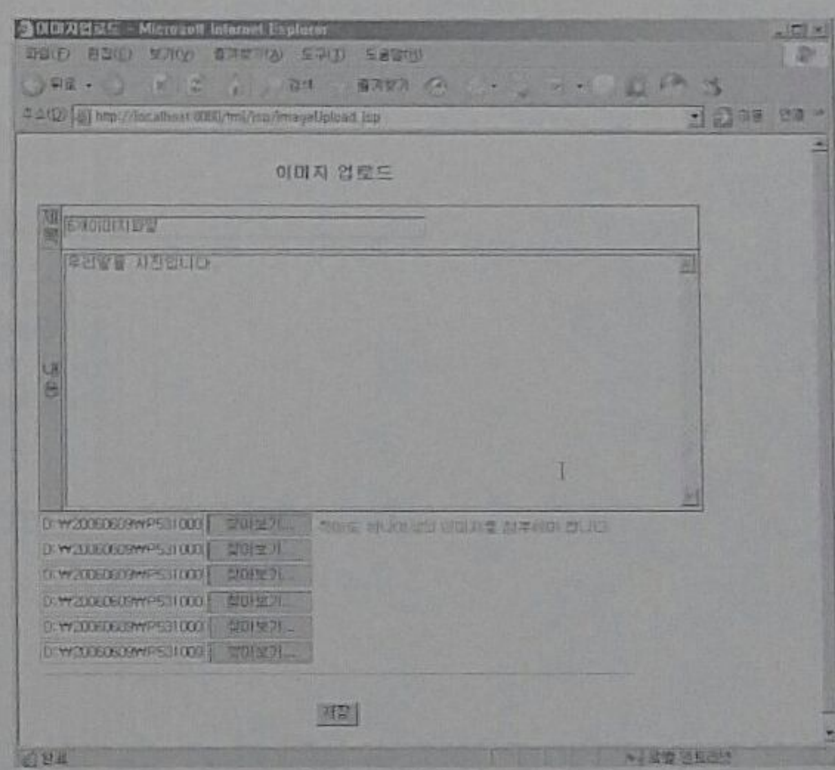
▶ javascript추가

⓬ 브라우저에서 imageUploadList.jsp를 호출하면 입력버튼이 생성이 되었을 것이다.



▶ 입력 버튼의 생성

⓭ 입력화면으로 이동하여 데이터를 입력하고 6개의 이미지파일을 모두 선택 후 저장버튼을 클릭한다.



▶ 데이터 입력

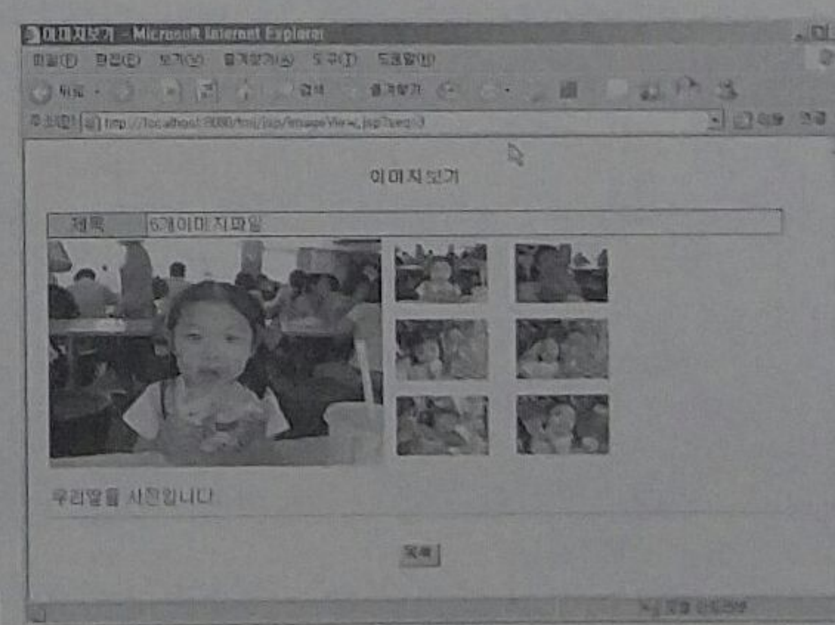
저장 후에 바로 리스트로 이동하려면 imageUpload.jsp 파일에서 저장 성공 시에 이동되는 스크립트에 imageUploadList. jsp를 주면 된다.

```
<script language="javascript">
   alert('정상적으로 처리 되었습니다.');
   //location.href="./imageUpload.jsp";
   location.href="./imageUploadList.jsp";
</script>
<%
```

▶ imageUpload.jspdml 수정

⓮ 브라우저에서 imageUploadList를 호출하고 6개의 이미지가 올려진 리스트의 제목을 클릭하면 해당 데이터의 상세가 출력된다.

⓯ 마우스를 이용하여 작은 이미지 가까이 가면 마우스오버 스크립트에 의해 작은 이미지가 화면의 왼쪽 큰 화면에 보이게 된다.



▶ 이미지 보기

이미지파일 업로드 예제는 앞 장에서 사용되었던 pageCount를 그대로 사용하였다. pageCount.jsp는 앞으로 include되어 어느 jsp이건 간에 형식만 맞춰주면 사용한다.

이미지 업로드 예제를 보면 2가지 부분이 빠진 것을 느낄 수 있다. 하나는 수정이며 하나는 삭제 부분이다. 수정과 삭제 부분은 쉽게 해결이 가능하다. 왜냐하면 이미 저장되는 부분이 있음으로 그 부분을 약간의 손질을 통해 다듬기만 하면 되며 삭제는 imageDelete.jsp를 만들어 사용하면 된다. 이미지 예제를 마저 완성해 보도록 하자.

따라하기 이미지 수정 및 삭제

❶ 이미 작성되어진 imageView.jsp에 수정과 삭제의 버튼을 추가한다.

```
<td width="670" align="center">
<hr size=1><br>
<input type="button" name="save" value="목록" onClick="javascript:go_move()">
<input type="button" name="save" value="수정" onClick="javascript:go_mod('<%=vseq%>')">
<input type="button" name="save" value="삭제" onClick="javascript:go_del('<%=vseq%>')">
</td>
</tr>
</table>
```

▶ 버튼의 추가

❷ 삭제 스크립트를 작성한다.

```
function go_move()
{
location.href="./imageUploadList.jsp";

}
function go_mod(seq)
{
location.href="./imageUploadmod.jsp?seq="+seq;

}
function go_del(seq)
{
location.href="./imageUploadDelete.jsp?seq="+seq;

}
```

▶ script의 추가

❸ imageUploadmod.jsp를 작성한다. 이미 imageUpload.jsp 파일을 만들었기 때문에 이를 다른 이름으로 저장하고 이미지 업로드를 하는 부분에 이미지를 보여주고 해당 이미지의 이름과 seq를 script에 파라메타 값으로 전달할 수 있도록 한다.

▶ [imageUploadmod.jsp]

```
<%@ page language="java" contentType="text/html; charset=euc-kr" %>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%!
//null 체크를 위한 function으로 nullpoint 에러를 잡는다.
public static String nullcheck(String str) throws Exception {
 if (str == null)
 {
    return "";
  }
  else
```

다음 페이지에 계속 ➊

```
 {
   return str;
 }
}
%>
<%
String vseq= nullcheck(request.getParameter("seq"));

String subject="";
String contexts="";
String file1="";
String file2="";
String file3="";
String file4="";
String file5="";
String file6="";
String path="c:/tmj/upload/image/";

String sql="select * from imageupload where seq='"+vseq+"'";
%>
<%@ include file="../include/try.jsp" %>
<%
 rs = stmt.executeQuery(sql);
 if(rs.next())
 {
   subject = nullcheck(kor2Db(rs.getString("subject")));
   contexts =nullcheck(kor2Db(rs.getString("context")));
   file1 =nullcheck(rs.getString("image1"));
   file2 =nullcheck(rs.getString("image2"));
   file3 =nullcheck(rs.getString("image3"));
   file4 =nullcheck(rs.getString("image4"));
   file5 =nullcheck(rs.getString("image5"));
   file6 =nullcheck(rs.getString("image6"));
 }
%>
<%@ include file="../include/catch.jsp" %>
<html>
<head>
<title>이미지업로드</title>
<script language="javascript">

function go_move()
{
 location.href="./imageUploadList.jsp";
}

//이미지 수정으로 이동
function go_modSave(num,seq)
{
 var TheForm = document.formm;
 TheForm.encoding = "multipart/form-data";
 TheForm.action="./imageUploadmodSave.jsp?seq="+seq+"&num="+num;
 TheForm.submit();
}
//제목과 내용을 수정으로 이동
function go_mod(seq)
```

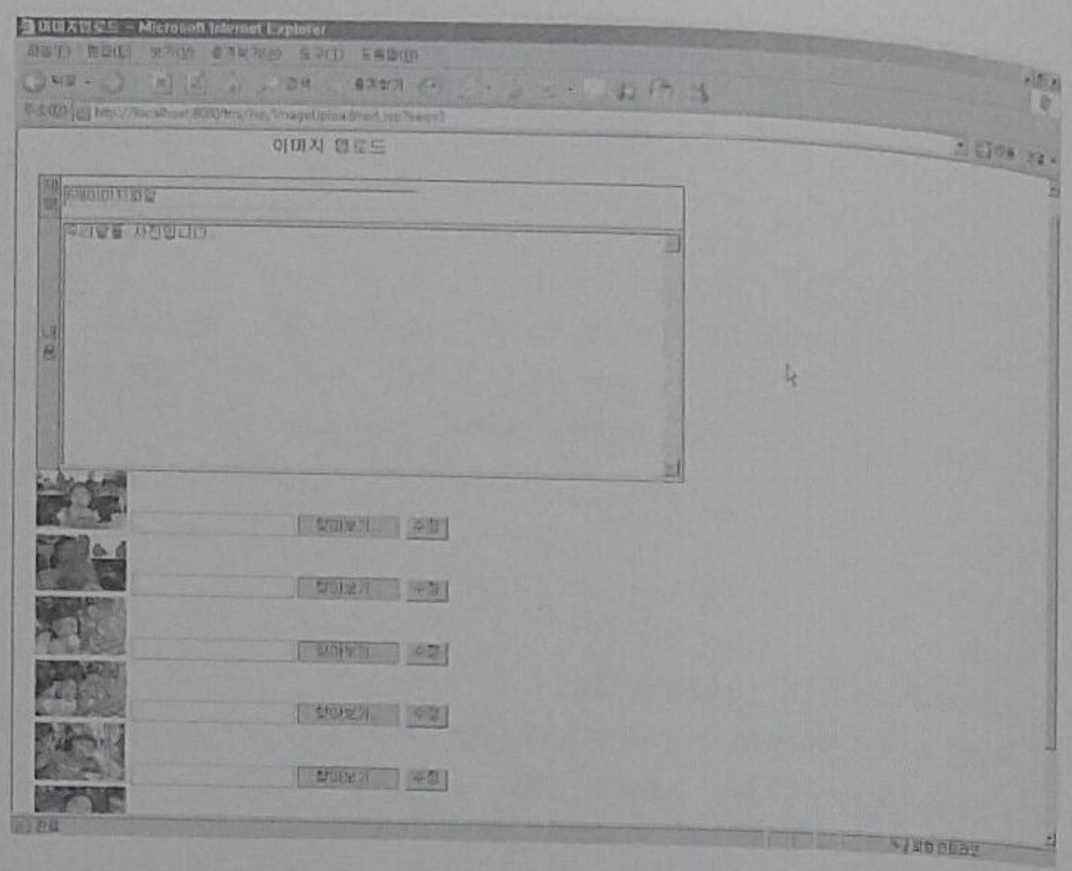
다음 페이지에 계속

```
{
 var TheForm = document.formm;
 if(TheForm.subject.value =="")
 {
   alert('제목을 입력하세요');
 }
 else if(TheForm.contexts.value =="")
 {
   alert('내용을 입력하세요');
 }
 else
 {
   TheForm.action="./imageUploadmodSave2.jsp?seq="+seq;
   TheForm.submit();
  }
}
</script>
</head>
<body leftmargin="20" topmargin="25">
<form name="formm" method="post" enctype="multipart/form-data">
<table cellpadding="0" cellspacing="0" width="545">
<tr>
<td width="545" bgcolor="white" align="center" height=22>
<font color="blue" size="3"><b>이미지 업로드</b></font>
</td>
</tr>
</table>
<br>
<table border="1" cellspacing="0" width="548" bgcolor="white"
                          bordercolordark="white" bordercolorlight="black">
<tr>
<td width="68" bgcolor="#CCCCCC" height=22 align="center">
<p>제목</p>
</td>
<td width="451" bgcolor="white" align="center">
<p align="left">
  <input type="text" name="subject" size="46" maxlength="100"
                                          value="<%=kor2Db(subject)%>">
</td>
</tr>
<tr>
<td width="68" bgcolor="#CCCCCC" align="center" height=22>
내용
</td>
<td colspan="2" width="470">
<textarea name="contexts" class="textarea" style="width:580;height:230;"
              rows="1" cols="20"><%=kor2Db(contexts)%></textarea>
</td>
</tr>
</table>
<table cellpadding="0" cellspacing="0" width="548">
<tr>
<td width="548" align="left">
<img src="<%=path%><%=file1%>" width="85" height="54" border="0">
<input type="file" name="add_file1" value="" style="border-style:groove;">
<input type="button" name="mod1" value="수정"
                                  onClick="go_modSave(1,'<%=vseq%>')">
```

다음 페이지에 계속

```
</td>
</tr>

<tr>
<td width="548" align="left">
<img src="<%=path%><%=file2%>" width="85" height="54" border="0">
<input type="file" name="add_file2" value="" style="border-style:groove;">
<input type="button" name="mod2" value="수정"
                                onClick="go_modSave(2,'<%=vseq%>')">
</td>
</tr>
<tr>
<td width="548" align="left">
 <img src="<%=path%><%=file3%>" width="85" height="54" border="0">
<input type="file" name="add_file3" value="" style="border-style:groove;">
<input type="button" name="mod3" value="수정"
                                onClick="go_modSave(3,'<%=vseq%>')">
</td>
</tr>
<tr>
<td width="548" align="left">
<img src="<%=path%><%=file4%>" width="85" height="54" border="0">
<input type="file" name="add_file4" value="" style="border-style:groove;">
<input type="button" name="mod4" value="수정"
                                onClick="go_modSave(4,'<%=vseq%>')">
</td>
</tr>
<tr>
<td width="548" align="left">
<img src="<%=path%><%=file5%>" width="85" height="54" border="0">
<input type="file" name="add_file5" value="" style="border-style:groove;">
<input type="button" name="mod5" value="수정"
                                onClick="go_modSave(5,'<%=vseq%>')">
</td>
</tr>
<tr>
<td width="548" align="left">
 <img src="<%=path%><%=file6%>" width="85" height="54" border="0">
<input type="file" name="add_file6" value="" style="border-style:groove;">
<input type="button" name="mod6" value="수정"
                                onClick="go_modSave(6,'<%=vseq%>')">
</td>
</tr>
</table>
<table cellpadding="0" cellspacing="0" width="542">
<tr>
<td width="542" align="center">
<hr size=1>
<input type="button" name="save" value="목록" onClick="javascript:go_move()">
<input type="button" name="save" value="수정"
                                onClick="javascript:go_mod('<%=vseq%>')">
</td>
</tr>
</table>
</form>
</body>
</html>
```

❹ 브라우저에서 imageUploadList.jsp → imageView.jsp를 호출, 수정버튼을 클릭한다.



▶ 수정화면

❺ 수정값을 전달받아 저장하는 imageUploadmodSave.jsp를 작성한다.

imageUploadmodSave는 imageUploadmod에서 전달받은 num과 seq를 가지고 데이터베이스에 수정되는 이미지의 이름을 update하고 이미지파일을 업로드한다.

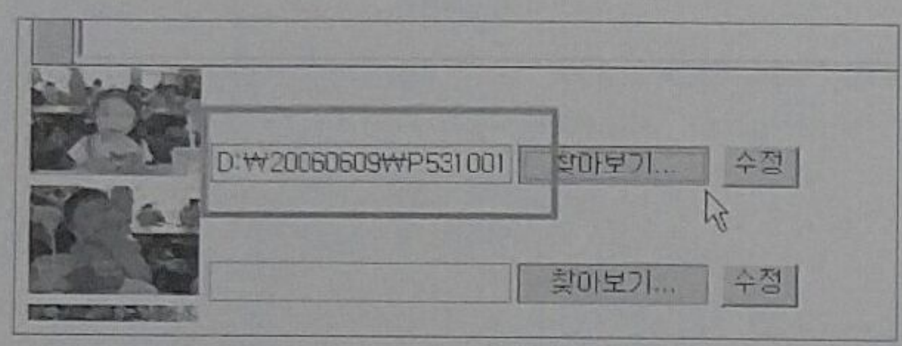
▶ [imageUploadmodSave.jsp]

```
<%@page contentType="text/html;charset=euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%!
 //null 체크를 위한 function으로 nullpoint 에러를 잡는다.
 public static String nullcheck(String str) throws Exception {

        if (str == null){
            return "";
        }else{
                    return str;
                }
    }
%>
<%
String savePath="C:/tmj/upload/image";
String vseq= nullcheck(request.getParameter("seq"));
String num= nullcheck(request.getParameter("num"));
int sizeLimit=1024*1024*5;
int message=0;

String sql=" ";
MultipartRequest multi =
new MultipartRequest(request,savePath,sizeLimit,new DefaultFileRenamePolicy());

if("1".equals(num))
{
      String fileName1 = nullcheck(multi.getFilesystemName("add_file1"));
      sql="update imageupload set image1='"+fileName1+"' where seq='"+vseq+"'";
}
```

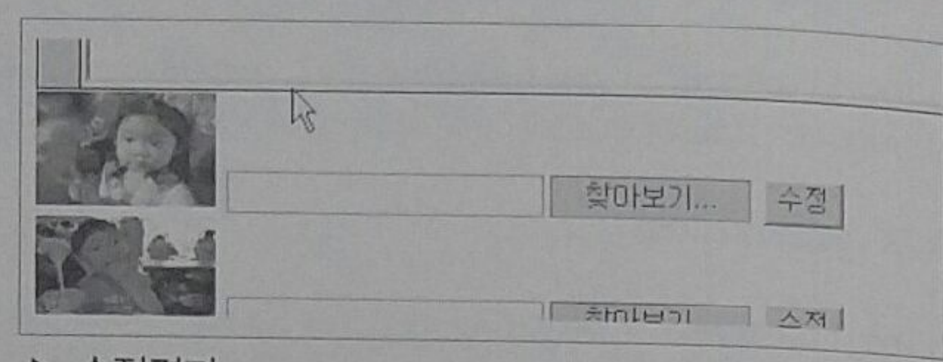
다음 페이지에 계속 ➲

```
else if("2".equals(num))
{
    String fileName2 = nullcheck(multi.getFilesystemName("add_file2"));
    sql="update imageupload set image2='"+fileName2+"' where seq='"+vseq+"'";
}
else if("3".equals(num))
{
    String fileName3 = nullcheck(multi.getFilesystemName("add_file3"));
    sql="update imageupload set image3='"+fileName3+"' where seq='"+vseq+"'";
}
else if("4".equals(num))
{
    String fileName4 = nullcheck(multi.getFilesystemName("add_file4"));
    sql="update imageupload set image4='"+fileName4+"' where seq='"+vseq+"'";
}
else if("5".equals(num))
{
    String fileName5 = nullcheck(multi.getFilesystemName("add_file5"));
    sql="update imageupload set image5='"+fileName5+"' where seq='"+vseq+"'";
}
else if("6".equals(num))
{
    String fileName6 = nullcheck(multi.getFilesystemName("add_file6"));
    sql="update imageupload set image6='"+fileName6+"' where seq='"+vseq+"'";
}


%>
<%@ include file="../include/try.jsp" %>
<%
      message=stmt.executeUpdate(sql);
      stmt.close();
%>
<%@ include file="../include/catch.jsp" %>
<%
if( message > 0 )
{
%>
<form name="formm" method="post" action="./imageUploadmod.jsp">
<input type="hidden" name="seq" value="<%=vseq%>">
</form>
<script language="javascript">
  alert('정상적으로 처리 되었습니다.');
        document.formm.submit();
</script>
<%
}
else
{
%>
  <script language="javascript">
   alert('입력오류 에러입니다.');
   history.go(-1);
</script>
<%
}
%>
```

❻ 브라우저에서 imageUploadmod를 호출하여 이미지를 수정한 후 수정버튼을 클릭하여 이미지를 수정한다.



▶ 수정이미지 선택



▶ 수정결과

❼ 제목과 내용을 수정하는 imageUploadmodSave2.jsp를 작성한다.

▶ [imageUploadmodSave2.jsp]

```
<%@page contentType="text/html;charset=euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%!
 //null 체크를 위한 function으로 nullpoint 에러를 잡는다.
 public static String nullcheck(String str) throws Exception {

        if (str == null){
            return "";
        }else{
                return str;
            }
     }
%>
<%
String vseq= nullcheck(request.getParameter("seq"));
String subject= nullcheck(request.getParameter("subject"));
String contexts= nullcheck(request.getParameter("contexts"));
String sql=" ";
int message=0;
sql="update imageupload set subject='"+subject+"',context=' "+contexts+"' ";
sql+=where seq=' "+vseq+"' ";
%>
<%@ include file="../include/try.jsp" %>
<%
    message=stmt.executeUpdate(sql);
    stmt.close();
%>
<%@ include file="../include/catch.jsp" %>
<%
if( message > 0 )
{
%>
```

다음 페이지에 계속 ➲

```
<form name="formm" method="post" action="./imageUploadmod.jsp">
<input type="hidden" name="seq" value="<%=vseq%>">
</form>
<script language="javascript">
  alert('정상적으로 처리 되었습니다.');
  document.formm.submit();
</script>
<%
}
else
{
%>
  <script language="javascript">
   alert('입력오류 에러입니다.');
   history.go(-1);
</script>
<%
}
%>
```

제목과 내용을 수정하고 하단에 있는 수정버튼을 클릭하면 수정된 것을 확인한다.

❽ 삭제를 위한 imageUploadDelete.jsp를 작성한다.

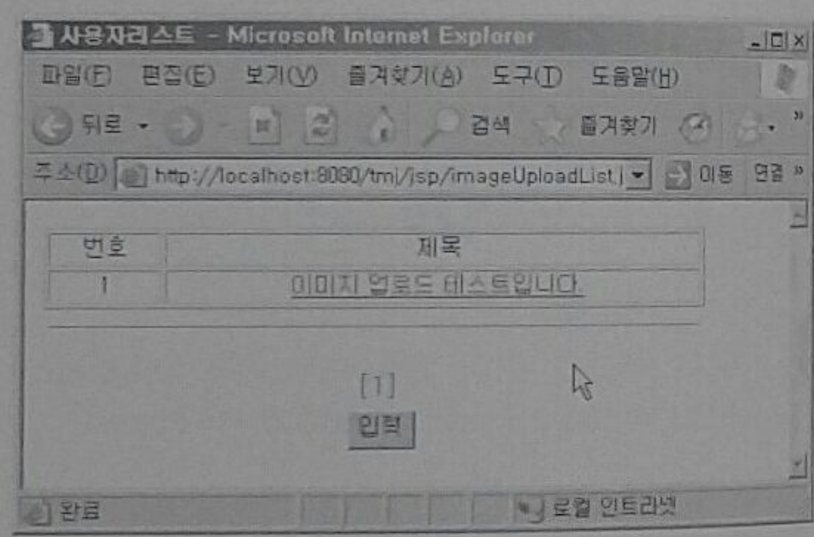
▶ [imageUploadDelete.jsp]

```
<%@page contentType="text/html;charset=euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%!
 //null 체크를 위한 function으로 nullpoint 에러를 잡는다.
 public static String nullcheck(String str) throws Exception {

       if (str == null){
           return " ";
       }else{
                return str;
            }
    }
%>
<%
String vseq= nullcheck(request.getParameter("seq"));
String sql=" ";
int message=0;
 sql="delete from imageupload where seq='"+vseq+"'";
%>
<%@ include file="../include/try.jsp" %>
<%
    message=stmt.executeUpdate(sql);
    stmt.close();
%>
<%@ include file="../include/catch.jsp" %>
```

다음 페이지에 계속 ➲

```
<%
if( message > 0 )
{
%>
<form name="formm" method="post" action="./imageUploadList.jsp">
<input type="hidden" name="seq" value="<%=vseq%>">
</form>
<script language="javascript">
   alert('정상적으로 처리되었습니다.');
   document.formm.submit();
</script>
<%
}
else
{
%>
   <script language="javascript">
    alert('입력오류 에러입니다.');
    history.go(-1);
</script>
<%
}
%>
```

imageView.jsp를 호출하여 삭제버튼을 클릭하면 삭제와 동시에 리스트로 이동, 리스트에서 삭제된 것을 알 수 있다.



