

# 데이터베이스를 이용한 회원 관리 시스템 구축하기