▶ 리스트에서 삭제확인

맺음말

이번 장에서는 페이지 나누기와 파일 업로드, 업로드 파일의 다운로드 및 이미지 파일업로드 예제를 통하여 실무에서 사용되는 방법에 대해서 공부하였다. 많이 사용되는 방법이므로 기억하는 것이 좋다. 다음 장에서 엑셀 파일을 이용하여 엑셀 데이터를 데이터베이스에 저장하고 데이터베이스에 있는 자료를 엑셀 파일로 다운 받는 방법과 리스트 화면에서 체크박스를 이용한 컨트롤에 대해서 공부한다.

12 Excel 데이터의 데이터베이스 저장과 체크박스를 이용한 리스트 컨트롤

안녕하세요

이번 장에서는 엑셀 데이터를 읽어 들여 데이터베이스에 저장하는 방법, 저장된 데이터를 엑셀파일로 만드는 2가지 방법과 체크박스를 이용한 컨트롤에 대해서도 공부한다. 이런 방법은 실무에서 자주 사용되는 수단으로 알아두면 유용한 팁이 된다.

Excel 데이터 저장방법

많은 양의 필요한 데이터를 저장해야 한다면 우리는 ODBC를 이용한 데이터 저장방법을 생각한다. 그러나 엑셀 파일 내에 존재하는 데이터를 가공하고 ODBC 설정 및 드라이버를 설치하는 등 많은 번거로움이 있다. 따라서 프로그램을 통해 바로 엑셀파일의 내용을 데이터베이스에 저장하고 저장된 데이터를 이용하여 필요한 정보를 뽑아내거나 또는 가공된 데이터를 다시 엑셀파일로 언제든지 다운로드 받을 필요성이 있다.

물론 엑셀파일의 내용을 데이터베이스에 저장하는 방법에는 약간의 제약이 따르지만 그 제약을 지킨다면 업무의 효율성을 높일 수 있디. 예제를 통해 이러한 방법을 알아보기로 한다.

엑셀 파일은 업로드할 때 업로드와 동시에 데이터베이스에 바로 저장되는 것이 아니라 먼저 엑셀 파일을 업로드한 후, 업로드된 엑셀 파일을 읽어 해당 엑셀 파일의 데이터를 데이터베이스에 저장한다.

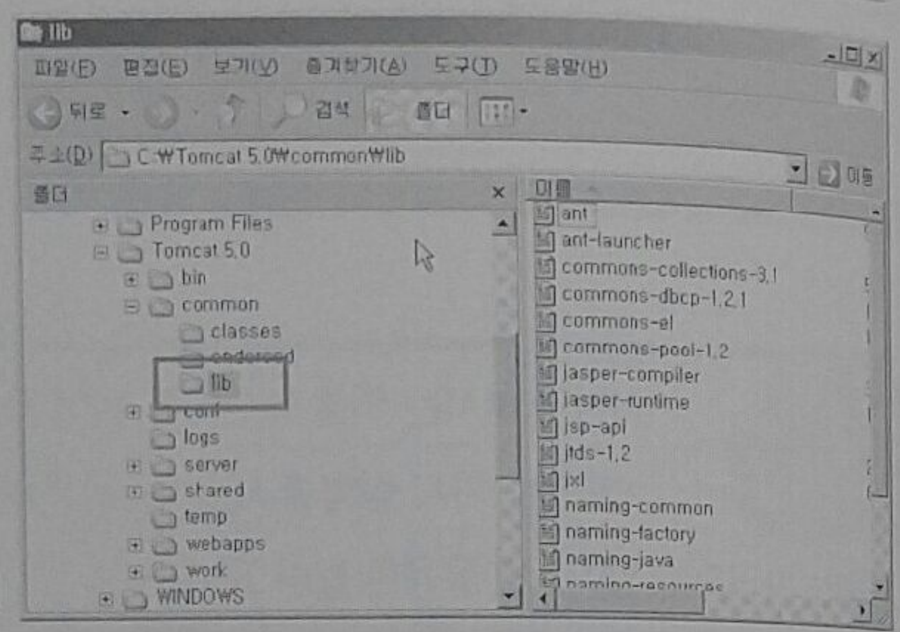
엑셀 파일을 데이터베이스에 저장하기 위한 방법은 JExcelAPI를 이용한 방법과 POI를 이용한 2가지의 방법을 자바에서 제공한다. 이 두 가지의 차이점은 JExcelAPI가 POI보다 많은 양의 데이터를 한꺼번에 처리할 수 있으며, 속도가 빠르다는 것이다. 단점은 저장된 데이터를 엑셀 파일로 다운로드하여 엑셀 프로그램에서 확인할 때는 JExcelAPI는 무늬만 엑셀 형식인 반면에 POI는 진짜 엑셀 형식이 된다는 것이다.

따라서 이 두 가지를 병행해서 사용하는데 단순한 엑셀 파일의 보기와 출력은 JExcelAPI을 사용하고 엑셀 파일을 다운로드 받아 엑셀 데이터를 가공(연산 등)하여 재사용할 때는 POI를 사용하게 된다.

JExcelAPI를 이용한 엑셀 파일의 저장

따라하기 JExcelAPI를 이용한 엑셀파일의 저장

❶ 도서출판 대림 홈페이지 자료실에서 다운받은 파일의 jxl 폴더의 jxl.jar 파일을 Tomcat의 common 아래 lib에 복사한다.



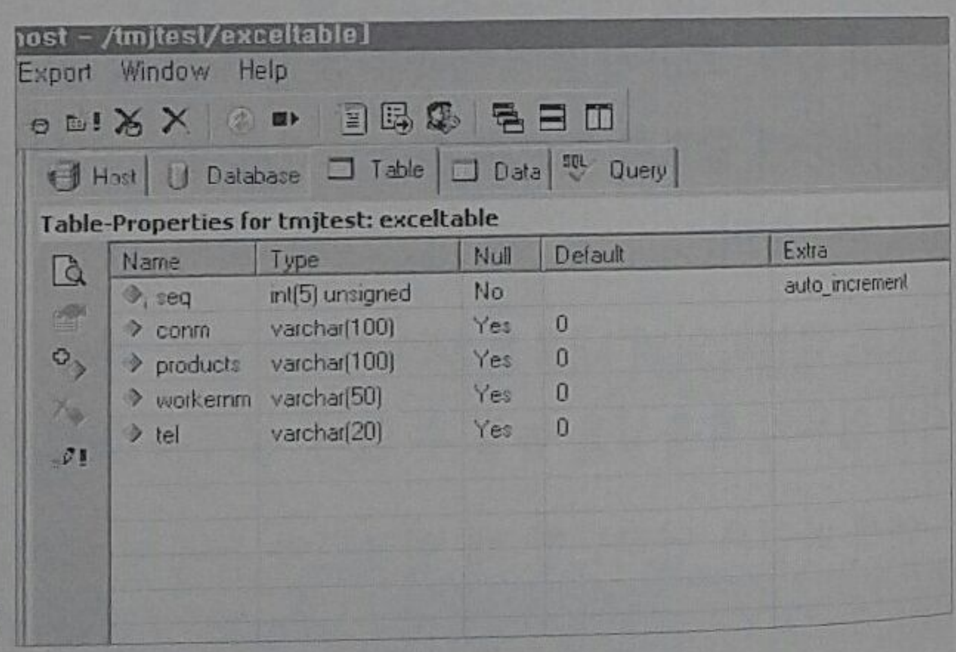
▶ Jxl.jar 파일

❷ 데이터베이스에 저장하려는 엑셀 파일(excelupload.xls)을 엑셀 프로그램을 이용하여 생성한다.

	A	B	C	D
1	회사명	주거래물품	거래처 담당자	연락처
2	금동나라	비누	금동이	011-222-3333
3	영호식품	일반식품	김영호	011-333-2222
4	용남택배	배송	이용남	016-777-0909
5	훈이문구	문구	이훈	016-342-0000
6	기천상사	일반문구	이기천	010-343-1111
7				
8				

▶ 예제 Excel 파일

❸ 엑셀 파일의 데이터를 저장할 테이블을 생성하고 저장하려는 엑셀 파일의 데이터 셀 구조와 같이 테이블의 칼럼을 생성한다.



▶ 테이블 생성

❹ 엑셀 파일을 업로드하는 화면을 작성한다.

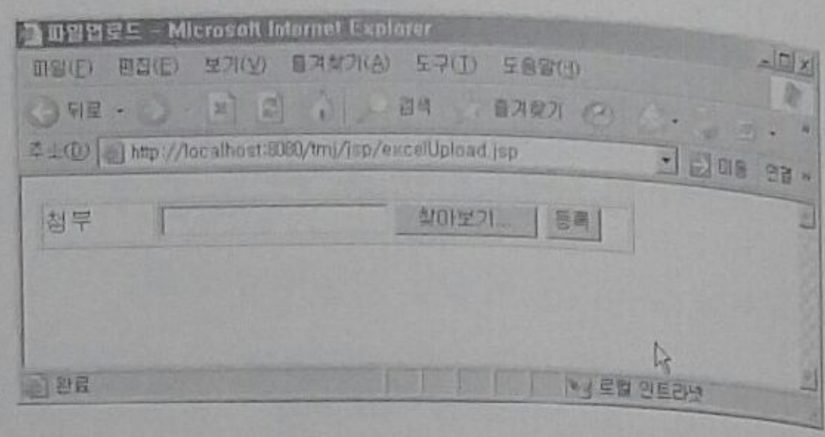
▶ excelUpload.jsp

```
<%@ page language="java" contentType="text/html; charset=euc-kr" %>
<html>
<head>
<title>파일업로드</title>
<script language="javascript">
function go_upload()
```

다음 페이지에 계속 ➲

```
{
var TheForm = document.uploadForm;
if(TheForm.addfile.value =='' )
{
alert( '업로드할 파일을 선택해 주세요' );
}
else
{

//엑셀파일만 업로드 할수 있게 체크하는 function
if(confirm( '엑셀데이터를 저장하시겠습니까?' ))
{
        var fname=TheForm.addfile.value;
        var arr=("file:///"+fname.replace(/ /gi,"%20").replace(/\\\\/gi,"/")).split("/");
        var fm =arr[arr.length-1];
        var m=fm.lastIndexOf(".");
        var e=fm.length;
        var extname=fm.substring(m+1,e);

if(extname !='xls'){
        alert( '엑셀파일만 업로드합니다.' );
}
else
{
        TheForm.action="./excelSavePOI.jsp";
        TheForm.submit();
        }

}

}
}
</script>
</head>
<body>
<form name="uploadForm" method="post" enctype="multipart/form-data">
<table border="1" width="400">
<tr>
<td width="80">첨부</td>
<td width="320">
<input type="file" name="addfile">
<input type="button" name="up_btn" value="등록" onClick="javascript:go_upload()">
</td>
</tr>
</table>
</form>
</body>
</html>
```

❺ 브라우저에 excelUpload.jsp를 호출하여 화면을 확인한다.



▶ excelUpload.jsp

❻ 엑셀 파일이 업로드되면 파일이 저장되는 디렉토리에서 엑셀 파일을 읽어 각 셀에 있는 값을 생성된 exceltable 테이블의 칼럼에 저장을 한다.

```
<%@page contentType="text/html;charset=euc-kr"%>
<!--jxl을 반드시 import 해야 한다.-->
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource,jxl.*" %>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%!
 //null 체크를 위한 function
 public static String nullcheck(String str) throws Exception {

        if (str == null){
            return "";
        }else{
            return str;
        }
    }
%>
<%!
 //한글 변환 function
 public static String dbTxt2kor(String sFromDb)
                            throws UnsupportedEncodingException
{
        if (sFromDb == null) return null;
        else return new String(sFromDb.getBytes("KSC5601"), "8859_1");
 }
%>
<%
Connection con=null;
//연속적인 쿼리 실행을 위한 PreparedStatement 선언
PreparedStatement ps = null;

String savePath="C:/tmj/upload/"; // 저장할 디렉토리
int sizeLimit=1024*1024*5; //용량 제한 5메가
String str_cell = null; //엑셀 파일 데이터의 셀의 값을 받는 변수
String sql="";

String xlsfilename="";


%>
```

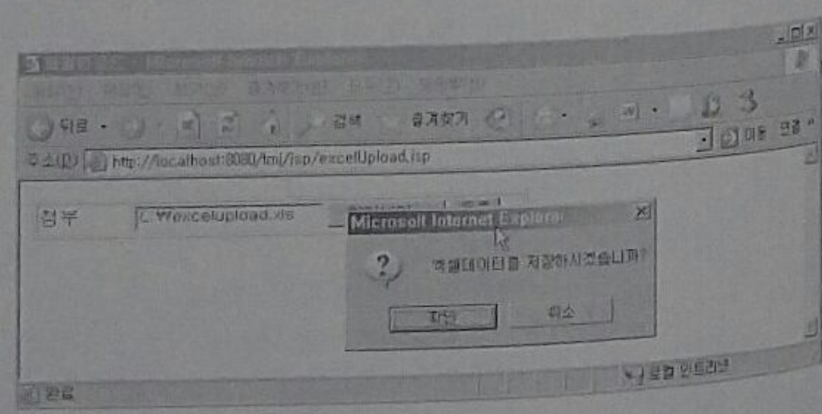
다음 페이지에 계속 ❼

```
//파일 업로드를 위한 Multipart의 선언
MultipartRequest multi
= new MultipartRequest(request,savePath,sizeLimit,new DefaultFileRenamePolicy());
String fileName1 = nullcheck(multi.getFilesystemName("addfile"));

///엑셀 데이터를 저장하기 위한 워크북 및 시트 등의 변수 선언
Workbook workbook = null;
Sheet sheet = null;
Cell cell = null;

//업로드된 경로와 파일의 이름
xlsfilename=savePath+fileName1;
sql="insert into exceltable(conm,products,workernm,tel) values (?,?,?,?)";
<%
 try
   {
       //업로드된 경로에서 엑셀 파일을 인지하여 불러온다.
       workbook = Workbook.getWorkbook( new File( xlsfilename));
       if( workbook != null)
       {
           //엑셀 파일에서 첫 번째 시트를 인식
           //만약 저장하려는 데이터가 다른 시트에 있다면
           //해당 시트의 번호를 넣어주면 된다.
           sheet = workbook.getSheet(0);
           if( sheet != null)
           {
             // 실제 엑셀 데이터가 시작되는 Row지정
              // 엑셀 데이터에 타이틀이 없는 경우는 0
           int nRowStartIndex = 1;
           //실제 엑셀 데이터가 끝 Row 지정
           int nRowEndIndex   = sheet.getColumn(0).length - 1;

            //실제 엑셀 데이터가 시작되는 Column 지정
            int nColumnStartIndex = 0;
            //실제 엑셀 데이터가 끝나는 Column 지정
            int nColumnEndIndex = sheet.getRow(0).length - 1;

       try
       {
        //include하지 않고 바로 선언
       Context initCtx = new InitialContext();
       Context envCtx = (Context) initCtx.lookup("java:comp/env");
       DataSource ds = (DataSource) envCtx.lookup("jdbc/tmjtest");
       con = ds.getConnection();
       ps = con.prepareStatement(sql);
       for( int nRow = nRowStartIndex; nRow <= nRowEndIndex; nRow++ )
       {
       for( int nColumn = nColumnStartIndex;
                           nColumn <= nColumnEndIndex ; nColumn++)
{

           //row에 해당하는 cell의 값
          cell = sheet.getCell( nColumn, nRow);
          //해당 cell의 값을 변수 str_cell에 대입
          str_cell=cell.getContents();
         //한글 변환 function을 사용하여 한글처리
```

다음 페이지에 계속

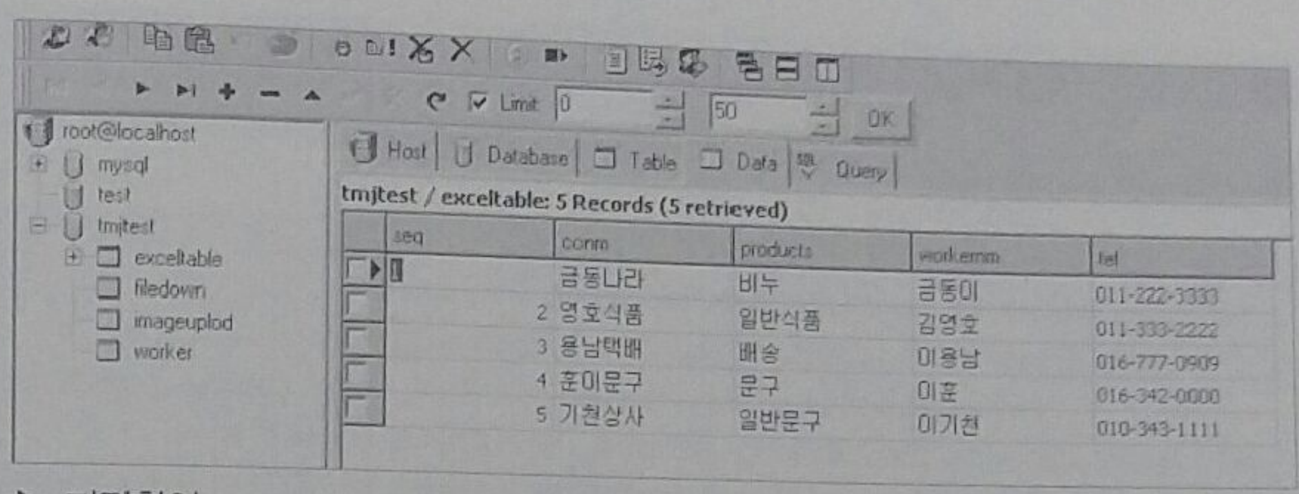
```
            ps.setString(nColumn+1, dbTxt2kor(str_cell));
          }
        ps.executeUpdate();
      }
      ps.close();
  }
catch(NamingException ne)
{
   out.println(ne.toString());
}
 }
}
}
catch( Exception e)
{
   e.printStackTrace();
   out.println(e);
%>
<form name="formm" method="post" action="./excelUpload.jsp">
</form>
<script language="javascript">
        alert('오류가 발생하였습니다.');
        document.formm.submit();
</script>
<%
    }
    finally
    {
        if( workbook != null)
        {
            workbook.close();
        }
%>
<form name="formm" method="post" action="./excelUpload.jsp">
</form>
<script language="javascript">
        alert('정상적으로 저장되었습니다.');
        document.formm.submit();
</script>
<%
    }
%>
```

❺ 브라우저를 열고 excelUpload.jsp를 호출하여 예제로 만들어진 excelupload.xls 파일을 선택하고 등록 버튼을 클릭한다.



▶ 파일선택

❻ 등록 버튼을 클릭 후 데이터베이스에 저장되었는지 확인한다.



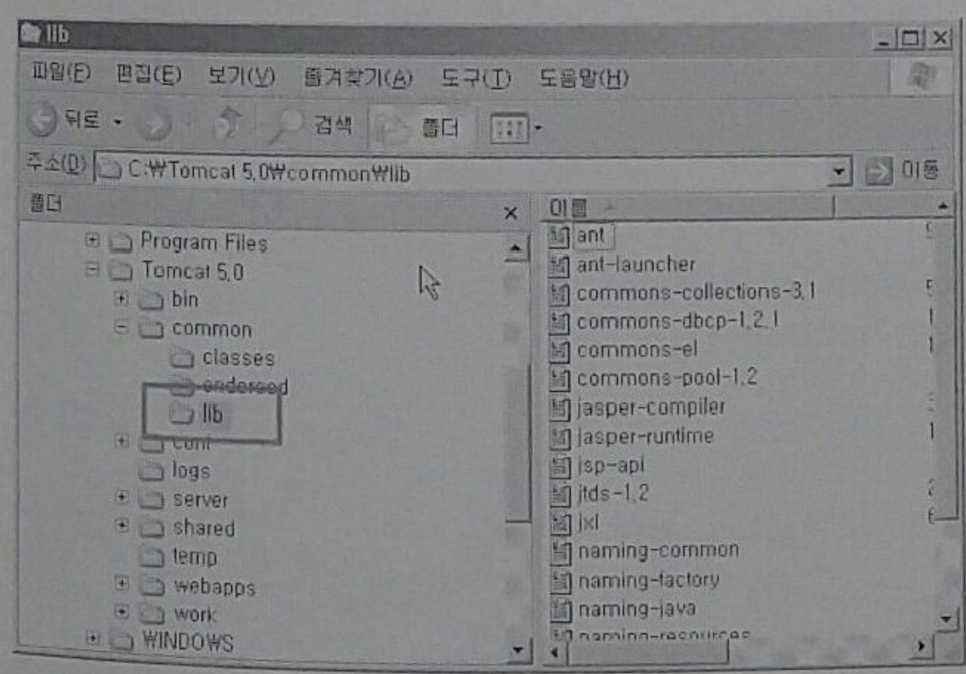
▶ 저장확인

POI를 이용한 엑셀 파일의 저장

JExcelAPI를 이용한 엑셀 파일의 저장 시 문제점은 셀의 타입을 지정할 수 없는 것이다. 이런 단점은 엑셀파일의 데이터가 가지는 각 셀의 타입에 따라 저장되지 않아 추후 해당 데이터를 다운로드하여 엑셀 프로그램에서 가공 시 계산, 날짜형 등에 오류를 나타내기도 한다. 하지만 POI 방법은 각 해당 셀의 타입을 지정할 수 있다.

따라하기 POI를 이용한 엑셀파일의 저장

❶ POI를 사용하기 위해 도서출판 대림 홈페이지 자료실에서 다운받은 poi-3.0-alpha2-20060616.jar 파일을 jxl.jar와 마찬가지 위치인 Tomcat의 commom 아래의 lib에 복사한다.



▶ POI 복사

❷ 업로드한 엑셀 파일의 데이터를 저장하는 excelSavePOI.jsp을 작성한다.

▶ [excelSavePoi.jsp]

```
<%@page contentType="text/html;charset=euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
          javax.naming.*,javax.sql.DataSource, org.apache.poi.hssf.usermodel.*" %>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%!
//null 체크를 위한 function으로 nullpoint 에러를 잡는다.
public static String nullcheck(String str) throws Exception {

        if (str == null){
            return " ";
        }else{
                return str;
            }
    }
%>
<%!
 public static String dbTxt2kor(String sFromDb)
                              throws UnsupportedEncodingException
{
        if (sFromDb == null) return null;
        else return new String(sFromDb.getBytes("KSC5601"), "8859_1");
}
%>
<%
Connection con=null;
//연속적인 쿼리 실행을 위한 PreparedStatement 선언
PreparedStatement ps = null;

String savePath="C:/tmj/upload/"; //저장할 디렉토리
int sizeLimit=1024*1024*5; //용량설정
String str_cell = null;
String sql=" ";
String xlsfilename=" ";
MultipartRequest multi =
new MultipartRequest(request,savePath,sizeLimit,new DefaultFileRenamePolicy());
String fileName1 = nullcheck(multi.getFilesystemName("addfile"));

//엑셀 데이터를 저장하기 위한 워크북 및 시트 등의 변수 선언
HSSFWorkbook workbook = null;
HSSFSheet sheet = null;
HSSFRow row = null;
HSSFCell cell = null;

//업로드된 엑셀 파일의 경로와 파일의 이름을 받는다.
xlsfilename=savePath+fileName1;
sql="insert into exceltable(conm,products,workernm,tel) values (?,?,?,?)";
 try
    {
    //엑셀 파일이 업로드된 디렉토리에서
    //해당 엑셀 파일을 FileInputStream를 이용하여 읽어 들이고
    //HSSFWorkbook의 객체를 새로 생성하여 엑셀파일의 각 셀의 데이터를 대입한다.
```

다음 페이지에 계속 ❶

```
 FileInputStream fis = new FileInputStream(new File(xlsfilename));
 workbook = new HSSFWorkbook(fis);
 fis.close();
if (workbook != null)
{
 // 불러들일 시트 선언
 // 만약 불러들일 엑셀 데이터가 다른 시트에 있다면
 // 시트의 값을 변경하면 된다.
 sheet = workbook.getSheetAt(0);
  if (sheet != null)
   {
     //실제 엑셀 데이터가 시작되는 Row 지정
     int nRowStartIndex = 1;
     //실제 엑셀 데이터가 끝 Row 지정
     int nRowEndIndex = sheet.getLastRowNum();
     //실제 엑셀 데이터가 시작되는 Column 지정
     int nColumnStartIndex = 0;
     //실제 엑셀 데이터가 끝나는 Column 지정
     int nColumnEndIndex = sheet.getRow(0).getLastCellNum();
     //해당 셀의 값을 받는 변수 선언
     String szValue = "";
     try
     {
      //include하지 않고 바로 선언
      Context initCtx = new InitialContext();
      Context envCtx = (Context) initCtx.lookup("java:comp/env");
      DataSource ds = (DataSource) envCtx.lookup("jdbc/tmjtest");
      con = ds.getConnection();
      ps = con.prepareStatement(sql);
      for (int i = nRowStartIndex; i <= nRowEndIndex; i++)
       {
        row = sheet.getRow(i);
       for (int nColumn = nColumnStartIndex;
                    nColumn <= nColumnEndIndex; nColumn++)
       {
         cell = row.getCell((short) nColumn);
         //빈 셀의 경우 스킵한다.
        if (cell == null)
        {
          continue;
        }
        else
        {
         szValue =cell.getStringCellValue();
         ps.setString(nColumn+1,dbTxt2kor(szValue));
        }
        }
        ps.executeUpdate();
       }
      ps.close();
     }
     catch(NamingException ne)
    {
      out.println(ne.toString());
    }
```

다음 페이지에 계속

```
        }
       }
      }
      catch( Exception e)
     {
        e.printStackTrace();
         out.println(e);
%>
<form name="formm" method="post" action="./excelUpload.jsp">
</form>
<script language="javascript">
         alert('오류가 발생하였습니다.');
         document.formm.submit();
</script>
<%
    }
    finally
    {
%>
<form name="formm" method="post" action="./excelUpload.jsp">
</form>
<script language="javascript">
         alert('정상적으로 저장되었습니다.');
         document.formm.submit();
</script>
<%
    }
%>
```

```
rce org.apache.poi.hssf.usermodel.*" %>
```

▶ 반드시 import 해야 한다.

❸ excelUpload.jsp에서 저장을 호출하는 action을 TheForm.action="./excelSavePOI.jsp"; 으로 바꿔준다.

❹ 브라우저를 열고 예제를 위해 만든 엑셀 파일을 업로드하고 저장이 정상적으로 되었는지 데이터베이스에서 확인한다.

seq	conm	products	workemm	tel
1	금동나라	비누	금동이	011-222-3333
2	영호식품	일반식품	김영호	011-333-2222
3	용남택배	배송	이용남	016-777-0909
4	훈이문구	문구	이훈	016-342-0000
5	기천상사	일반문구	이기천	010-343-1111
20	A	상품1	김1	011-111-2222
21	B	상품2	김2	011-111-2223
22	C	상품3	김3	011-111-2224
23	D	상품4	김4	011-111-2225
24	E	상품5	김5	011-111-2226
25	F	상품6	김6	011-111-2227
26	G	상품7	김7	011-111-2228
27	H	상품8	김8	011-111-2229
28	I	상품9	김9	011-111-2230
29	J	상품10	김10	011-111-2231
30	K	상품11	김11	011-111-2232
31	L	상품12	김12	011-111-2233
32	M	상품13	김13	011-111-2234
33	N	상품14	김14	011-111-2235

▶ 저장확인

JExcelAPI를 이용한 엑셀파일 다운로드

JExcelAPI를 사용하여 저장된 데이터를 엑셀 파일로 다운로드 받는 방법은 아주 간단하다. 그러나 JExcelAPI은 진정한 엑셀 파일의 형식이 아니라 겉모습만 엑셀 파일 형식이다. 따라서 저장된 데이터를 엑셀 파일 형식으로 출력하거나 단순히 보여주는 기능만이 필요하다면 JExcelAPI를 사용하면 좋다.

따라하기 JExcelAPI를 이용한 엑셀파일 다운로드

❶ 데이터베이스에 저장된 데이터를 가져와 리스트를 출력하는 excelList.jsp를 작성한다.

▶ [excelList.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%
String listSql="select * from exceltable";
String pageCountQuery = "select count(1) from exceltable ";

// 페이지 설정을 위한 변수 선언
int view_rows =10;
int total_pages =1;                    //전체 페이지 (default 1)
int tpage=1;                           //현재 페이지 (default 1)

String url = "";
   url = "./excelList.jsp?tpage=";

if (!"ok".equals(request.getParameter("fullview"))
                                  && request.getParameter("tpage") != null)
   tpage = Integer.parseInt(request.getParameter("tpage"));

if (!"ok".equals(request.getParameter("fullview")))
   listSql += " limit "+((tpage-1)*view_rows)+","+view_rows;
%>
<%@ include file="../include/try.jsp" %>
<%
        rs = stmt.executeQuery(listSql);
%>
<html>
<head>
<title>excel 다운로드</title>
<script language="javascript">
  function downExcel()
  {
    var TheForm = document.excelForm;
    TheForm.action="./exceldown.jsp";
    TheForm.submit()
  }
```

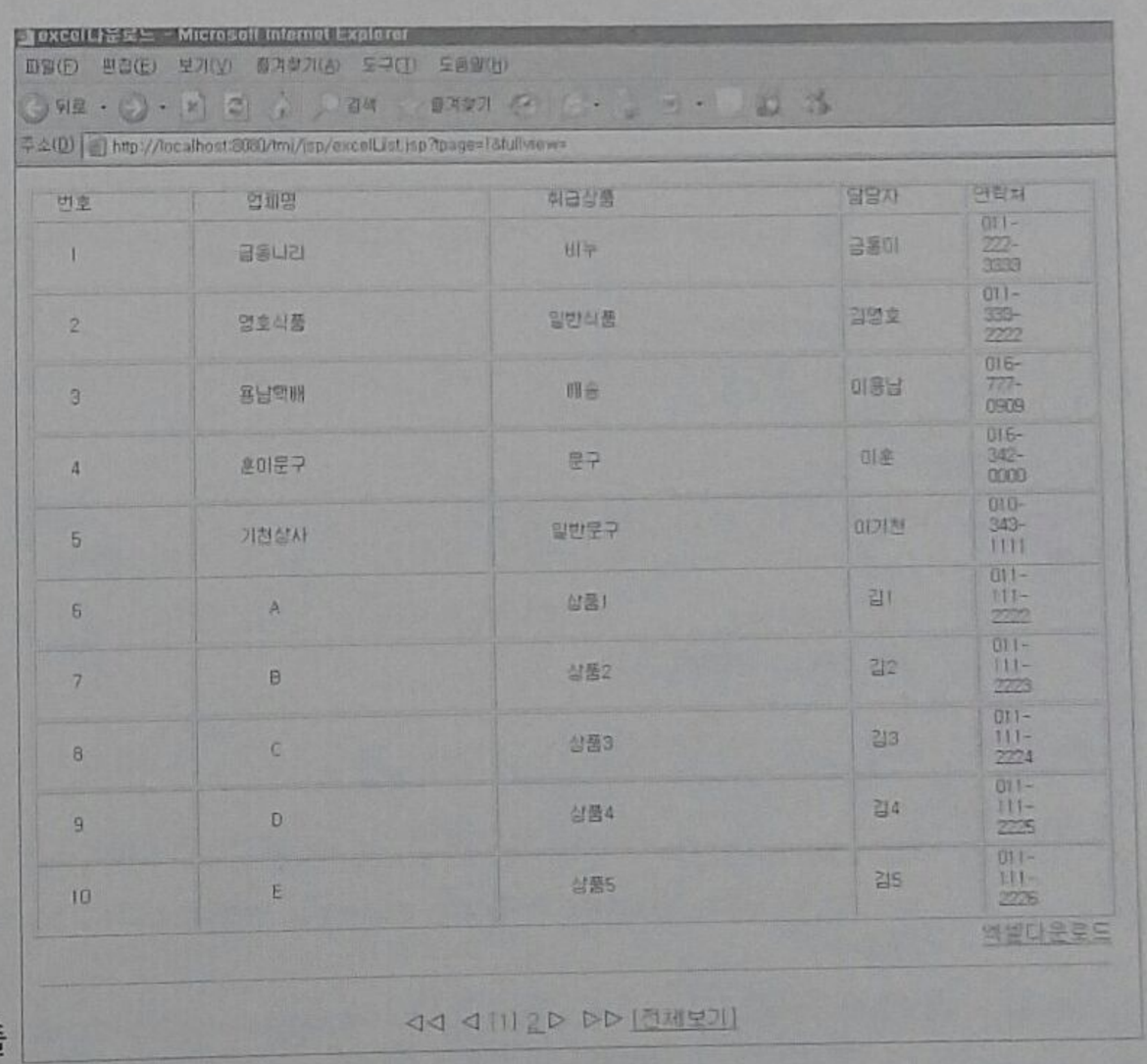
다음 페이지에 계속 ➲

```
</script>
</head>
<body >
<form name="excelForm" method="post">
<table border="1">
    <tr>
        <td width="60">
            <p align="center"><font size="2">번호</font></p>
        </td>
        <td width="115">
            <p align="center"><font size="2">업체명</font></p>
        </td>
        <td width="135">
            <p align="center"><font size="2">취급상품</font></p>
        </td>
        <td width="45">
            <p align="center"><font size="2">담당자</font></p>
        </td>
                <td width="45">
            <p align="center"><font size="2">연락처</font></p>
        </td>
    </tr>
<!--while문을 통해 리스트를 출력한다.-->
<%
         int num=1;
         if (!"ok".equals(request.getParameter("fullview")))
        num = ((tpage-1)*view_rows)+1;
         while(rs.next())
         {
%>
    <tr>
        <td width="60">
            <p align="center"><font size="2">><%=num%></font></p>
        </td>
        <td width="115">
            <p align="center"><font size="2">><%=rs.getString("conm")%></font></p>
        </td>
        <td width="135">
            <p align="center">
               <font size="2">><%=rs.getString("products")%></font>
            </p>
        </td>
        <td width="45">
            <p align="center">
              <font size="2">><%=rs.getString("workernm")%></font>
            </p>
        </td>
           <td width="45">
            <p align="center">><font size="2">><%=rs.getString("tel")%></font></p>
        </td>
    </tr>
<%
        //번호에 대한 증가 값
        num++;
        }
```

다음 페이지에 계속

```
        if(num ==1)
        {
%>
        <tr><td colspan=4 align=center>해당하는 데이터가 없습니다.</td></tr>
<%
        }
%>
</table>
<table border="0" width="383">
<tr><td align="right"><a href="#" onClick="downExcel()">Excel 다운로드</a></td></tr>
</table>
<table border="0" width="383">
<tr><td align="center"><%@ include file="../include/pageCount.jsp"%></td></tr>
</table>
</form>
</body>
</html>
<%@ include file="../include/catch.jsp" %>
```

❷ 브라우저를 열고 작성된 excelList.jsp를 호출한다.



▶ excelList.jsp의 호출

❸ 리스트에서 엑셀 다운로드를 클릭하면 다운로드 역할을 하는 exceldown.jsp를 작성한다. exceldown.jsp의 작성은 excelList.jsp를 그대로 카피하고 헤더 부분만 추가하고 exceldown.jsp으로 저장하면 된다.

▶ [exceldown.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%
String listSql="select * from exceltable";
//엑셀 파일 형식으로 다운로드를 위한 헤더의 설정
```

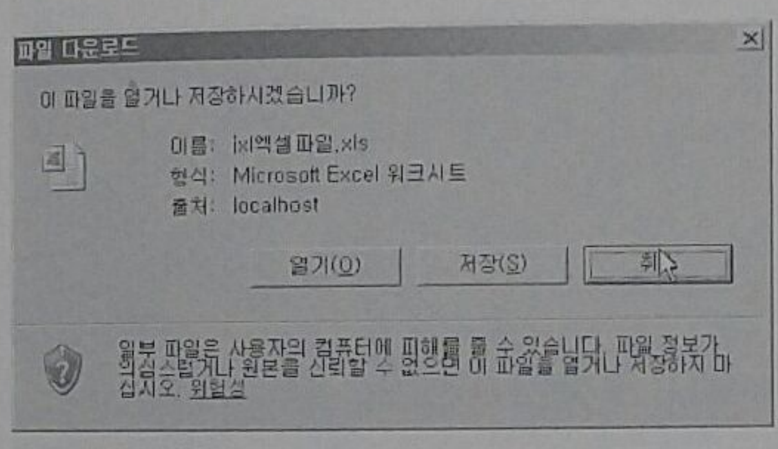
다음 페이지에 계속 ➲

```
response.setHeader("Content-Disposition", "attachment; filename=jxlExcel파일.xls");
response.setHeader("Content-Description", "JSP Generated Data");

%>
<%@ include file="../include/try.jsp" %>
<%
        rs = stmt.executeQuery(listSql);
%>
<html>
<head>
<title>excel 다운로드</title>
</head>
<body >
<form name="exceldown" method="post">
<table border="1" width="800">
    <tr>
        <td width="60">
            <p align="center"><font size="2">번호</font></p>
        </td>
        <td width="115">
            <p align="center"><font size="2">업체명</font></p>
        </td>
        <td width="135">
            <p align="center"><font size="2">취급상품</font></p>
        </td>
        <td width="45">
            <p align="center"><font size="2">담당자</font></p>
        </td>
          <td width="45">
            <p align="center"><font size="2">연락처</font></p>
        </td>
    </tr>
<!--while문을 통해 리스트를 출력한다.-->
<%
         int num=1;
         while(rs.next())
         {
%>
    <tr>
        <td width="60">
            <p align="center"><font size="2"><%=num%></font></p>
        </td>
        <td width="115">
            <p align="center"><font size="2"><%=rs.getString("conm")%></font></p>
        </td>
        <td width="135">
            <p align="center">
              <font size="2"><%=rs.getString("products")%></font>
            </p>
        </td>
        <td width="45">
            <p align="center">
              <font size="2"><%=rs.getString("workernm")%></font>
            </p>
        </td>
            <td width="45">
            <p align="center"><font size="2"><%=rs.getString("tel")%></font></p>
        </td>
```

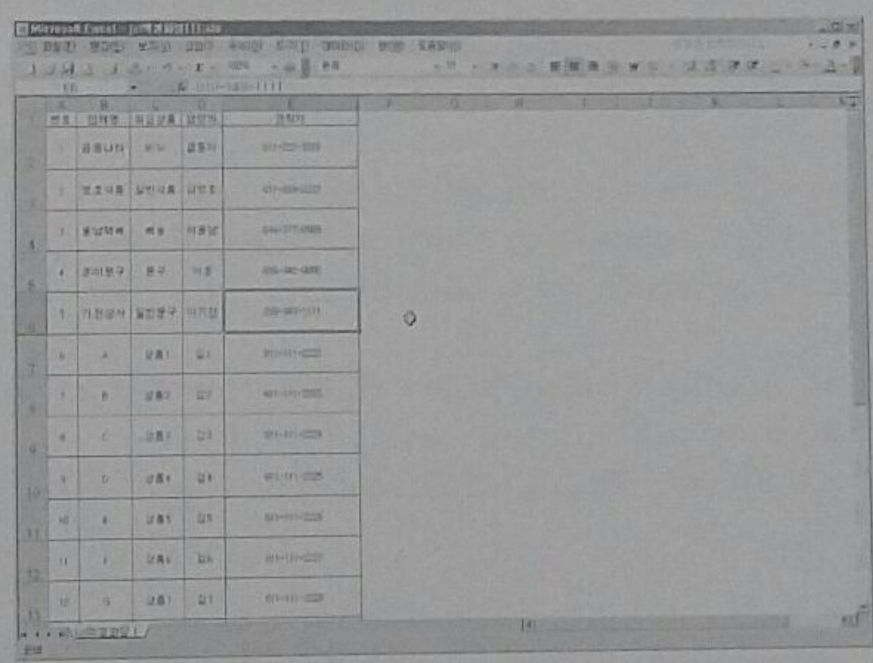
다음 페이지에 계속

```
    </tr>
<%
        //번호에 대한 증가 값
        num++;
        }
        if(num ==1)
        {
%>
        <tr><td colspan=4 align=center>해당하는 데이터가 없습니다.</td></tr>
<%
            }
%>
</form>
</body>
</html>
<%@ include file="../include/catch.jsp" %>
```

❹ 엑셀 다운로드를 클릭하면 저장 여부가 나온다. 바로 열기를 통해 엑셀을 확인해 본다.



▶ 저장여부



▶ Excel 프로그램을 이용해 읽은 데이터

JExcelAPI를 통해 엑셀파일로 다운로드 받으면 약간 이상한 모양의 엑셀파일 형태를 본다. 이는 단지 화면상에 보이는 데이터를 엑셀파일 데이터처럼 보이게 하는 것뿐이다.

POI를 이용한 엑셀파일 다운로드

JExcelAPI를 통해 전해 받은 엑셀파일은 계산 등, 몇몇 부분들이 완전한 엑셀형식과 다르다. 앞서서 이야기했지만 JExcelAPI는 단지 화면상에 있는 리스트를 HTML 형식으로 엑셀 프로그램에서 불러와 보여주는 것뿐이다. 하지만 POI는 엑셀형식으로 보일 때 각 셀의 타입을 지정해서 데이터가 들어가기 때문에 완전한 엑셀 형식의 파일이라고 볼 수 있다.

따라하기 POI를 이용한 엑셀파일 다운로드

❶ excelList.jsp에서 엑셀파일을 다운받는 function의 action을 exceldownPOI.jsp로 바꿔준다.

```
title>사용자리스트</title>
script language="javascript">
 function downExcel()
 {
        var TheForm = document.excelForm;
        TheForm.action="./exceldownPOI.jsp";
        TheForm.submit()
 }
/script>
/head>
```

▶ action 변경

❷ POI 방식의 예제를 위해 exceldownPOI.jsp를 작성한다.

▶ [exceldownPOI.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                        javax.naming.*,javax.sql.DataSource" %>
<%@ page import="org.apache.poi.poifs.filesystem.*" %>
<%@ page import="org.apache.poi.poifs.filesystem.POIFSFileSystem.*" %>
<%@ page import="org.apache.poi.poifs.dev.*" %>
<%@ page import="org.apache.poi.hssf.record.*" %>
<%@ page import="org.apache.poi.hssf.record.formula.*" %>
<%@ page import="org.apache.poi.hssf.model.*" %>
<%@ page import="org.apache.poi.hssf.usermodel.*" %>
<%@ page import="org.apache.poi.hssf.util.*" %>
<%!
  public static String kor2Db(String sToDb) throws UnsupportedEncodingException{
        if (sToDb == null) return null;
        else return new String(sToDb.getBytes("8859_1"), "KSC5601");
    }
%>
<%
String listSql="select conm,products,workernm,tel from exceltable";

//엑셀 데이터를 저장하기 위한 워크북 및 시트 등의 변수 선언
HSSFWorkbook wb = null;
HSSFSheet sheet = null;
HSSFRow row = null;
HSSFCell cell = null;
int rsCnt=1;

String filename = "POIExcel다운.xls";
String filepath="C:/tmj/upload/";
%>
<%@ include file="../include/try.jsp" %>
<%
     stmt = con.createStatement();

     //워크북 객체 생성
     wb = new HSSFWorkbook();
     //새로운 시트 생성
     sheet = wb.createSheet("new sheet");
     /*셀 서식 */
     HSSFCellStyle style = wb.createCellStyle();
     style.setBorderBottom(HSSFCellStyle.BORDER_THIN);
     style.setBottomBorderColor(HSSFColor.BLACK.index);
```

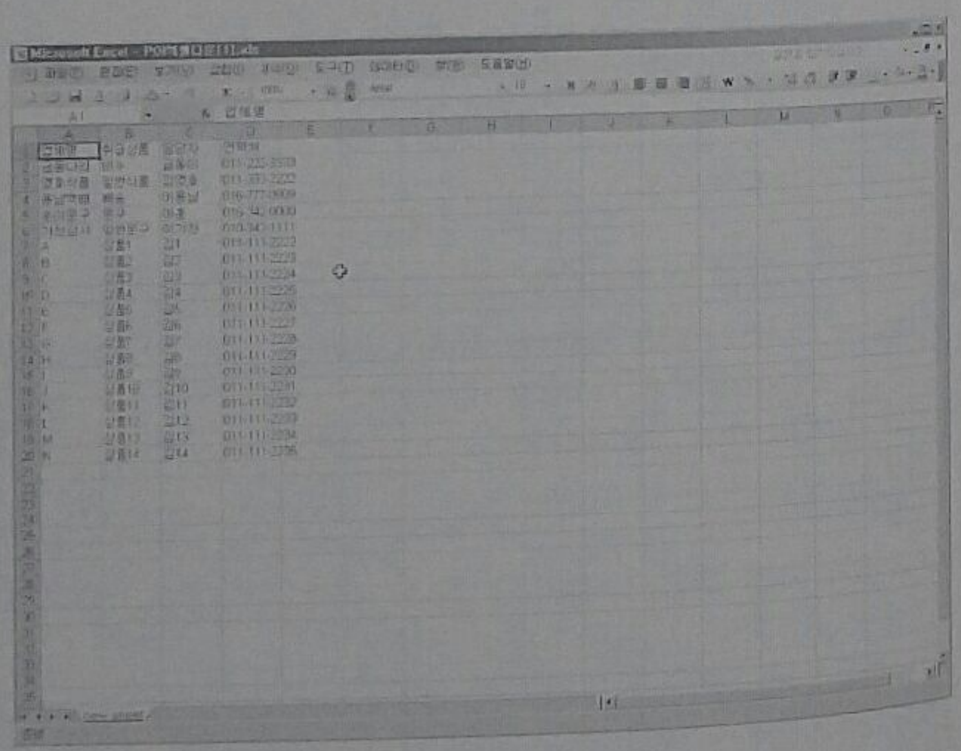
다음 페이지에 계속 ❶

```
style.setBorderLeft(HSSFCellStyle.BORDER_THIN);
style.setLeftBorderColor(HSSFColor.GREEN.index);
style.setBorderRight(HSSFCellStyle.BORDER_THIN);
style.setRightBorderColor(HSSFColor.BLUE.index);
style.setBorderTop(HSSFCellStyle.BORDER_MEDIUM_DASHED);
style.setTopBorderColor(HSSFColor.BLACK.index);

rs = stmt.executeQuery(listSql);
//메타 데이터 생성(테이블의 칼럼명, 속성 등의 정보를 가지고 온다.)
ResultSetMetaData rsmd =  rs.getMetaData();
//칼럼 수 카운트
int numberOfColumns = rsmd.getColumnCount();
//Excel 파일의 타이들을 만들기 위한 배열값
String[] tbl_title   = {"업체명","취급상품","담당자","연락처"};
//칼럼의 값을 받을 배열 변수 선언
String[] ColumnsName=new String[numberOfColumns];
//칼럼의 타입 정보(즉 String, Integer, Decimal 등 자료형 정보)
int[] ColumnsType = new int[numberOfColumns];
row = sheet.createRow((short)0);
for(int i=0 ; i<numberOfColumns;i++)
{
  ColumnsType[i]=rsmd.getColumnType(i+1);
  cell = row.createCell((short)(i));
  // kor2Db을 통해 한글처리
  // 타이틀을 세팅한다.
  cell.setCellValue(kor2Db(tbl_title[i]));
}

//데이터베이스의 내용을 출력
while(rs.next())
{
 row = sheet.createRow((short)rsCnt);
 for(int k=0 ; k<numberOfColumns;k++)
 {
  cell = row.createCell((short)k);
  //kor2Db을 통해 한글처리
  cell.setCellValue(kor2Db(rs.getString(k+1)));
  }

  rsCnt++;
}
 //파일 스트림으로 파일을 읽어 들인다.
 FileOutputStream fileOut  = new FileOutputStream(filepath+filename);
 wb.write(fileOut );
 fileOut.close();
%>
<%
//여기부터 파일 다운로드 창이 자동으로 뜨게 하기 위한 코딩
//(임시 파일을 스트림으로 저장)
//해당 경로의 파일 객체를 만든다.
File file = new File (filepath+filename);

//파일 스트림을 저장하기 위한 바이트 배열 생성.
byte[] bytestream = new byte[(int)file.length()];

//파일 객체를 스트림으로 불러온다.
FileInputStream filestream = new FileInputStream(file);
```

다음 페이지에 계속

```
 //파일 스트림을 바이트 배열에 넣는다.
 int i = 0;
 int j = 0;
 while((i = filestream.read()) != -1)
{
  bytestream[j] = (byte)i;
  j++;
 }
 filestream.close();   //FileInputStream을 닫아줘야 file이 삭제된다.

 try
{
  //파일을 생성과 동시에 byte[] 배열에 입력 후 파일은 삭제
  boolean  success = file.delete();
  if(!success) System.out.println("<script>alert('not success')</script>");
 }
catch(IllegalArgumentException e)
{
  System.err.println(e.getMessage());
}

//Content-Disposition 헤더에 파일 이름 세팅.
 response.setHeader("Content-Disposition","attachment; filename="+filename);

// 응답 스트림 객체를 생성한다.
 OutputStream outStream = response.getOutputStream();

// 응답 스트림에 파일 바이트 배열을 쓴다.
 outStream.write(bytestream);
 outStream.close();

%>
<%@ include file="../include/catch.jsp" %>
```

❸ 리스트에서 엑셀 다운로드를 클릭하여 다운로드 받은 엑셀 파일을 확인한다.



▶ POIExcel 받기 결과

POI의 결과를 보면 JExcelAPI의 결과와 다르다는 것을 알 수 있다. 두 가지 방법 중에 어떤 것이 좋을지는 사용자가 판단하여 사용한다.

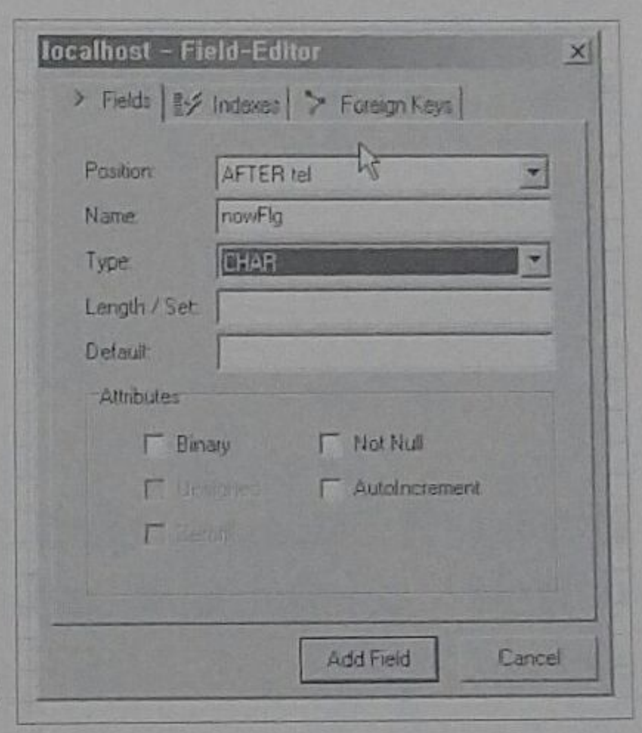
체크박스를 이용한 리스트 컨트롤

실무에서 많이 사용되는 방법으로 리스트 앞에 체크 박스를 두어 삭제하거나 또는 선택한 데이터에 다른 작업을 할 때 많이 사용되는 방법이다. 만약에 30건의 데이터가 있다면 30건의 데이터 중에 10건 A, 10건은 B, 10건은 C라고 주어야 한다면 일일이 해당 데이터를 클릭하여 값을 지정하거나 또는 데이터베이스의 쿼리 조작을 통해 값을 변경해 주어야 한다.

이런 방법은 업무의 효율성을 저하시킨다. 따라서 데이터의 리스트를 화면에 출력하고 사용자가 원하는 데이터의 값을 한 번에 처리하는 방법이 필요하다. 예제를 통해서 그 방법을 알아보기로 하자.

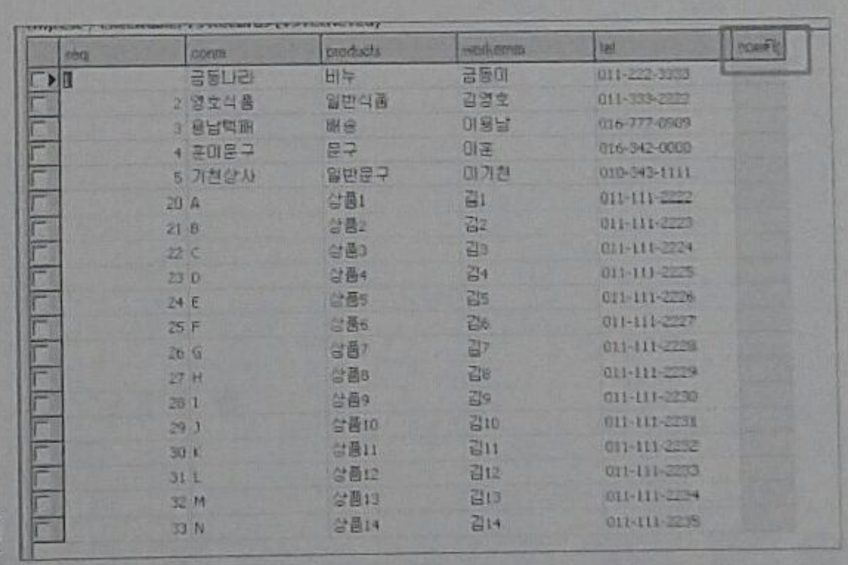
따라하기 체크박스를 이용한 리스트 컨트롤

❶ 엑셀파일 업로드 예제에서 사용된 exceltable 테이블에서 플래그 값을 저장하는 nowFlg의 칼럼을 추가한다.



▶ 칼럼의 추가

❷ 데이터베이스를 확인하면 새로운 칼럼 nowFlg가 추가된 것을 확인한다.



▶ 칼럼 추가 확인

❸ 리스트를 출력해주는 excelList.jsp에서 리스트의 앞에 체크박스를 생성한다. 생성 시에 hidden값으로 리스트의 seq값을 넣는다. 왜냐하면 체크박스를 선택 시 해당 seq의 값을 가져오기 위해서이다.

```
<table border="1">
    <tr>
        <td width="60" align="center">
            <font size="2"> </font>
        </td>
        <td width="60" align="center">
            <font size="2">번호</font>
        </td>
```

```
>
<td width="60">
    <p align="center">&lt;input type="checkbox" name="formCheck"&gt;</p>
    <input type="hidden" name="seq" value="<%=rs.getString("seq")%>" >
</td>
<td width="60">
```

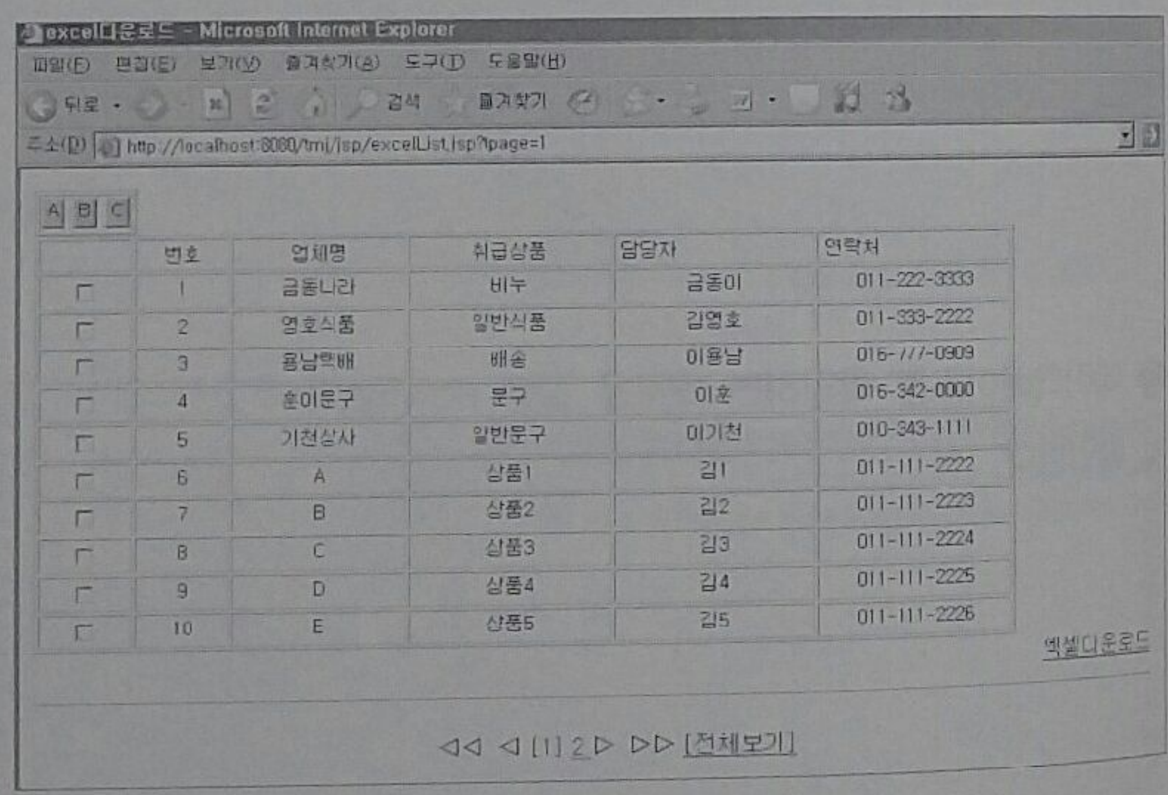
▶ 체크박스의 생성

❹ A, B, C를 클릭하기 위한 버튼을 추가한다.

```
<table border="1">
<tr>
    <td align="left">
        <input type="button" name="A" value="A" onClick="go_valueA()">
        <input type="button" name="B" value="B" onClick="go_valueB()">
        <input type="button" name="C" value="C" onClick="go_valueC()">
    </td>
</tr>
</table>
<table border="1">
    <tr>
        <td width="60" align="center">
            <font size="2"> </font>
```

▶ 버튼의 생성

❺ 브라우저를 열고 excelList.jsp를 호출하면 체크박스 및 버튼이 생성된 화면이 출력된다.



▶ 생성확인

❻ 현재 보이는 리스트에서 num의 최종 값을 form의 아래에 hidden 값으로 준다. num의 값을 주는 것은 스크립트 문에서 해당 리스트의 개수만큼 for문을 돌면서 체크된 값을 찾기 위한 것이다.

```
<table border="0" width="800">
<tr><td align="right">&lt;a href="#" onClick="downExcel(
</table>
<table border="0" width="800">
<tr><td align="center">&lt;%@ include file="../include/p
</table>
<input type="hidden" name="cnt" value="<%=num%>">
</form>
</body>
</html>
<%@ include file="../include/catch.jsp" %>
```

▶ 리스트의 num값 hidden처리

❼ 리스트에서 체크된 리스트 데이터에 A값을 저장하는 go_valueA() 스크립트를 작성한다.

```
//체크된 데이터를 받기위한 배열변수
  var data =[] ;
//체크박스의 체크여부를 카운터 하는 변수
var count=0;
//체크박스가 체크가 되었는지를 체크하는 변수
var incnt=0;
 function go_valueA()
{
 var TheForm = document.excelForm;
 //현재 리스트의 카운터
 var cnt = TheForm.cnt.value;
 //값 대입이 제대로 되었으면 true로 아니면 flase
 var check = false;
 //만약 데이터가 하나 있을 경우에 처리한다.
 //처리하지 않을 시 스크립트 에러
 if(cnt == 1)
{
  //체크박스가 참이면 그 값을 변수 data에 대입한다.
  //값을 대입시 데이터 처리를 위해 구분자 @을 첨부한다.
  //count를 증가시키는 것은 화면에서 체크박스가 체크되었는지 확인
  if(TheForm.formCheck.checked == true )
   {
     data=TheForm.seq.value+"@A";
     count++;
    }
 }
  //데이터의 리스트가 하나 이상일 경우
 else
 {
 //현재 리스트의 총 개수만큼 반복문을 통해 돌면서
 //체크박스가 참인 데이터를 변수 data에 대입한다.
 //form에 같은 이름의 오브젝트가 있다면 배열의 순서대로
 //0부터 n가지의 이름을 갖는다.
 //예로 seq가 5개가 같은 이름으로 있다면
 //seq[0], seq[1]...seq[4]란 이름을 갖는다.
 //cnt의 값에 -1을 해 주어야 한다.
 //num의 초기 값이 1부터이기 때문이다.
 //배열은 항상 0부터 시작되기 때문이다.
for(var i =0 ; i<(cnt-1);i++)
{
  if(TheForm.formCheck[i].checked ==true)
  {
   data[i]=TheForm.seq[i].value+"@A";
   count++;
   check=true;
   }
  }
 }

//체크된 값이 있어 check가 true일 때 실행
if(check)
{
 //count의 값이 0이거나 0보다 작다면 선택한 데이터가 없음으로 메시지처리

   if(count <= 0)
```

다음 페이지에 계속 ➲

```
{
  alert('한개 이상의 데이터를 선택 하세요');
}
else
{
  //대입받는 data의 변수를 form의 hidden으로 선언된
  //data, nowFlg의 폼 필드에 대입한다.
  //count의 값 또한 updataFlg에서 처리를 위한 값으로 count 폼 필드에 대입한다.
  //post 방식으로 값을 넘겨주기 위해서이다.
 TheForm.data.value=data;
  TheForm.count.value=count;
  TheForm.action="./updataFlg.jsp";
  TheForm.submit();
 }
}
 else
{
alert("Flg를 처리할 데이터를 체크해야 합니다.");
}
```

❽ form에 data와 count의 값을 받는 hidden 폼 필드를 생성한다.

```
8  </head>
9  <body >
0  <form name="excelForm" method="post">
1  <input type="hidden" name="data" value="">
2  <input type="hidden" name="count" value="">
3  <table border="1">
4  <tr>
```

▶ 폼 필드생성

❾ 체크된 리스트에서 data,nowFlg,count의 값을 받아 업데이트하는 updataFlg.jsp를 작성한다.

▶ updataFlg.jsp

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                        javax.naming.*,javax.sql.DataSource,KumdongUtil.*" %>
<%
Connection con=null;

//트랜잭션 처리와 빠른 insert를 위한 PreparedStatement 선언
PreparedStatement pstmt= null;

//전달받은 파라메타를 변수에 대입한다.
//KumdongUtil.nullcheck은 null값을 체크해 주는 function으로
//부록의 Util을 참고하기 바란다.
String data =KumdongUtil.nullcheck(request.getParameter("data"));
int cnt =Integer.parseInt(request.getParameter("count"));

String upSql="";

//seq를 받기위한 배열 변수선언
String [] upseq = null;
//Flg를 받기 위한 배열 변수선언
String [] nowFlg = null;

//트랜잭션 처리를 위한 변수선언
```

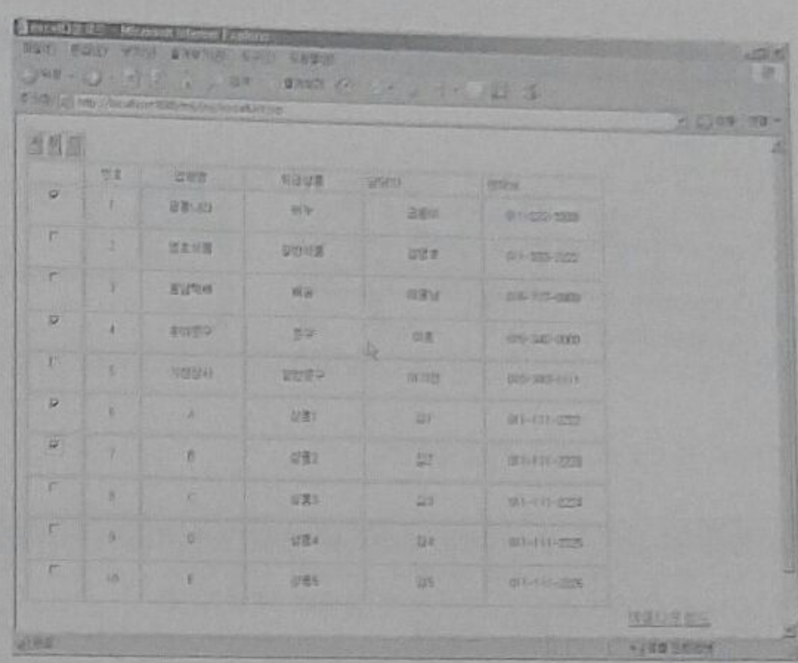
다음 페이지에 계속 ➲

```
int[] trans = null;
boolean commit_flag= true;

try
{
    //트랜잭션 처리를 위해 바로 선언
    Context initCtx = new InitialContext();
    Context envCtx = (Context) initCtx.lookup("java:comp/env");
    DataSource ds = (DataSource) envCtx.lookup("jdbc/tmjtest");
     con = ds.getConnection();
    //트랜잭션 처리를 위해 commit를 false를 반드시 먼저 해야 한다.
    con.setAutoCommit(false);

   //data로 받는 값들이 , 구분자를 자동으로 가지고 전달되기 때문에
   //StringTokenizer을 이용하여 , 기준으로 잘라 내어 값을 대입한다.
   //cnt를 받은 것은 선택된 데이터의 개수만큼 배열의 길이를 선언하기 위해서이다.
   //트랜잭션 처리를 위한 변수 trans도 같은 배열의 길이를 준다.
    StringTokenizer st = new StringTokenizer(data, ",");
    String [] piece = new String[cnt];
    upseq = new String[cnt];
    nowFlg=new String[cnt];
    trans  = new int[cnt];

    // ,을 기준으로 잘라진 데이터를 nextToken()을 이용하여
    //piece 변수의 배열에 저장한다.
    int i = 0;
    while(st.hasMoreTokens())
    {
      piece[i] = st.nextToken();
      i++;
    }

   //cnt(선택한 데이터의 개수) 만큼의 반복문
     for(int k =0;k<cnt;k++)
    {
      //구분자(,)로 받은 데이터를 다시 @으로 구분하여 Token하고
      //그 값을 upseq에 차례로 대입한다.
      st = new StringTokenizer(piece[k] , "@");
      int aaa = st.countTokens();
      upseq[k]=st.nextToken();
      nowFlg[k]=st.nextToken();
      try
      {
       //쿼리를 생성하고 prepareStatement를 한다.
       //이 때 tans에 executeUpdate의 리턴 값을 대입한다.
       upSql="update exceltable set nowFlg ='"+nowFlg[k]+"' ";
       upSql+="where seq ='"+upseq[k]+"' ";
       pstmt= con.prepareStatement(upSql);
       trans[k]= pstmt.executeUpdate();
      }
      catch(Exception ee)
     {}
    }//for close
    pstmt.close();
   }catch(NamingException ne){
       out.println(ne.toString());
```

다음 페이지에 계속

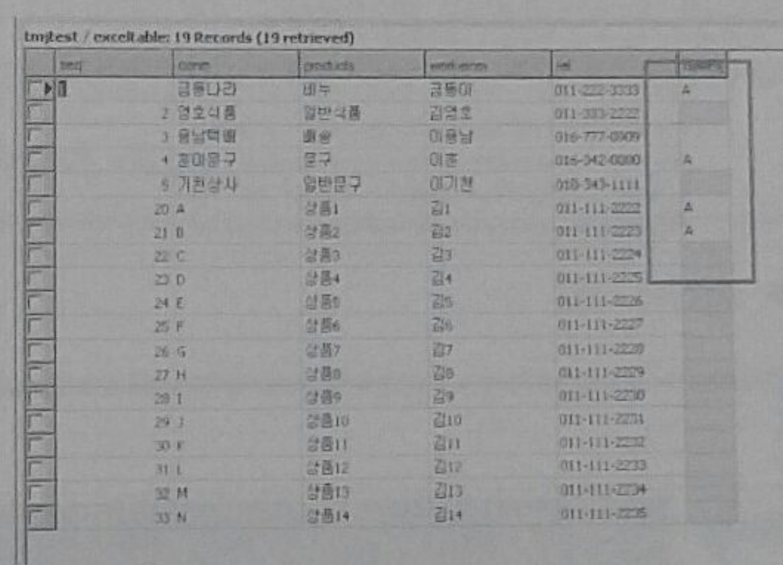
```
 }catch(SQLException se){
      out.println(se.toString());
  //에러가 발생하면 trans에 -1 대입
      trans[0] = -1;
 }catch(Exception e){
     out.print(e);
    //에러가 발생하면 trans에 -1 대입
    trans[0] = -1;
}finally{
        try { if (pstmt != null) pstmt.close(); } catch (Exception e) {}
}

try{
     //선택한 데이터의 개수 cnt만큼 반복을 통해
     //trans가 -1 의 값이 하나라도 있으면
     //commit_flag에 flase값을 주고 반복문을 바로 빠져나온다.
     for(int i=0; i< cnt; i++){
      if( trans[i] == -1  ){
            commit_flag = false;
            break;
        }
      }

    //commit_flag의 값이 true이면 데이터베이스에 저장한다.
     if( commit_flag ){
         con.commit();
         con.setAutoCommit(true);
         con.close();
%>
<form name="formm" method="post" action="./excelList.jsp">
</form>
<script language="javascript">
    alert('정상적으로 처리 되었습니다.');
    document.formm.submit();
</script>
<%
         }else{
         con.rollback();
         con.close();
%>
<form name="formm" method="post" action="./excelList.jsp">
</form>
<script language="javascript">
   alert('오류가 발생되었습니다.');
   document.formm.submit();
</script>
<%
    }
 }
catch(SQLException se)
{
         out.println(se);
}
%>
```

⑩ 브라우저를 열고 excelList.jsp를 호출한 후 리스트에서 몇 개의 체크박스를 선택한다.



▶ 데이터의 선택

⑪ 체크박스를 선택한 후 A 버튼을 클릭한 후 데이터베이스에서 exceltable의 데이터를 확인하면 Flg가 변경된 것을 확인한다.



▶ 데이터의 확인

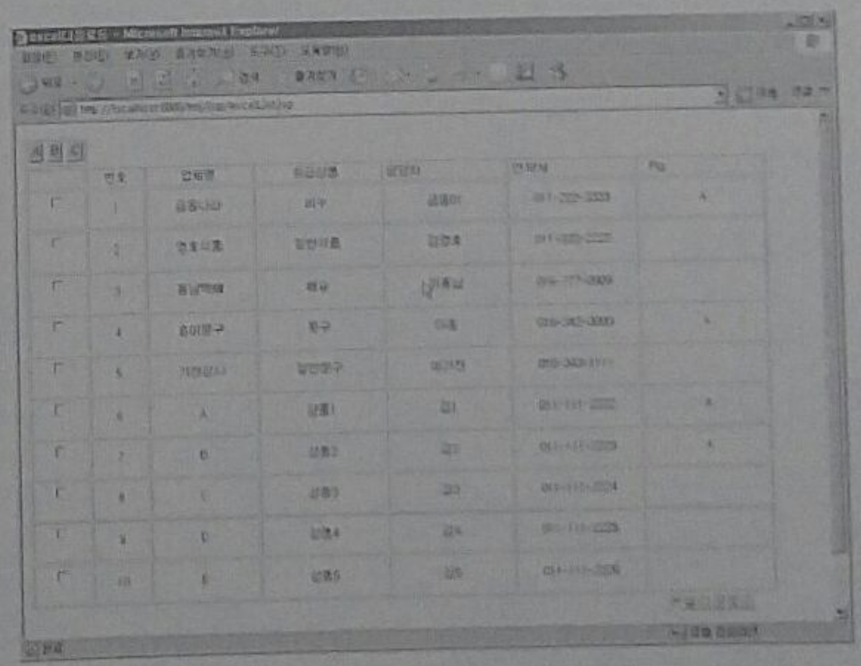
⑫ 정상적인 Flg값의 변경 후 출력되는 리스트에서 추가된 Flg에 대한 화면출력 필드를 excelList.jsp에 추가한다.

```
        </td>
        <td width="45" align="center">
          <font size="2">여락처</font>
        </td>
        <td width="45" align="center">
          <font size="2">Flg</font>
        </td>
    </tr>
<!--while문을 통해 리스트를 출력한다.-->
<%
```

```
        <td  width="135">
          <p align="center"><font size="2"> <font size="2"><%=KumdongUtil.nullcheck(rs.getString("nowFlg"))%></font></font></p>
        </td>
    </tr>
<%
    //번호에 대한 증가값
    num++;
    }
```

▶ Flg 출력 추가

⑬ excelList.jsp를 다시 호출하면 추가된 Flg의 값이 출력된다.



⑭ 같은 방법으로 스크립트에 B, C도 작성한다. 단지 data의 값에 B, C를 각각 대입하면 될 것이다.

```
//count를 증가시키는 것은 화면에서 체크박스가 체:
if(TheForm.formCheck.checked == true )
{
    data=TheForm.seq.value+"@B";
    count++;
    check=true;
}


)이터의 리스트가 하나 이상일경우
e

 //현재 리스트의 총갯수만큼 반복문을 통해 돌면서
 //체크박스가 참인데이터를 변수 data에 대입한다.
 //form에 같은이름의 오브젝트가 있다면 배열의 순
 //0부터n가지의 이름을 갖는다.
 //예로 seq가 5개가 같은 이름으로 있다면
 //seq[0],seq[1]...seq[4]란 이름을 갖는다.
 //cnt의 값에 -1을 해 주어야 한다.
 //num의 초기값이1부터이기 때문이다.
 //배열은 항상 0 부터 시작되기 때문이다.
 for(var i =0 ; i<(cnt-1);i++)
 {
     if(TheForm.formCheck[i].checked ==true)
     {
         data[i]=TheForm.seq[i].value+"@B";
         count++;
         check=true;
```

```
//count를 증가시키는 것은 화면에서 체크박스가 체크:
if(TheForm.formCheck.checked == true )
{
    data=TheForm.seq.value+"@C";
    count++;
    check=true;
}


/데이터의 리스트가 하나 이상일경우
lse

 //현재 리스트의 총갯수만큼 반복문을 통해 돌면서
 //체크박스가 참인데이터를 변수 data에 대입한다.
 //form에 같은이름의 오브젝트가 있다면 배열의 순서(
 //0부터n가지의 이름을 갖는다.
 //예로 seq가 5개가 같은 이름으로 있다면
 //seq[0],seq[1]...seq[4]란 이름을 갖는다.
 //cnt의 값에 -1을 해 주어야 한다.
 //num의 초기값이1부터이기 때문이다.
 //배열은 항상 0 부터 시작되기 때문이다.
 for(var i =0 ; i<(cnt-1);i++)
 {
     if(TheForm.formCheck[i].checked ==true)
     {
         data[i]=TheForm.seq[i].value+"@C";
         count++;
```

▶ flg값의 대입

맺음말

이번 장에서 우리는 엑셀파일의 데이터를 데이터베이스에 저장하고 ,가 저장된 데이터를 다시 엑셀파일로 다운로드 받는 예제를 공부했다. 또한 저장된 데이터를 출력하는 리스트에서 체크박스를 통해 컨트롤 하는 방법에 대해서도 공부했다. 업데이트만 예제로 다루었지만 저장, 수정, 삭제 등도 예제를 이용해 작성할 수 있다. 다음 장에서는 앞 장에서 배운 예제를 이용한 하나의 프로젝트를 진행하게 된다.

13 프로젝트 관리프로그램 [관리자]

안녕하세요

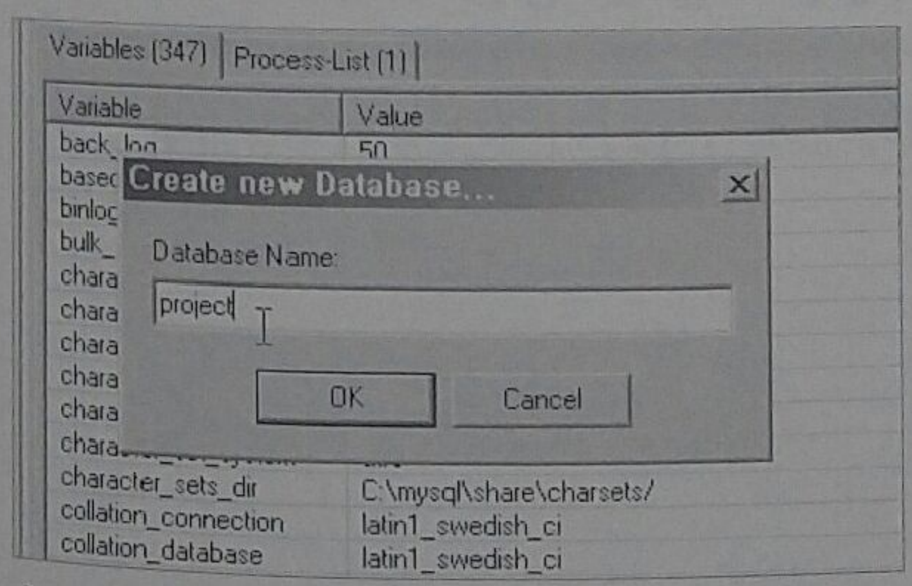
이번 장에서는 앞서 배운 예제를 통해 종합적이고 간단한 프로젝트 진행으로 한 단계 진보적인 발전을 한다. 이번 장에서 새롭게 배우게 될 것은 session의 사용법과 게시판의 생성, 프로젝트에 따라서 투입되는 인력 그리고 해당 인력에 대한 게시판의 권한 등에 대해서 공부하게 된다.

테이블의 생성 –관리자 모드

A라는 프로젝트를 수행한다면 A프로젝트에 대한 관리가 필요하다. 개발투입 인력관리 또는 개발자가 단위로 프로그램을 개발하고 난 후 제반문서 등등 기타 사항에 대한 관리가 필요할 것이다. 따라서 프로젝트의 관리 프로그램의 사용자는 개발자/관리자로 한정된다. 관리자의 입장에서는 프로젝트의 생성/개발인력에 대한 권한 설정, 해당 프로젝트에 해당하는 게시판을 생성이며 사용자는 사용자 등록 후 권한을 부여 받아 권한 있는 게시판을 이용한다.

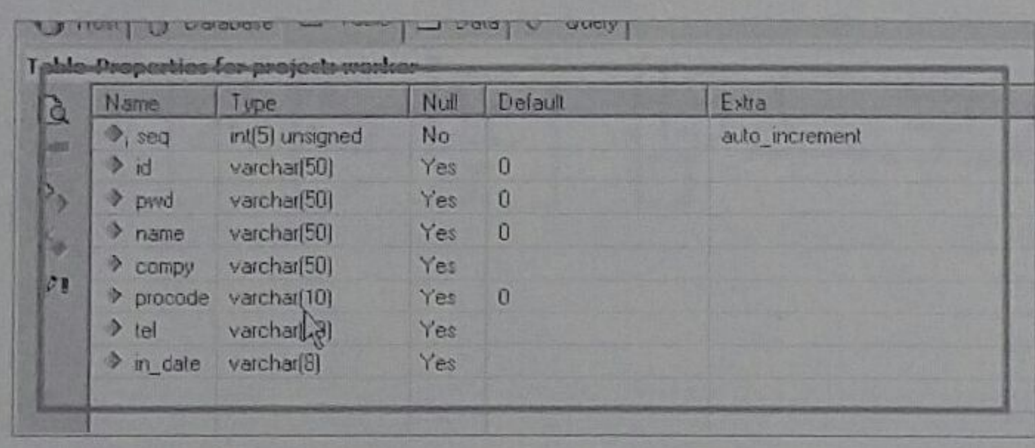
따라하기 테이블의 생성

❶ mysqlFont에서 프로젝트 관리프로그램의 데이터베이스를 생성한다.



▶ project 데이터베이스의 생성

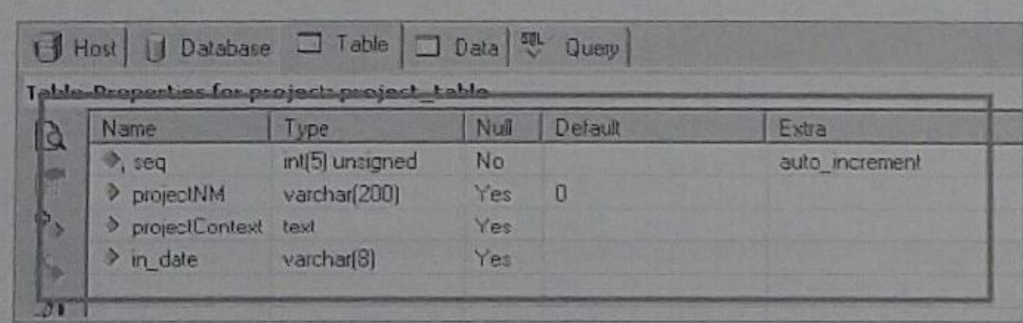
❷ 사용자의 정보를 보관하는 테이블을 생성한다.



Name	Type	Null	Default	Extra
seq	int(5) unsigned	No		auto_increment
id	varchar(50)	Yes	0	
pwd	varchar(50)	Yes	0	
name	varchar(50)	Yes	0	
compy	varchar(50)	Yes		
procode	varchar(10)	Yes	0	
tel	varchar(13)	Yes		
in_date	varchar(8)	Yes		

▶ worker 테이블의 생성

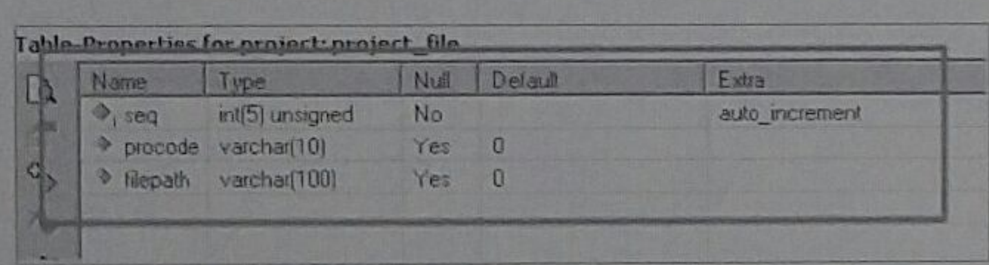
❸ 프로젝트 데이터를 저장하는 테이블을 생성한다.



Name	Type	Null	Default	Extra
seq	int(5) unsigned	No		auto_increment
projectNM	varchar(200)	Yes	0	
projectContext	text	Yes		
in_date	varchar(8)	Yes		

▶ project_table 테이블의 생성

❹ 해당 프로젝트의 파일 업로드 정보를 저장하는 테이블을 생성한다.



Name	Type	Null	Default	Extra
seq	int(5) unsigned	No		auto_increment
procode	varchar(10)	Yes	0	
filepath	varchar(100)	Yes	0	

▶ project_file 테이블의 생성

각각의 테이블을 살펴보면 다음과 같다.

worker 테이블

seq	일련번호
id	사용자 아이디
pw	사용자 암호
name	사용자 이름
compy	소속회사
tel	연락처
procode	프로젝트 코드
in_date	등록일

project_table

seq	일련번호
projectNM	프로젝트 이름
projectContext	프로젝트 내용
in_date	등록일

project_file

seq	일련번호
procode	프로젝트 코드
filepath	파일업로드 디렉토리

Worker 테이블에는 관리자가 초기에 등록되어야 한다. 왜냐하면 관리자 페이지에서 프로젝트를 생성하고 개발자에게 권한을 주기 위해서이다. 따라서 관리자의 아이디와 패스워드는 데이터베이스에서 sql문을 통해 입력 또는 mysql-Front에서 직접 입력한다.

Host | Database | Table | Data | Query

project / worker: 0 Records (1 retrieved)

seq	id	pwd	name	procode	in_date
1	admin	admin	관리자	0	20061001

▶ 데이터베이스에서의 관리자 등록

Context의 생성 및 폴더생성

10장에서 공부한 톰켓에서 Context를 생성하는 방법에 따라 projectPP라는 Context를 추가하고 저장한다.

❶ server.xml에 context추가

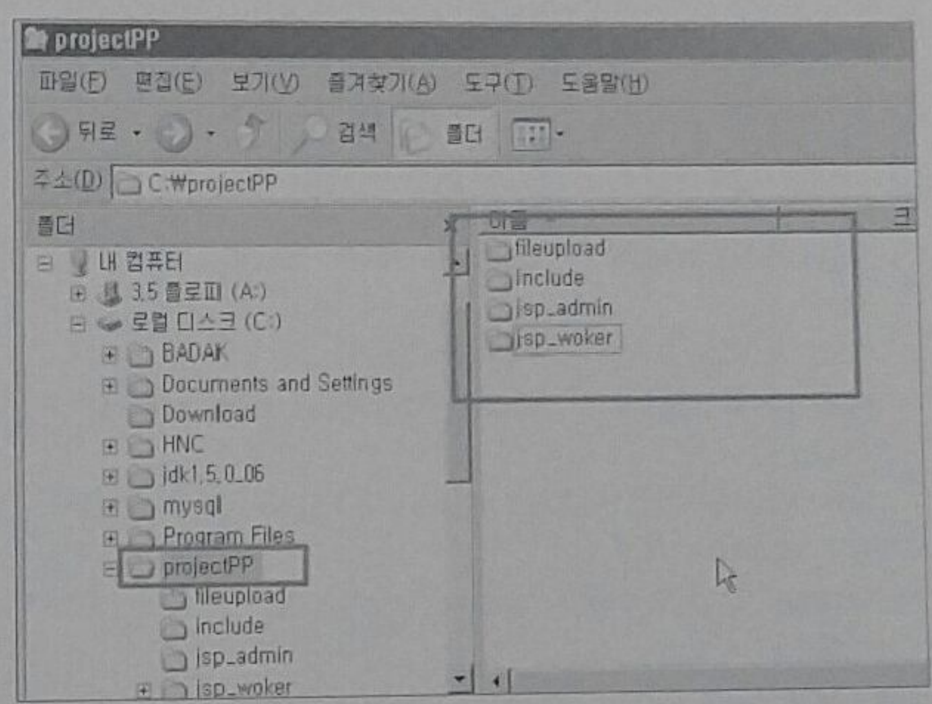
```
<!-- Tomcat projectPP Context -->
  <Context path="/projectPP" docBase="c:\projectPP" debug="0" reloadable="true">
   <Logger  className="org.apache.catalina.logger.FileLogger" prefix="tmjtest_log." suffix=".txt"
timestamp="true"/>
  <Resource name="jdbc/project" auth="Container" type="javax.sql.DataSource"/>
  <ResourceParams name="jdbc/project">
    <parameter>
      <name>username</name>
      <value>root</value>
    </parameter>
    <parameter>
      <name>password</name>
      <value></value>
    </parameter>
    <parameter>
      <name>driverClassName</name>
      <value>com.mysql.jdbc.Driver</value>
    </parameter>
    <parameter>
      <name>url</name>
      <value>jdbc:mysql://localhost:3306/project?autoReconnect=true</value>
    </parameter>
    <parameter>
      <name>removeAbandoned</name>
      <value>true</value>
    </parameter>
    <parameter>
      <name>removeAbandonedTimeout</name>
      <value>60</value>
    </parameter>
    <parameter>
      <name>logAbandoned</name>
      <value>true</value>
    </parameter>
    <parameter>
      <name>maxActive</name>
      <value>1000</value>
    </parameter>
    <parameter>
      <name>maxIdle</name>
      <value>10</value>
    </parameter>
    <parameter>
      <name>maxWait</name>
      <value>180</value>
    </parameter>
  </ResourceParams>
</Context>
```

❷ server.xml 파일 내에 context를 추가하였다면 web.xml 파일에도 생성한 project 데이터베이스의 이름을 정의한다.

```
    </resource-ref>

    <resource-ref>
        <description>mysql Datasource</description>
        <res-ref-name>jdbc/project</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <res-auth>Container</res-auth>
    </resource-ref>

</web-app>
```

▶ web.xml 추가

❸ 디렉토리에 projectPP라는 폴더를 만들고 필요한 폴더를 생성한다.



▶ 폴더의 생성과 서브폴더의 생성

앞으로 13장 14장의 예제는 생성된 projectPP라는 폴더에서 기본적으로 호출된다.

로그인 작성

기본적인 사항이 완료가 됐다면 관리자/사용자 로그인 메인페이지를 만들도록 한다. 메인에서 신규등록을 클릭하면 개발자가 직접 개발자의 정보를 등록한다. 따라서 신규등록은 사용자모드에서 추가로 설명한다.

따라하기 로그인 만들기

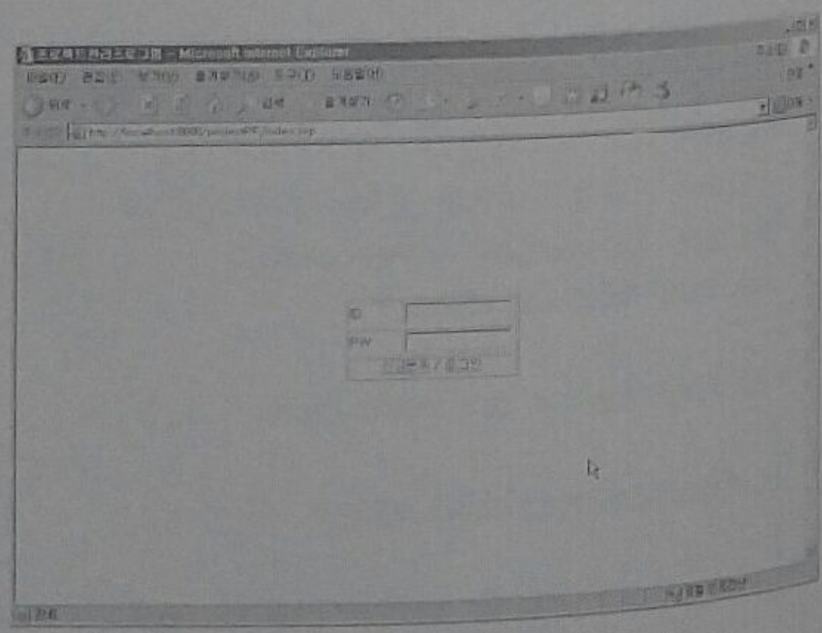
❶ 로그인을 하는 index.jsp 파일을 생성하고 projectPP 폴더에 저장한다.

```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<html>
<head>
<title>프로젝트 관리 프로그램</title>
<script language="javascript">
function worker_check()
{
var theForm  = document.frm;

if(theForm.workerId.value =='')
{
   alert('아이디를 입력하세요');
```

다음 페이지에 계속 ➲

```
}
else if(theForm.workerPw.value =='')
{
  alert('암호를 입력하세요');
}
else
{
  theForm.action="./workerCheck.jsp";
  theForm.submit();
 }
}
</script>
</head>
<body>
<form name="frm" method="post">
<table border="0" width="100%" height="368">
<tr>
<td width="100%" height="362">
<table border="1" width="180" align="center">
<tr>
<td width="52"><font size="2">ID</font></td>
<td width="112">
<input type="text" name="workerId" size="14">
</td>
</tr>
<tr>
<td width="52" height="11"><font size="2">PW</font></td>
<td width="112" height="23">
<input type="password" name="workerPw" size="14">
</td>
</tr>
<tr>
<td width="170" height="11" colspan="2" align="center">
<font size="2">
<a href="#">신규등록</a> /
<a href="#" onClick="worker_check()">로그인</a>
</font></td>
</tr>
</table>
</td>
</tr>
</table>
<form>
</body>
</html>
```

브라우저를 열고 projectPP 폴더에 있는 index.jsp을 호출한다.



▶ index.jsp의 호출

❷ 호출된 index.jsp에서 아이디/패스워드를 입력받아 아이디/패스워드를 체크하는 workerCheck.jsp 파일을 작성하고 projectPP 폴더에 저장한다.

▶ [workerCheck.jsp]

```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<%@ page  import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%
  //입력되는 아이디와 패스워드를 받는 파라메타 변수
  String workId = request.getParameter("workerId");
  String workPw = request.getParameter("workerPw");

//아이디 패스워드를 체크 쿼리
String checksql="select count(*) from worker where  id='"+workId+"' and pwd='"+workPw+"'";

//아이디를 가져오는 쿼리
String sql_id ="select id from worker where  id='"+workId+"' and pwd='"+workPw+"'";

//아이디와 패스워드를 체크 변수
int cnt001=0;

//아이디를 저장하는 변수
String reg_id="";
%>
<%@ include file="./include/try.jsp" %>
<%
 //아이디 패스워드를 체크, count의 값을 cnt001 변수에 대입한다.
  rs =stmt.executeQuery(checksql);
      if(rs.next())
      {
              cnt001=rs.getInt(1);
      }
      rs.close();

     //cnt001 변수값에 따라서 아이디를 가져오는 쿼리를 실행한다.
      if( cnt001 > 0)
      {
              rs = stmt.executeQuery(sql_id);
               if(rs.next())
               {
                  reg_id = rs.getString(1);
               }
               rs.close();

      }

%>
<%@ include file="./include/catch.jsp" %>
<%
//사용자이고 관리자이면 아이디를 세션으로 저장한다.
if( cnt001 > 0 && "admin".equals(reg_id) )
{
   session.setAttribute("id",reg_id);

%>
<form name="frm" method="post" action="./jsp_admin/main.jsp">
</form>
```

다음 페이지에 계속 ➲

```
<script language="javascript">
   document.frm.submit();
</script>
<%
}
//사용자가 아니면 메시지 출력
else if( cnt001 ==0 && "".equals(reg_id))
{
%>
<form name="frm" method="post" action="./index.jsp">
</form>
<script language="javascript">
        alert('아이디 패스워드를 확인 하세요');
      document.frm.submit();
</script>
<%
}
//사용자이고 관리자가 아니면 아이디를 세션으로 저장한다.
else if( cnt001 > 0 && !"admin".equals(reg_id))
{
   session.setAttribute("id",reg_id);
%>
<form name="frm" method="post" action="./jsp_worker/main.jsp">
</form>
<script language="javascript">
        document.frm.submit();
</script>
<%
}
%>
```

❸ 10장 예제에서 만들어 두었던 try.jsp 파일과 catch.jsp 파일을 include에 복사하고 데이터베이스에 접속하기 위한 이름을 변경한다.

```
try {
    Context initCtx = new InitialContext();
    Context envCtx = (Context) initCtx.lookup("java:comp/env");
    DataSource ds = (DataSource) envCtx.lookup("jdbc/project");

    con = ds.getConnection();
    stmt = con.createStatement();
%>
```

▶ try.jsp 파일의 수정

❹ 관리자 확인 후 호출되는 main.jsp 파일을 작성한다. main.jsp는 top.jsp, left.jsp, body.jsp 파일로 만들어진 프레임으로 구성이 된다. top.jsp에는 이미지 또는 텍스트가 존재할 수 있고 left.jsp에는 메뉴가 존재, body.jsp에는 해당 메뉴를 클릭 시 리스트 또는 기타 제반 사항들이 출력된다. 도서출판 대림 홈페이지 자료실에서 다운받은 파일을 jsp_admin 폴더에 저장 또는 새롭게 작성한다.

▶ [main.jsp]

```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<html>
<head>
<title>프로젝트관리프로그램-관리자</title>
</head>
<frameset rows="17%, 83%" cols="1*">
<frame name="banner" scrolling="no" marginwidth="10" marginheight="14" namo_target_frame=
"contents" src="./top.jsp">
<frameset rows="1*" cols="20%, 80%">
```

다음 페이지에 계속

```
<frame name="contents" scrolling="no" marginwidth="10" marginheight="14" namo_target_frame=
"detail" src="./left.jsp">
<frame name="detail" scrolling="yes" marginwidth="10" marginheight="14" src="./body.jsp">
</frameset>
<noframes>
<body bgcolor="#FFFFFF" text="#000000" link="#0000FF" vlink="#800080" alink="#FF0000">
<p>이 페이지를 보려면, 프레임을 볼 수 있는 브라우저가 필요합니다.</p>
</body>
</noframes>
</frameset>
</html>
```

▶ [top.jsp]

```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<title>프로젝트 관리 프로그램</title>
<meta name="generator" content="Namo WebEditor">
<base target="contents"></head>
<body bgcolor="white" text="black" link="blue" vlink="purple" alink="red">
<p><font size="2">프로젝트 관리 프로그램</font></p>
</body>
</html>
```

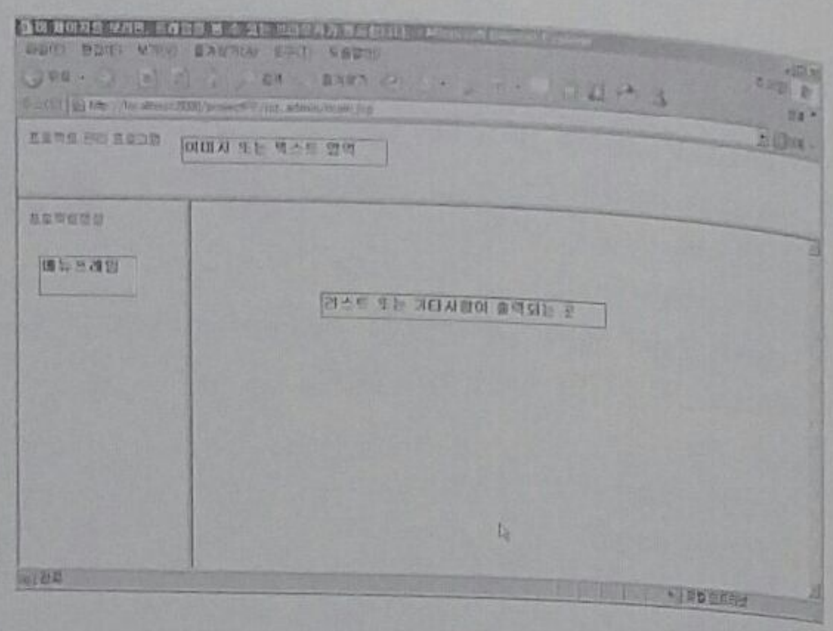
▶ [left.jsp]

```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<html>
<head>
<title>프로젝트생성</title>
<meta name="generator" content="Namo WebEditor">
<base target="detail"></head>
<body bgcolor="white" text="black" link="blue" vlink="purple" alink="red">
<p><font size="2">프로젝트생성</font></p>
</body>
</html>
```

▶ [body.jsp]

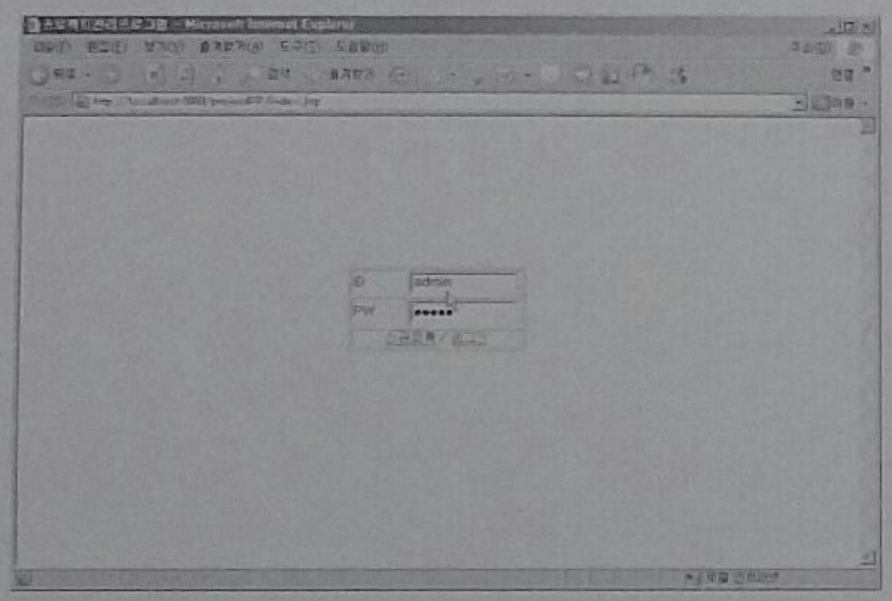
```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<html>
<head>
<title>제목 없음</title>
<meta name="generator" content="Namo WebEditor">
</head>
<body>
</body>
</html>
```

브라우저를 열어 main.jsp를 호출한다.

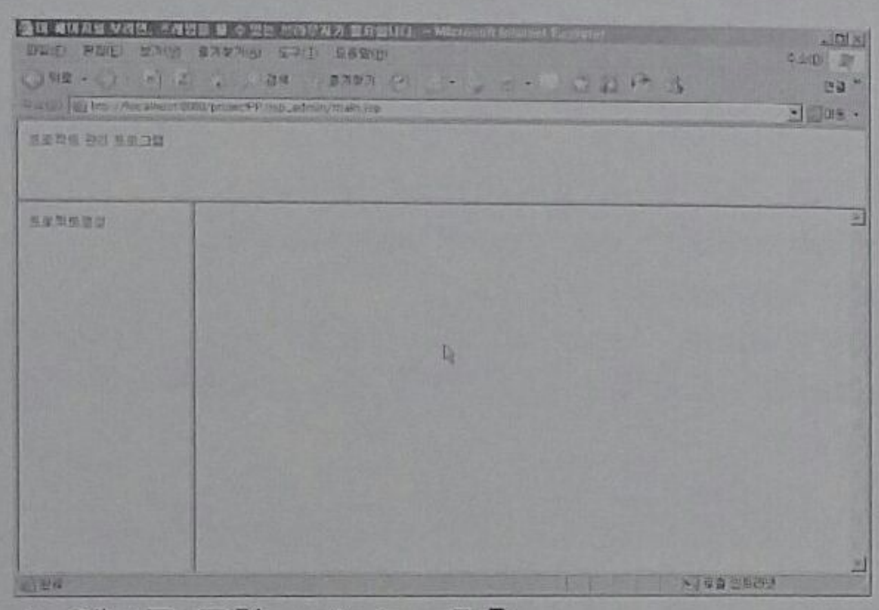


▶ main.jsp 호출과 그에 따른 프레임의 역할

❺ 브라우저에서 로그인 되는 index.jsp을 호출하고 데이터베이스에 저장되어 있는 관리자의 아이디와 암호를 입력, 로그인을 클릭하면 main.jsp가 출력된다.



▶ index.jsp 호출



▶ 체크를 통한 main.jsp 호출

로그인을 하여 main.jsp을 호출하였다면 이제 잠시 다른 문제를 생각해 볼 때이다. 만약 사용자 체크를 이용하지 않고 누군가가 의도적으로 브라우저에서 직접 main.jsp을 호출한다면 아주 커다란 문제가 발생한다. 따라서 그러한 문제를 해결하기 위해 세션(session)이라는 것을 사용하게 된다.

일정한 시간동안 로그인 된 사용자의 특정정보를 암호화하여 세션(session)으로 저장시키고 저장된 세션(session)이 브라우저의 종료와 함께 끊어지게 되어 의도적인 접근을 막는 역할을 한다. 세션(session)은 사용자의 접속유무 체크뿐만 아니라 쇼핑몰의 장바구니에서도 사용된다.

세션 저장하기

세션저장하기

❶ 작성된 workerCheck.jsp 파일에서 관리자 또는 사용자가 맞는 경우 세션을 저장하는 부분을 추가한다.

```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>

<%
 //입력되는 아이디와 패스워드를 받는 변수
 String workId = request.getParameter("workerId");
 String workPw = request.getParameter("workerPw");

//아이디 패스워드 체크 쿼리
String checksql="select count(*) from worker where  id='"+workId+"' and pwd='"+workPw+"'";

//아이디, 프로젝트 코드를 가져오는 쿼리
String sql_id ="select id,procode from worker where  id='"+workId+"' and pwd='"+workPw+"'";

//아이디와 패스워드를 체크하기 위한 변수
int cnt001=0;

//아이디를 저장하는 변수
String reg_id="";
String procode="";
%>
<%@ include file="./include/try.jsp" %>
<%
    rs =stmt.executeQuery(checksql);
        if(rs.next())
        {
                cnt001=rs.getInt(1);
        }
        rs.close();

        if( cnt001 > 0)
        {
                rs = stmt.executeQuery(sql_id);

                if(rs.next())
                {
                        reg_id = rs.getString(1);
                        procode = rs.getString(2);
                }
                rs.close();
        }
%>
<%@ include file="./include/catch.jsp" %>
<%
//사용자이고 관리자이면 setAttribute()을 이용하여 세션저장
if( cnt001 > 0 && "admin".equals(reg_id) )
{
   session.setAttribute("id",reg_id);
%>
<form name="frm" method="post" action="./jsp_admin/main.jsp">
</form>
<script language="javascript">
 document.frm.submit();
</script>
<%
}
//사용자가 아니면 메시지 출력
```

다음 페이지에 계속

```
else if( cnt001 ==0 && "".equals(reg_id))
{
%>
<form name="frm" method="post" action="./index.jsp">
</form>
<script language="javascript">
        alert('아이디 패스워드를 확인하세요');
    document.frm.submit();
</script>
<%
}
//사용자이고 관리자가 아니면 메시지 출력
// 사용자이면 사용자 아이디와 프로젝트의 코드를 세션 저장
else if( cnt001 > 0 && !"admin".equals(reg_id) )
{
  if("0".equals(procode) || "".equals(procode))
        {
%>
<form name="frm" method="post" action="./index.jsp">
</form>
<script language="javascript">
        alert('투입되어진 프로젝트가 없습니다. 관리자에게 문의하세요');
        document.frm.submit();
</script>
<%
        }
    else
        {
    session.setAttribute("id",reg_id);
    session.setAttribute("procode",procode);
%>
<form name="frm" method="post" action="./jsp_worker/main.jsp">
</form>
<script language="javascript">
        document.frm.submit();
</script>
<%
        }
}
%>
```

추가된 부분은 아이디를 가져오는 쿼리와 세션으로 저장하는 session.setAttribute이다. 이렇게 로그인시 저장된 세션은 각 프로그램의 상단에 include되어 로그인을 체크하게 된다.

❷ 저장된 세션(session)을 체크하는 sessionChk.jsp를 작성하고 include 폴더에 저장한다.

▶ [sessionChk.jsp]

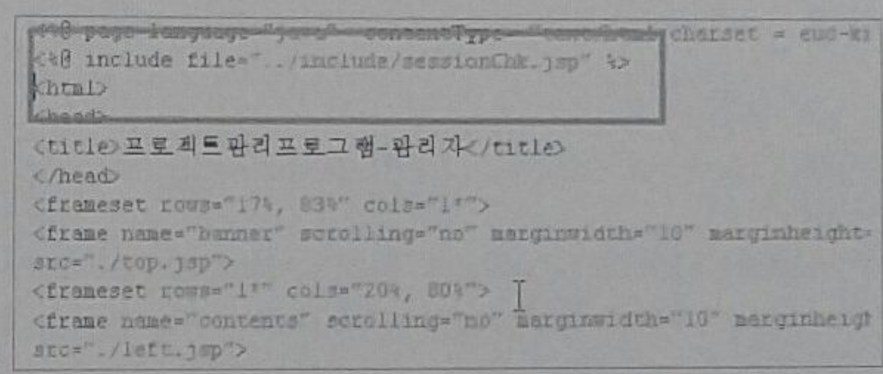
```
<%
String session_id = (String)session.getAttribute("id");

 if("".equals(session_id) || session_id == null )
   {
%>
<form name="okForm" method="post">
</form>
```

다음 페이지에 계속 ➲

```
<script language="javascript">
        alert('접속이 되지 않았습니다. 로그인을 다시하세요');
    top.location.href='../index.jsp';
</script>
<%
  }
%>
```

String session_id = (String)session.getAttribute("id");이 저장된 세션의 정보를 가져온다. id뿐만 아니라 workerCheck.jsp에서 필요한 항목을 쿼리하여 setAttribute에 저장하고 getAttribute를 사용하여 세션(session)의 정보를 이용한다.

❸ 작성된 sessionChk.jsp 파일을 main.jsp 파일의 상단에 include한다.

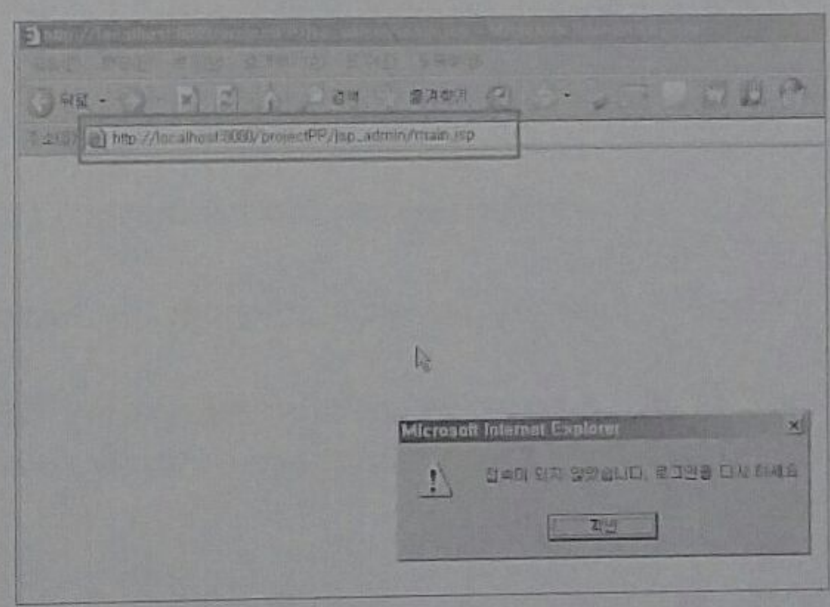


▶ sessionChk.jsp 파일의 include

앞 예제에서도 이야기했지만 include 폴더에 존재하는 파일은 공통으로 사용되는 모듈을 저장하는 곳이다. 따라서 세션(session)을 체크하는 seesionChk.jsp 파일도 공통으로 사용되기 때문에 이곳에 저장한다.

❹ 브라우저에서 로그인을 통하지 않고 main.jsp을 호출하면 세션(session)이 존재하지 않기 때문에 에러 메시지를 출력한다.

정상적인 로그인을 통하면 관리자의 main.jsp가 정상적으로 호출된다.



▶ 의도적인 주소접근 시

프로젝트 리스트 만들기

관리자가 메뉴에서 프로젝트 생성을 클릭하면 현재 저장되어 있는 프로젝트 리스트가 출력이 되고 리스트 아래 새로운 프로젝트를 생성하는 버튼을 만든다.

따라하기 프로젝트 리스트

❶ 저장된 프로젝트 리스트를 가져오는 project_List.jsp을 작성한다.

▶ [project_List.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                    javax.naming.*,javax.sql.DataSource" %>
<%!
//null 처리를 위한 function
public static String nullcheck(String str) throws Exception {

if (str == null){
return "";
}else{
return str;      // 넘어온 Stirng을 그대로 다시 return
}
}

%>
<!--세션체크-->
<%@ include file="../include/sessionChk.jsp" %>
<%
String listSql="select seq,projectNM,in_date from project_table";
String pageCountQuery = "select count(seq) from project_table ";

// 페이지 설정을 위한 변수
int view_rows =10;                          //한 페이지에 출력되는 레코드의 개수
int total_pages =1;                       //전체 페이지 (default 1)
int tpage=1;                              //현재 페이지 (default 1)

//페이지 이동시 사용되는 변수선언
String url = "";
url = "./project_List.jsp?tpage=";

if (!"ok".equals(request.getParameter("fullview")) && request.getParameter("tpage") != null)
//전체보기 확인
tpage = Integer.parseInt(request.getParameter("tpage"));

if (!"ok".equals(request.getParameter("fullview")))
listSql += " limit "+((tpage-1)*view_rows)+","+view_rows;

%>
<%@ include file="../include/try.jsp" %>
<%
rs = stmt.executeQuery(listSql);
%>

<html>
<head>
<title>프로젝트 리스트</title>
<script language="javascript">
  function go_project()
  {
          var theForm = document.frm;
          theForm.action="./project_Crt.jsp";
          theForm.submit();
  }
</script>
</head>
<body>
<form name="frm" method="post">
```

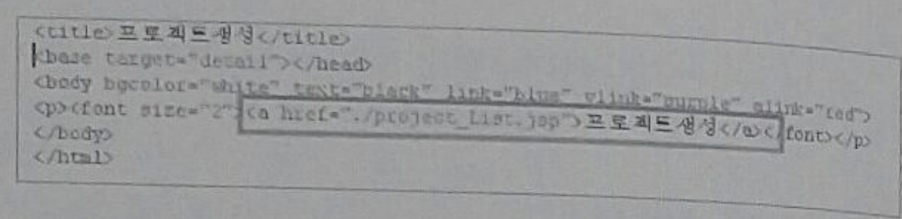
다음 페이지에 계속 ➲

```
<table border="1" width="524">
<tr>
        <td width="98"><font size="2">프로젝트번호</font></td>
        <td width="293"><font size="2">프로젝트명</font></td>
        <td width="111"><font size="2">등록일</font></td>
</tr>
<!--while문을 통해 리스트를 출력한다.-->
<%
int num=1;

 if (!"ok".equals(request.getParameter("fullview")))
   num = ((tpage-1)*view_rows)+1;

 while(rs.next())
 {

 %>
 <tr>
        <td width="98" height="11">
              <font size="2"><%=nullcheck(rs.getString(1))%></font>
       </td>
        <td width="293" height="23">
             <font size="2"><%=nullcheck(rs.getString(2))%></font>
       </td>
        <td width="111" height="23">
             <font size="2"><%=nullcheck(rs.getString(3))%></font>
      </td>
</tr>
<%
//번호에 대한 증가값
num++;
}
if(num ==1)
{
%>
<tr>
      <td colspan=3 align=center>
         <font size="2">해당하는 데이터가 없습니다.</font>
     </td>
</tr>
<%
}
%>
</table>
<table border="0" width="524">
<tr>
<td align="center">
        <!--11장에서 만든 페이지 나누기 pageCount.jsp 파일을
                projectPP/include 파일에 복사하여 바로 사용한다.-->
        <%@ include file="../include/pageCount.jsp"%>
</td>
</tr>
</table>
<%@ include file="../include/catch.jsp" %>
<table border="0" width="524">
<tr>
  <td width="524">
```

다음 페이지에 계속

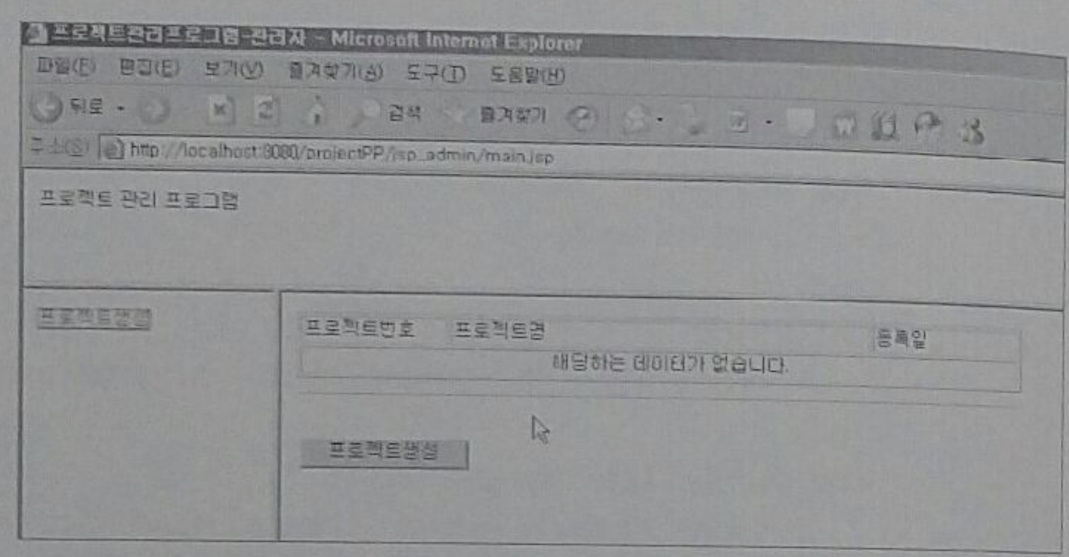
```
<input type="button" name="btn_save" value="프로젝트생성" onClick="go_project()">
</td>
</tr>
</table>
</form>
</body>
</html>
```

❷ 이미 작성된 left.jsp 파일에서 project_List.jsp를 호출하는 링크를 작성한다.



▶ 링크작업

❸ 관리자로 로그인하여 left 메뉴에서 프로젝트 생성을 클릭하면 프로젝트 리스트가 출력이 된다.



▶ 프로젝트 리스트

project_List.jsp 파일을 살펴보면 앞서 배운 예제에서 크게 벗어나지 않는다. 즉 페이지 나누기를 담당하는 pageCount.jsp 파일을 앞에서 예제로 사용했던 tmj 폴더에서 복사하여 projectPP/include에 덧붙여 그대로 사용했다. 누누이 이야기하지만 공통모듈을 만들면 그걸 공통적으로 사용하는 것이 바람직하다.

프로젝트 생성

리스트가 만들어 졌다면 리스트에서 프로젝트 생성 버튼을 클릭하여 생성되는 프로젝트의 입력값을 받는 화면을 출력하고 입력된 데이터를 저장하는 프로그램을 작성한다.

따라하기 프로젝트 입력화면

❶ project_List.jsp에서 정의된 스크립트 function go_project()에서 호출되는 project_Crt.jsp를 작성한다.

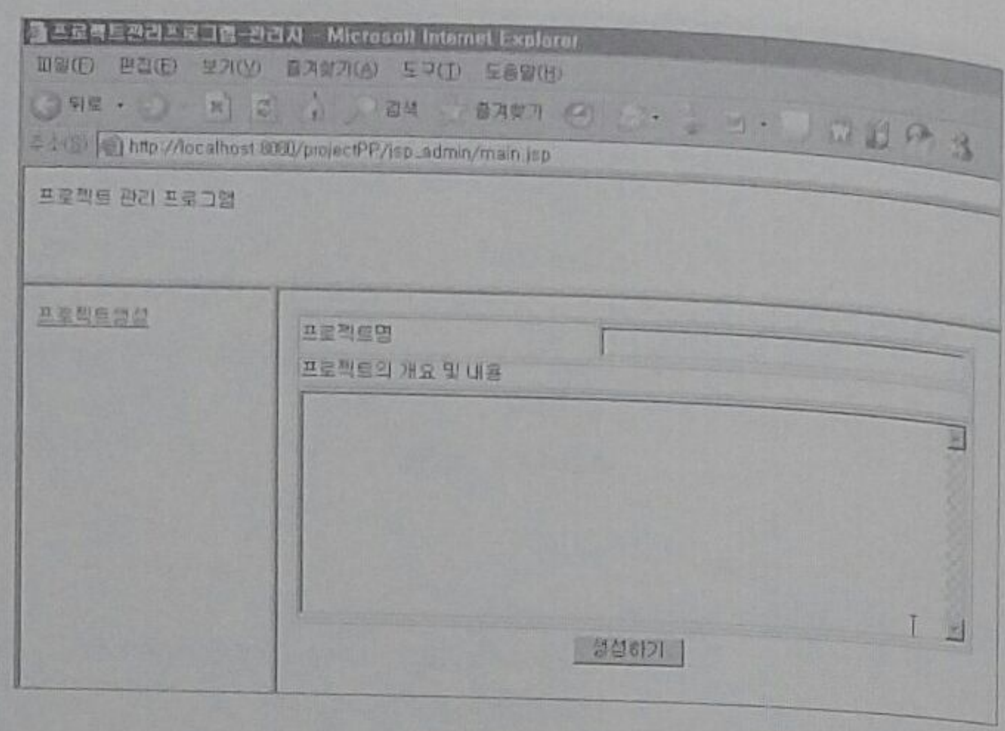
▶ [project_Crt.jsp]

```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<%@ include file="../include/sessionChk.jsp" %>
<html>
<head>
<title>프로젝트 생성</title>
<script language="javascript">
 function go_save ()
```

다음 페이지에 계속 ➲

```
{
        var theForm = document.frm;
        if(theForm.projectNm.value =='')
         {
                alert('프로젝트 이름을 입력 하세요');
         }
         else if( theForm.projectContent.value=='')
         {
                 alert('프로젝트의 내용을 입력 하세요');
         }
         else
         {
                 theForm.action="./project001_save.jsp";
                 theForm.submit();
         }
 }
</script>
</head>
<body>
<form name="frm" method="post">
<table border="1" width="473">
    <tr>
        <td width="183"><font size="2">프로젝트명</font></td>
        <td width="274">
          <font size="2"><input type="text" name="projectNm" size="37">
</font></td>
    </tr>
    <tr>
        <td width="463" colspan="2">
         <font size="2">프로젝트의 개요 및 내용</font>
       </td>
    </tr>
    <tr>
        <td width="463" height="11" colspan="2">
           <font size="2">
            <textarea name="projectContent" rows="10" cols="66"></textarea>
           </font>
        </td>
    </tr>
</table>
<table border="0" width="475">
    <tr>
        <td width="465" align="center">
         <input type="button" name="btn_save" value="생성하기" onClick="go_save()">
         </td>
    </tr>
</table>
</form>
</body>
</html>
```

❷ 브라우저에서 프로젝트 생성 버튼을 클릭하면 프로젝트에 대한 정보를 입력하는 화면을 출력한다.



▶ 프로젝트 입력 폼

❸ project_Crt.jsp에서 스크립트에서 정의한 입력값을 저장하는 project001_save.jsp를 작성한다.

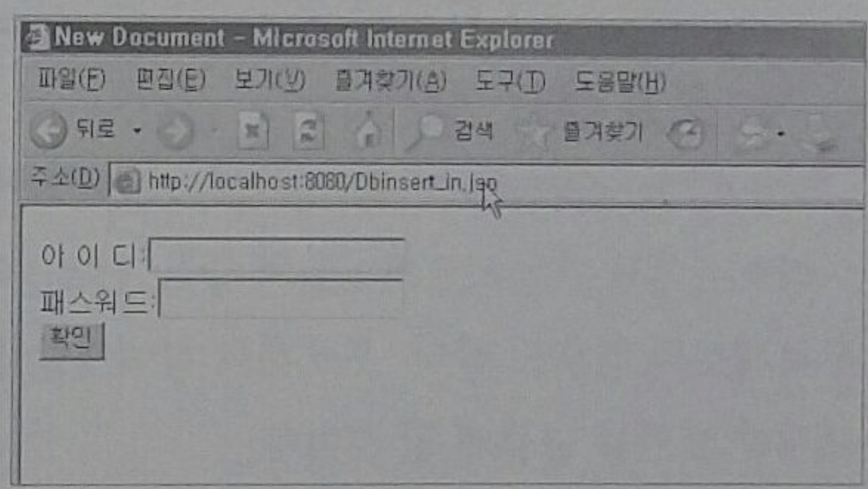
▶ [project001_save.jsp]

```
<%@ page language="java"  contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%@ include file="../include/sessionChk.jsp" %>
<%!
//null 처리를 위한 function
public static String nullcheck(String str) throws Exception {

if (str == null){
return "";
}else{
return str;      // 넘어온 Stirng을 그대로 다시 return
}
}

%>
<!--세션체크-->
<%@ include file="../include/sessionChk.jsp" %>
<%
//프로젝트 이름과 내용의 파라메타를 받는 변수
String projectNm = nullcheck(request.getParameter("projectNm"));
String projectContent = nullcheck(request.getParameter("projectContent"));

//저장 쿼리
String sql="insert into project_table(projectNM,projectContext,in_date) ";
      sql+="values( '"+projectNm+"' , '"+projectContent+"' ,DATE_FORMAT(sysdate(), '%Y%m%d' ) )";

//저장 결과를 받기위한 변수
int msg=0;
%>
<%@ include file="../include/try.jsp" %>
<%
      //정상적인 저장이면 msg에는 0보다 큰값이 저장, 오류일경우 0보다 적은값이 저장
          msg=stmt.executeUpdate(sql);
%>
```

다음 페이지에 계속 ➲

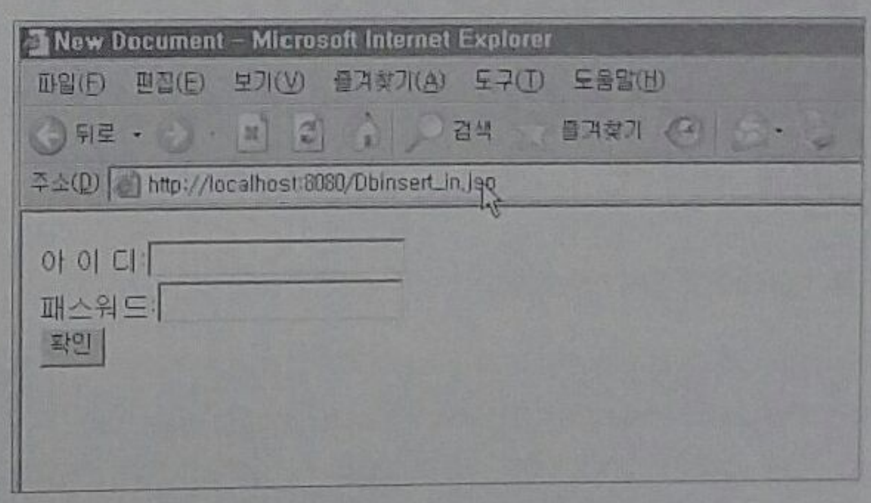
```
<%@ include file="../include/catch.jsp" %>
<%
 //msg 1: 정상저장 2: 오류발생
  if(msg > 0 )
  {
%>
<form name="resultForm" method="post">
</form>
<script language="javascript">
   document.resultForm.action="./project_List.jsp";
   document.resultForm.submit();
</script>
<%
  }
else
{
%>
<form name="resultForm" method="post">
</form>
<script language="javascript">
   document.resultForm.action="./project_Crt.jsp";
   document.resultForm.submit();
</script>
<%
  }
%>
```

❹ project_Crt.jsp에서 데이터를 입력하고 생성하기 버튼을 클릭한다.



▶ 데이터의 입력

❺ 정상적인 저장의 경우 project_List.jsp를 호출, 입력된 프로젝트의 리스트를 출력한다.



▶ 저장된 프로젝트 리스트

새로운 프로젝트가 생성이 되었다. 프로젝트의 리스트에서 해당 프로젝트의 제목을 클릭하면 클릭한 프로젝트의 상세와 프로젝트의 파일업로드 폴더의 지정 및 개발자 등록 게시판 생성에 대한 메뉴를 화면에 출력하는 project_sangse.jsp 파일을 작성한다.

먼저 project_List.jsp에서 프로젝트의 제목을 출력하는 소스에 project_sangse.jsp을 호출하는 링크를 추가하고 스크립트를 작성한다.

```
function go_view(seq)
{
    var theForm = document.frm;
    theForm.seq.value=seq;
    theForm.action="./project_sangse.jsp";
    theForm.submit();
}
</script>
</head>
<body>
<form name="frm" method="post">
<input type="hidden" name="seq">
<table border="1" width="604">
<tr>
    <td width="98"><font size="2">프로젝트번호</font></td>
    <td width="293"><font size="2">프로젝트명</font></td>
    <td width="111"><font size="2">등록일</font></td>
</tr>
<!--while문을 통해 리스트를 출력한다.-->
<%
int num=1;

if ("ok".equals(request.getParameter("fullview")))
   num = ((tpage-1)*view_rows)+1;

while(rs.next())
{

%>
<tr>
    <td width="98" height="11"><font size="2"><%=nullcheck(rs.getString(1))%></font></td>
    <td width="293" height="23">
    <font size="2">
        <a href="#" onClick="javascript:go_view('<%=nullcheck(rs.getString(1))%>')">
                <%=nullcheck(rs.getString(2))%>
        </a>
    </font>
    </td>
```

▶ 링크추가

프로젝트 상세

리스트에서 프로젝트의 seq의 값을 전달받아 해달 프로젝트의 제목과 내용을 출력, 그 아래에 해당 프로젝트의 업로드 폴더, 프로젝트 참여개발자 추가 및 리스트, 해당 프로젝트의 게시판 생성과 생성된 게시판 보기를 만들게 될 것이다.

따라하기 프로젝트 상세보기

❶ 리스트에서 프로젝트명을 클릭하면 상세보기에 필요한 파라메타의 값을 전달받아 해당 프로젝트에 대한 전반적인 정보를 보여주는 project_sangse.jsp를 작성한다.

▶ [project_sangse.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%!
//null 처리를 위한 function
public static String nullcheck(String str) throws Exception {

if (str == null){
return "";
```

다음 페이지에 계속 ➲

```
}else{
return str;      // 넘어온 Stirng을 그대로 다시 return
}
}

%>
<!--세션 체크-->
<%@ include file="../include/sessionChk.jsp" %>
<%
//해당 프로젝트의 seq값을 받는 변수
String seq = nullcheck(request.getParameter("seq"));

//게시판이 생성되었는지의 유무를 받는 변수
String ct = nullcheck(request.getParameter("ct"));

//해당 프로젝트의 제목과 내용을 가져오는 쿼리 작성
String sql_project="select projectNM,projectContext from project_table where seq='"+seq+"'";
String projectNm="";
String projectCon="";

//해당 프로젝트의 파일 업로드 관련 정보를 가져오는 쿼리
String sql_file="select filepath from project_file where procode='"+seq+"'";

//해당 프로젝트의 참여 개발자를 가져오는 쿼리
String sql_deplist ="select name,compy,tel from worker  where procode='"+seq+"'";

//생성 게시판 유무 쿼리
String sql_board="select max(projectNm) from board"+seq+" where procode='"+seq+"' and worker='관
리자' ";

//프로젝트 이름을 가져오는 변수
String board_projectNm="";
%>
<%@ include file="../include/try.jsp" %>
<%
   rs = stmt.executeQuery(sql_project);
   if(rs.next())
   {
            projectNm = nullcheck(rs.getString(1));
            projectCon=nullcheck(rs.getString(2));
   }
   rs.close();

//게시판이 생성이 되어야 실행
if("1".equals(ct))
{
   rs = stmt.executeQuery(sql_board);
   if(rs.next())
   {
            board_projectNm = nullcheck(rs.getString(1));
   }
   rs.close();
}

%>
<html>
<head>
```

다음 페이지에 계속

```
<title>프로젝트 상세</title>
<script language="javascript">
  function go_path()
  {
          var theForm = document.frm;
          theForm.action="./project_file.jsp";
          theForm.submit();
  }
  function go_search(seq)
  {
   var url="./project_person.jsp?seq="+seq;
   window.open(url, '_blank1', 'menubar=no,resizable=no,toolbar=no,
                              scrollbars=yes,status=no,width=400,height=200');
   }

   function go_board()
   {
          var theForm = document.frm;
           theForm.action="./project_board.jsp";
           theForm.submit();
   }

</script>
</head>
<body>
<form name="frm" method="post">
<input type="hidden" name="seq" value="<%=seq%>">
<input type="hidden" name="projectNm" value="<%=projectNm%>">
<table border="1" width="422">
    <tr>
        <td width="412"><font size="2">프로젝트명 :<%=projectNm%></font></td>
    </tr>
    <tr>
        <td width="412" align="center">
            <font size="2">프로젝트 개요 및 내용</font>
        </td>
    </tr>
    <tr>
        <td width="412" height="11"><font size="2"><%=projectCon%></font></td>
    </tr>
</table>
<p> </p>
<table border="1" width="420">
    <tr>
        <td width="161"><font size="2">프로젝트 업로드 폴더</font></td>
        <td width="243">
            <font size="2">
             <input type="text" name="filepath" size="22">
              <input type="button" name="btn_pathsave"
                                 value="저장" onClick="go_path();">
            </font>
        </td>
    </tr>
<%
  rs = stmt.executeQuery(sql_file);
  if(rs.next())
  {
%>
```

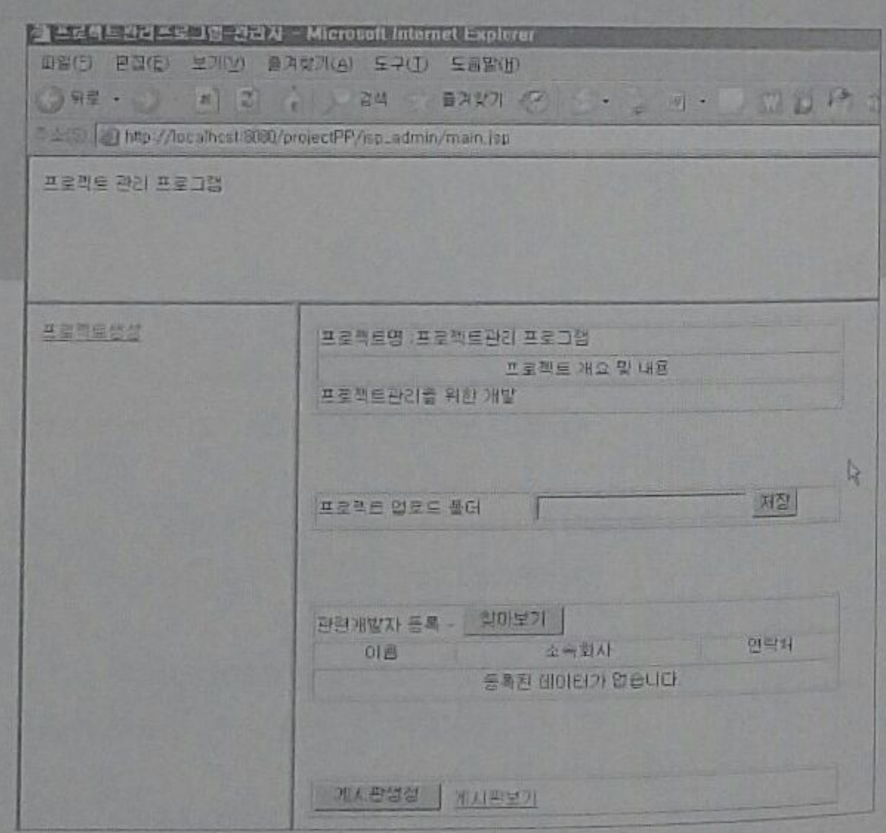
다음 페이지에 계속

```
        <tr>
      <td colspan="2">
        <font size="2">
            프로젝트 업로드 폴더 :<%=nullcheck(rs.getString(1))%>
        </font>
      </td>
    </tr>
<%
  }
rs.close();
%>
</table>
<p><font size="2"> </font></p>
<table border="1" width="420">
    <tr>
        <td width="410" colspan="3">
            <font size="2">관련개발자 등록 -</font>
                    <font size="2">
                     <input type="button" name="btn_search"
                      value="찾아보기" onClick="go_search( '<%=seq%>' )">
                  </font>
        </td>
    </tr>
    <tr>
        <td width="102" align="center"><font size="2">이름</font></td>
        <td width="184" align="center"><font size="2">소속회사</font></td>
        <td width="112" align="center"><font size="2">연락처</font></td>
    </tr>
<%
  int num=0;
  rs = stmt.executeQuery(sql_deplist);
  while(rs.next())
  {
%>
    <tr>
        <td width="102" align="center">
                <font size="2">><%=nullcheck(rs.getString(1))%></font>
        </td>
        <td width="184">
                <font size="2"><%=nullcheck(rs.getString(2))%></font>
        </td>
        <td width="112" align="center">
                <font size="2"><%=nullcheck(rs.getString(3))%></font>
        </td>
    </tr>
<%
        num++;
  }
  rs.close();
if(num ==0 )
{
%>
<tr>
    <td colspan="3" align="center">
        <font size="2">등록된 데이터가 없습니다.</font>
  </td>
</tr>
```

다음 페이지에 계속

```
<%
}
%>
</table>
<p> </p>
<table border="1" width="420">
    <tr>
        <td width="410">
            <font size="2">
                <input type="button" name="btn_board"
                        value="게시판생성" onClick="go_board()">
            </font>
        </td>
    </tr>
<%
  if(!"".equals(board_projectNm))
  {
%>
 <tr>
        <td width="410">
            <font size="2">
                <%=board_projectNm%> 게시판이 생성되었습니다.
            </font>
        </td>
    </tr>
<%
  }
%>
</table>
</form>
<%@ include file="../include/catch.jsp" %>
</body>
</html>
```

❷ 프로젝트 리스트가 출력된 화면에서 프로젝트명을 클릭하면 해당 프로젝트의 상세보기가 출력된다.



▶ 선택한 프로젝트 상세보기

❸ 선택한 프로젝트에 해당되는 파일 업로드 폴더를 저장하는 project_file.jsp를 작성한다.

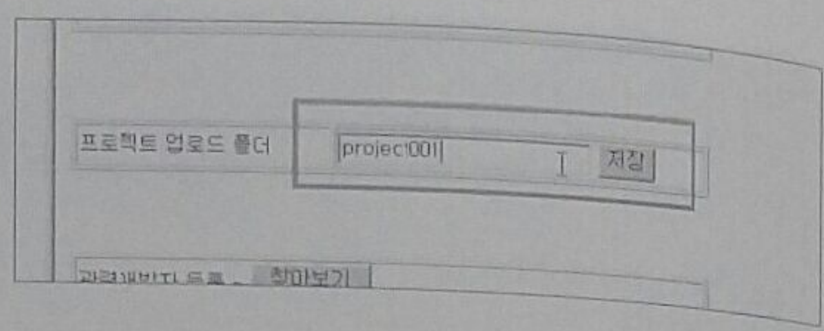
▶ [project_file.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%!
```

다음 페이지에 계속 ➡

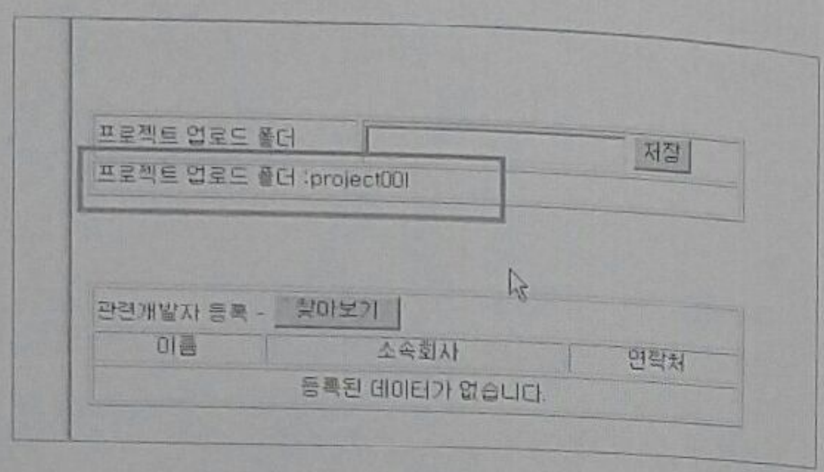
```
<%!
//null 처리를 위한 function
public static String nullcheck(String str) throws Exception {

if (str == null){
return "";
}else{
return str;      // 넘어온 Stirng을 그대로 다시 return
}
}

%>
<!--로그인 체크-->
<%@ include file="../include/sessionChk.jsp" %>
<%
//해당 프로젝트의 seq값을 받는 변수
String seq = nullcheck(request.getParameter("seq"));
String filepath = nullcheck(request.getParameter("filepath"));
//저장 쿼리
String sql="insert into project_file (procode,filepath) values('"+seq+"','"+filepath+"')";

//쿼리 결과 반환값
int msg=0;

%>
<%@ include file="../include/try.jsp" %>
<%
   msg = stmt.executeUpdate(sql);
%>
<%@ include file="../include/catch.jsp" %>
<%
 //msg 1: 정상저장 2: 오류발생
  if(msg > 0 )
  {
%>
<form name="resultForm" method="post">
<input type="hidden" name="seq" value="<%=seq%>">
</form>
<script language="javascript">
   document.resultForm.action="./project_sangse.jsp";
   document.resultForm.submit();
</script>
<%
  }
else
{
%>
<form name="resultForm" method="post">
<input type="hidden" name="seq" value="<%=seq%>">
</form>
<script language="javascript">
   alert('오류발생');
   document.resultForm.action="./project_sangse.jsp";
   document.resultForm.submit();
</script>
<%
  }
%>
```

❹ 브라우저에서 프로젝트 업로드 폴더에 데이터를 입력하고 저장하기 버튼을 클릭한다.



▶ 데이터의 입력

❺ 정상적인 저장이 되면 저장된 데이터의 리스트가 나타나게 된다.



▶ 업로드 폴더의 저장결과

❻ 사용자모드에서 개발자가 신규등록한 정보를 찾아 선택한 프로젝트의 개발자로 지정하는 project_persion.jsp를 작성한다.

▶ [project_persion.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                javax.naming.*,javax.sql.DataSource" %>
<%!
//null 처리를 위한 function
public static String nullcheck(String str) throws Exception {

if (str == null){
return "";
}else{
return str;     // 넘어온 Stirng을 그대로 다시 return
}
}

%>
<!--세션 체크-->
<%@ include file="../include/sessionChk.jsp" %>
<%
//해당 프로젝트의 seq값을 받는 변수
String seq = nullcheck(request.getParameter("seq"));

//등록된 개발자를 가지고 오는 쿼리
String sql="select name,compy,tel,seq from worker where id !='admin'";

int num=0;
%>
<html>
<head>
<title>개발자 선택</title>
<script language="javascript">

//체크된 데이터를 받기위한 배열 변수
var data =[] ;
//체크박스의 체크여부를 카운터하는 변수
```

다음 페이지에 계속 ➲

```
var count=0;
//체크박스가 체크 되었는지를 체크하는 변수
var incnt=0;

 function go_save(num)
 {
     var theForm = document.frm;
    //값 대입이 제대로 되었으면 true로 아니면 flase
    var check = false;

  if(num =0)
     {
              alert('체크할 데이터가 존재하지 않습니다.');
              return false;
     }

 if(num =1)
     {
             if(theForm.okcheck.checked = true)
              {
                       data=TheForm.pseq.value;
                            count++;
                            check=true;
              }
     }
     else
     {
             for(var i=0 ; i<num ; i++)
             {
                     if(TheForm.okcheck[i].checked =true)
                       {
                                  data[i]=TheForm.pseq[i].value;
                                  count++;
                                  check=true;
                            }
             }
 }

  if(check)
 {
         //count의 값이 0이거나 0보다 작다면 선택한 데이터가 없음으로
         //메시지를 띄운다.
          if(count <= 0)
          {
             alert('한개 이상의 데이터를 선택하세요');
          }
          else
          {
                  //post 방식으로 값을 넘겨주기 위해서이다.
                   theForm.data.value=data;
                   theForm.count.value=count;
                   theForm.action ="./project002_save.jsp";
                   theForm.submit();
          }
 }
 else
        {
```

다음 페이지에 계속

```
                    alert("Flg를 처리할 데이터를 체크해야 합니다.");

            }

}

 function go_cancel(num)
{
   var theForm = document.frm;
   //값 대입이 제대로 되었으면 true로 아니면 flase
   var check = false;

   if(num ==0)
       {
               alert('체크할 데이터가 존재하지 않습니다.');
               return false;
       }

  if(num ==1)
       {
              if(theForm.okcheck.checked == true)
               {
                 data=TheForm.pseq.value;
                 count++;
                 check=true;
               }
       }
       else
       {
               for(var i=0 ; i<num ; i++)
               {
                       if(TheForm.okcheck[i].checked ==true)
                         {
                              data[i]=TheForm.pseq[i].value;
                              count++;
                              check=true;
                              }
               }
       }

        if(check)
       {
              //count의 값이 0이거나 0보다 작다면 선택한 데이터가 없음으로
              //메시지를 띄운다.
               if(count <= 0)
               {
                     alert('한개 이상의 데이터를 선택하세요');
              }
              else
              {
              theForm.data.value=data;
                theForm.count.value=count;
                theForm.action ="./project003_save.jsp";
               theForm.submit();
              }
       }
       else
```

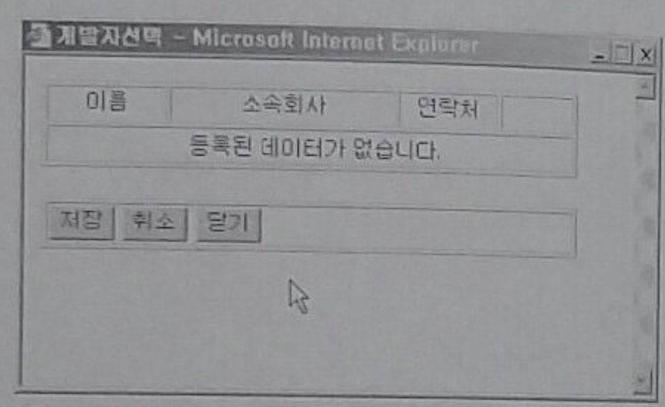
다음 페이지에 계속

```
                {
                        alert("Flg를 처리할 데이터를 체크해야 합니다.");
                }
  }

</script>
</head>
<body>
<form name="frm" method="post">
 <input type="hidden" name="seq" value="<%=seq%>">
 <input type="hidden" name="data" >
 <input type="hidden" name="count">
 <table border="1" width="332">
    <tr>
        <td width="68"  align="center"><font size="2">이름</font></td>
        <td width="132"  align="center"><font size="2">소속회사</font></td>
        <td width="59"  align="center"><font size="2">연락처</font></td>
        <td width="45"  align="center"> </td>
    </tr>
<%@ include file="../include/try.jsp" %>
<%

rs = stmt.executeQuery(sql);
while(rs.next())
{
%>
    <tr>
      <td width="68"  align="center">
          <font size="2">=nullcheck(rs.getString(1))%></font>
      </td>
      <td width="132">
        <font size="2">=nullcheck(rs.getString(2))%></font>
      </td>
      <td width="59"  align="center">
          <font size="2">=nullcheck(rs.getString(3))%></font>
      </td>
      <td width="45">
        <input type="checkbox" name="okcheck">
                <input type="hidden" name="pseq"
                      value="=nullcheck(rs.getString(4))%>">
      </td>
    </tr>
<%
        num++;
}
rs.close();
if(num == 0 )
{
%>
<tr>
  <td colspan="4" align="center">
        <font size="2">등록된 데이터가 없습니다.</font>
  </td>
</tr>
<%
}
%>
```

다음 페이지에 계속

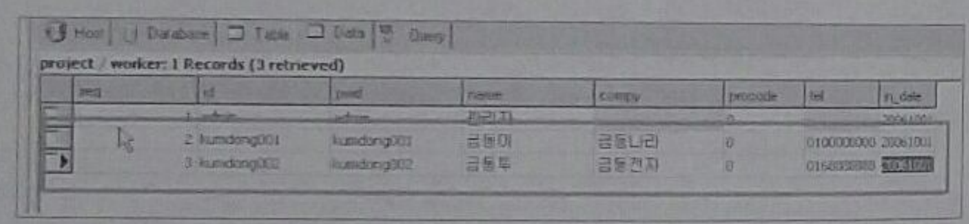
```
<%@ include file="../include/catch.jsp" %>
</table>
<br>
<table border="1" width="332">
    <tr>
        <td width="322">
            <input type="button" name="btn_ok"
                        value="저장" onClick="go_save( '<%=num%>' )">
            <input type="button" name="btn_no"
                        value="취소" onClick="go_cancel( '<%=num%>' )">
            <input type="button" name="btn_close" value="닫기" onClick="self.close()">
        </td>
    </tr>
</table>
</form>
</body>
</html>
```

❼ 상세에서 관련개발자 등록 – 찾아보기 버튼을 클릭하면 팝업창으로 등록되어진 개발자의 리스트를 출력한다.



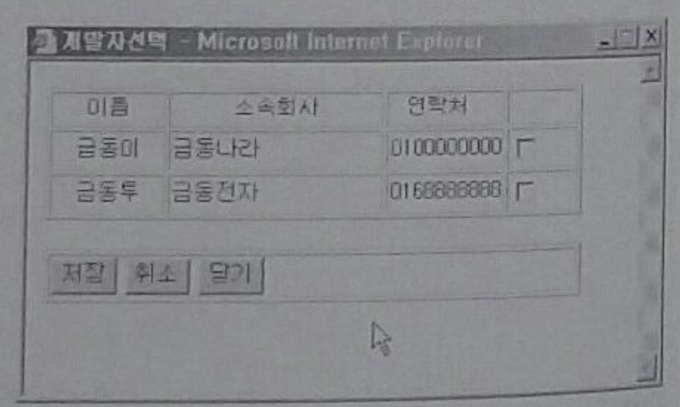
▶ 개발자 리스트

❽ 예제를 위해 임의적으로 worker 테이블에 테스트 데이터를 직접 입력한다.



▶ 테스트 데이터입력

❾ 프로젝트 상세보기에서 개발자 찾아보기를 하면 저장된 개발자 데이터의 리스트가 출력이 된다.



▶ 개발자 리스트

❿ 리스트에서 해당 프로젝트에 관련되는 개발자를 체크하고 저장 버튼을 클릭하면 worker 테이블의 procode 컬럼에 해당 프로젝트의 번호를 업데이트 하는 project002_save.jsp 파일을 작성한다.

▶ [project002_save.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%!
//null 처리를 위한 function
public static String nullcheck(String str) throws Exception {
```

다음 페이지에 계속 ➲

```
if (str == null){
return "";
}else{
return str;      // 넘어온 Stirng을 그대로 다시 return
}
}

%>
<%

Connection con=null;
//트랜잭션 처리와 빠른 insert를 위한 PreparedStatement 선언
PreparedStatement pstmt= null;

//전달받은 파라메타를 변수에 대입한다.
String data =nullcheck(request.getParameter("data"));
String seq=nullcheck(request.getParameter("seq"));
int cnt =Integer.parseInt(request.getParameter("count"));

String upSql="";

//seq를 받기위한 배열 변수선언
String [] upseq = null;

//트랜잭션 처리를 위한 변수선언
int[] trans = null;
boolean commit_flag= true;

try
{

    //트랜잭션 처리를 위해 바로 선언
    Context initCtx = new InitialContext();
    Context envCtx = (Context) initCtx.lookup("java:comp/env");
    DataSource ds = (DataSource) envCtx.lookup("jdbc/project");
    con = ds.getConnection();

    //트랜잭션 처리를 위해 commit를 false를 반드시 먼저 해야 한다.
  con.setAutoCommit(false);

    //data로 받아지는 값들이 , 구분자를 자동으로 가지고 전달되기 때문에
    //StringTokenizer을 이용하여 , 기준으로 잘라내어 값을 대입한다.
    //cnt는 선택된 데이터의 개수만큼 배열의 길이를 선언하기 위해서이다.
    //트랜잭션 처리를 위한 변수 trans도 같은 배열의 길이를 준다.
    StringTokenizer st = new StringTokenizer(data, ",");
    upseq = new String[cnt];
    trans  = new int[cnt];

    //,으로 기준으로 데이터를 token 하여 upseq에 대입한다.
    int i = 0;
     while(st.hasMoreTokens())
      {
       upseq[i] = st.nextToken();
       i++;
      }

//cnt(선택한 데이터의 개수) 만큼의 반복문
```

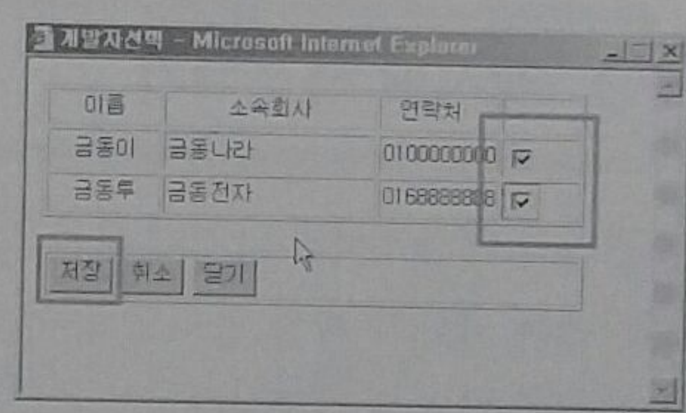
다음 페이지에 계속

```
for(int k =0;k<cnt;k++)
{
 try
 {
  //쿼리를 생성하고 prepareStatement를 한다.
  //이때 trans에 executeUpdate의 결과값을 대입한다.
  upSql="update worker set procode ='"+seq+"' where seq ='"+upseq[k]+"' ";
  pstmt= con.prepareStatement(upSql);
  trans[k]= pstmt.executeUpdate();
 }
   catch(Exception ee)
  {}
}//for close
pstmt.close();

  }catch(NamingException ne){
       out.println(ne.toString());
   }catch(SQLException se){
       out.println(se.toString());
        //에러가 발생하면 transdp -1 대입
        trans[0] = -1;
    }catch(Exception e){
        out.print(e);
        //에러가 발생하면 transdp -1 대입
        trans[0] = -1;
        }finally{
       try { if (pstmt != null) pstmt.close(); } catch (Exception e) {}
    }

try{
                    //선택한 데이터의 개수 cnt만큼 반복을 통해
                    //trans가 -1 의 값이 하나라도 있으면
                    //commit_flag에 flase값을 주고 반복문을 바로 빠져나온다.
                  //따라서 데이터베이스에 저장되지 않는다.
                    for(int i=0; i< cnt; i++){
                            if( trans[i] == -1  ){
                                    commit_flag = false;
                                    break;
                            }
                    }

                    //commit_flag의 값이 true이면 데이터베이스에 저장한다.
                    if( commit_flag ){
                            con.commit();
                            con.setAutoCommit(true);
                            con.close();
%>
<form name="formm" method="post" action="./project_sangse.jsp">
<input type="hidden" name="seq" value="<%=seq%>">
</form>
<script language="javascript">
        alert('정상적으로 처리 되었습니다.');
        window.close();
    window.opener.document.frm.target="detail";
    window.opener.document.frm.submit();
</script>
<%
```

다음 페이지에 계속

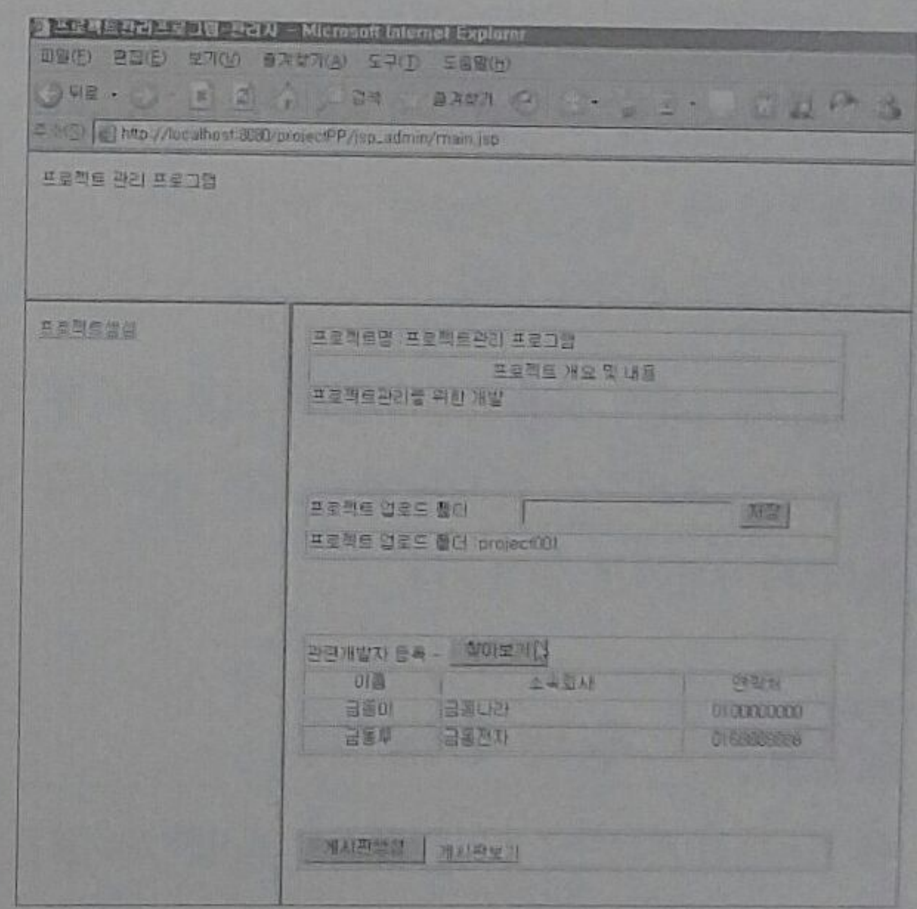
```
                }else{
                //에러발생시 데이터 롤백이 되어 데이터베이스에 저장되지 않는다.
               con.rollback();
                con.close();

%>
<form name="formm" method="post" action="./project_persion.jsp">
<input type="hidden" name="seq" value="<%=seq%>">
</form>
<script language="javascript">
        alert('오류가 발생되었습니다.');
        window.close();
       window.opener.document.frm.target="detail";
       window.opener.document.frm.submit();
</script>
<%
                }

        }catch(SQLException se){
                out.println(se);
        }
%>
```

⓫ 개발자를 선택하고 저장을 클릭한다.



▶ 데이터의 선택

⓬ 저장이 정상적이면 관련개발자 등록 –찾아보기를 통해 선택한 개발자의 리스트가 출력이 된다.



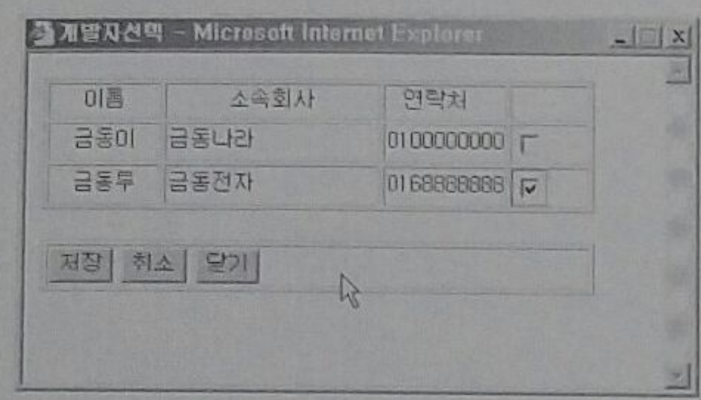
▶ 선택한 개발자 리스트

⓭ 선택한 개발자를 취소하기 위한 project003_save.jsp 파일을 작성한다. project003_save.jsp의 작성은 project002_save.jsp 파일을 그대로 복사하여 project003_save.jsp로 저장한 후 update 쿼리에서 procode의 값을 null 처리해주면 된다.

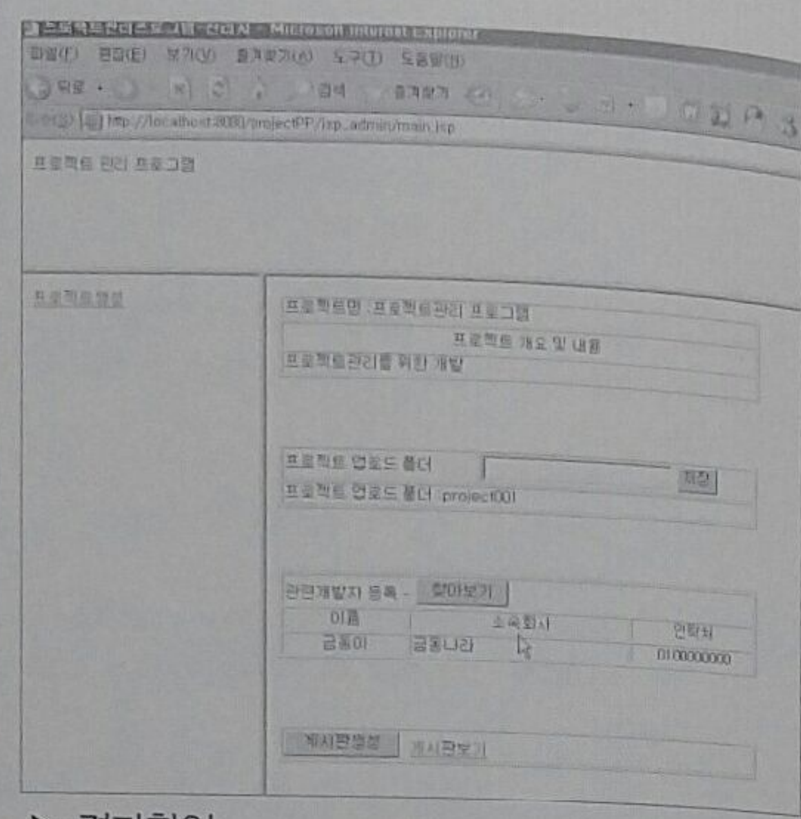
```
try
{
    //쿼리를 생성하고 prepareStatement를 한다.
    //이때 trans에 executeUpdate의 리턴값을 대입한다.
    upSql="update worker set procode ='' where seq ='"+upseq[k]+"' ";
    pstmt=con.prepareStatement(upSql);
    trans[k]= pstmt.executeUpdate();
}
catch(Exception ee)
```

▶ project003_save.jsp의 수정

⓮ 관련 개발자 찾아보기 팝업창에서 취소하고자 하는 개발자를 선택하고 취소 버튼을 클릭하면 프로젝트의 상세 리스트에는 취소되지 않은 개발자의 리스트만을 출력하게 된다.



▶ 개발자 취소



▶ 결과확인

개발자의 다중선택 다중취소의 방식은 12장에서 배운 부분을 응용한 것이다.

게시판 생성하기

관리자 메인에서의 마지막 부분으로 해당 프로젝트의 원활한 커뮤니티를 위한 게시판을 생성한다. 게시판에는 글쓰기 파일 업로드 등이 포함이 되어 있으며 사용자가 파일 업로드시 관리자가 지정한 업로드폴더에 파일이 저장된다.

먼저 파일의 업로드를 위하여 앞서 배운 Multipart를 이용한다. 생성된 context의 공통적인 부분으로 Multipart를 사용하고자 한다면 Multipart는 톰캣의 WEB-IMF 폴더에 위치해야 하며 각각의 독립적으로 생성된 context로 구현되었다면 projectPP 폴더 아래에 WEB-INF 폴더가 존재하고 그 안에 파일들이 존재해야 한다.

tmj 예제에서 사용했던 WEB-IMF 폴더를 복사하여 projectPP 폴더에 붙여 넣고 사용하면 된다. 13장에서는 관리자가 게시판에 필요한 테이블을 생성하기 때문에 게시판의 글쓰기 파일업로드 부분은 사용자 페이지에서 설명된다.

따라하기 게시판 테이블 생성

❶ 상세에서 게시판생성 버튼 클릭시 게시판 테이블을 생성하는 project_board.jsp을 작성한다.

▶ [project_board.jsp]

```
<%@page contentType= "text/html;charset = euc-kr"%>
<%@ page import="java.net.*,java.io.*,java.sql.*,java.util.*,
                                        javax.naming.*,javax.sql.DataSource" %>
<%!
//null 처리를 위한 function
```

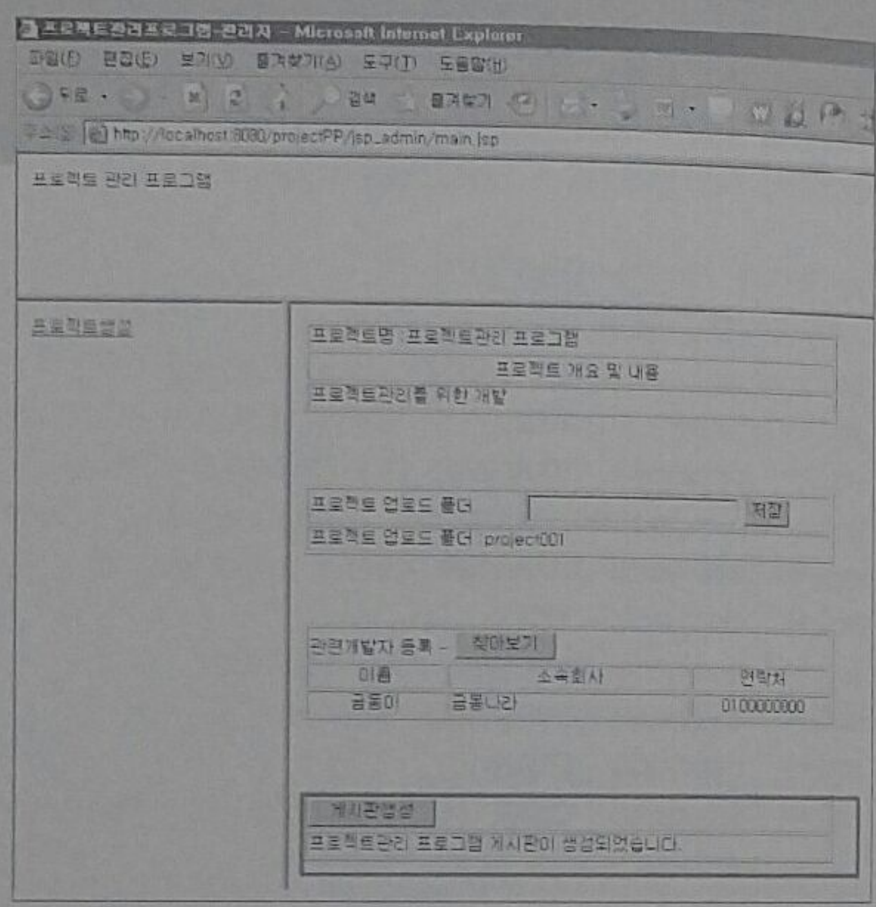
다음 페이지에 계속 ➲

```
public static String nullcheck(String str) throws Exception {

if (str == null){
return "";
}else{
return str;      // 넘어온 Stirng을 그대로 다시 return
}
}

%>
<!--세션 체크-->
<%@ include file="../include/sessionChk.jsp" %>
<%
//해당 프로젝트의 seq값을 받는 변수
String seq = nullcheck(request.getParameter("seq"));
String projectNm = nullcheck(request.getParameter("projectNm"));

//게시판 테이블 생성 쿼리
//일련번호, 프로젝트코드, 제목, 내용, 작성자, 작성일
String sql="CREATE TABLE `board"+seq+"` ";
sql+="(`seq` TINYINT (5) UNSIGNED NOT NULL AUTO_INCREMENT, ";
sql+="`procode` VARCHAR (5) DEFAULT '"+seq+"', ";
sql+="`projectNm` VARCHAR (100) DEFAULT '"+projectNm+"', ";
sql+="`subject` VARCHAR (100) DEFAULT '0', ";
sql+="`content` TEXT, ";
sql+="`worker` VARCHAR (50) DEFAULT '0', ";
sql+="`filename` VARCHAR (50) , ";
sql+="`repcode` CHAR(1) , ";
sql+="`in_date` CHAR (8) DEFAULT '0', PRIMARY KEY(`seq`), INDEX(`seq`)) ";

int msg=0;
//테이블이 생성되고 최고 관리자가 첫번째 데이터를 입력
String sql_in ="insert into board"+seq+" (subject,content,worker,repcode,in_date) ";
          sql_in+= "values( '"+projectNm+"게시판입니다','"+projectNm+"게시판입니다','관리자','
1',DATE_FORMAT(sysdate(), '%Y%m%d'))";

%>
<%@ include file="../include/try.jsp" %>
<%
  stmt.executeUpdate(sql);
  msg=stmt.executeUpdate(sql_in);
%>
<%@ include file="../include/catch.jsp" %>
<%
 //msg 1: 정상저장 2: 오류발생
 if(msg > 0 )
 {
%>
<form name="resultForm" method="post">
<input type="hidden" name="seq" value="<%=seq%>">
<!--ct: 게시판 테이블이 생성되었음을 나타냄-->
<input type="hidden" name="ct" value="1">
</form>
<script language="javascript">
   document.resultForm.action="./project_sangse.jsp";
   document.resultForm.submit();
</script>
```

다음 페이지에 계속

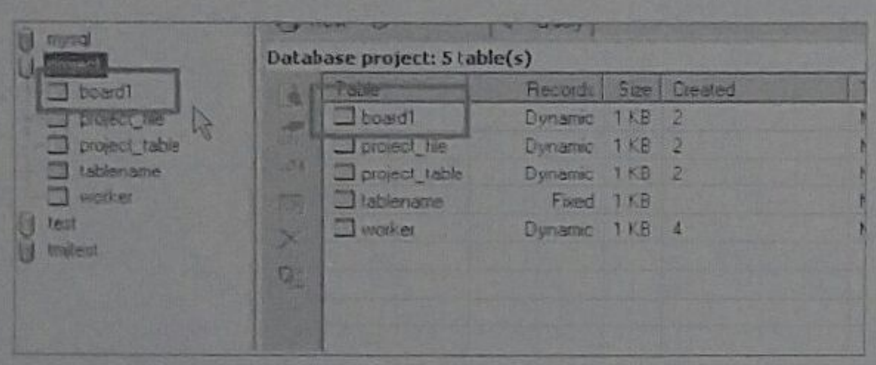
```
<%
  }
else
{
%>
<form name="resultForm" method="post">
<input type="hidden" name="seq" value="<%=seq%>">
</form>
<script language="javascript">
    document.resultForm.action="./project_sangse.jsp";
    document.resultForm.submit();
</script>
<%
  }
%>
```

❷ 프로젝트 상세보기에서 게시판생성 버튼을 클릭하면 생성된 게시판의 이름이 출력된다.

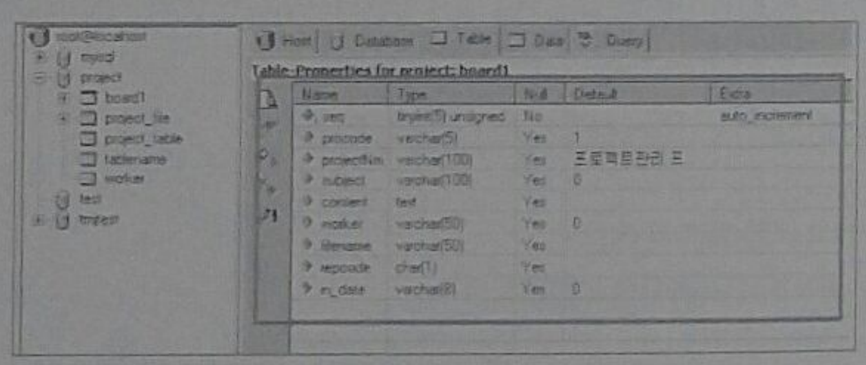


▶ 게시판의 생성

mysqlFront에서 확인해 보고 테이블이 생성된 것을 확인한다.



▶ 테이블의 생성



▶ 생성된 테이블의 칼럼

로그아웃

로그인을 했다면 로드아웃이 있어야 함은 당연하다. 로그아웃은 주어진 session을 없애고 초기 로그인의 화면으로 돌아가면 된다.

따라하기 로그아웃

❶ 메뉴를 보여주는 left.jsp 파일에 로그아웃에 대한 링크를 작성한다.

```
<base target="detail"></head>
<body bgcolor="white" text="black" link="blue" vlink="purple" alink="red">
<p><font size="3"><a href="./include/logout.jsp">로그아웃</a></font></p>
</body>
```

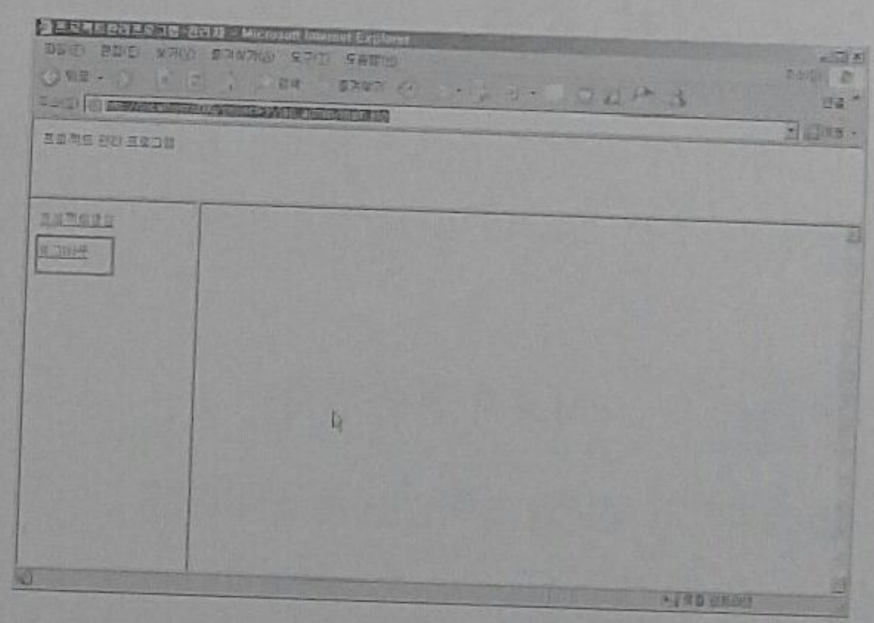
▶ 로그아웃의 추가

❷ include 폴더에 logout.jsp 파일을 작성한다.

▶ [logout.jsp]

```
<%@ page import="java.io.*,java.util.*" %>
<%@ page session="true" %>
<%@page contentType="text/html;charset=euc-kr"%>
<%
//id로 저장된 세션의 정보를 삭제한다.
session.removeAttribute("id");
%>
<script language=javascript>
    top.location.href='../index.jsp';
</script>
```

❸ 브라우저에서 로그아웃을 클릭하면 session의 정보를 지우고 초기 로그인의 화면으로 돌아가며 저장된 세션 값도 사라지게 된다.



▶ 로그아웃의 추가

맺음말

이번 장에서는 프로젝트 관리 프로그램이라는 가제를 통해 앞서 배운 예제를 종합적으로 응용하는 첫 번째를 공부했다. 관리자의 메뉴 부분을 텍스트로 만들지 않고 데이터베이스에 메뉴를 생성하여 불러들이는 작업이 있다면 더 활용성이 많을 것이다. 또한 파일의 업로드 부분의 경우 파일의 업로드 폴더의 이름만을 저장하지만 폴더를 생성하게 한다면 좋을 것이다. 이러한 방법은 앞에 있는 예제에 충분히 그 정보를 제공해 왔다. 이 부분을 남겨 놓은 것은 독자가 이 프로그램을 이용하여 작성해 보도록 하는 의도이다. 다음 장에서는 사용자 즉 개발자가 신규 등록을 하고 해당 게시판에 대한 권한을 가지고 게시판에 대한 글쓰기, 꼬리말, 파일업로드 등을 통해 이 프로젝트를 마무리하게 될 것이다.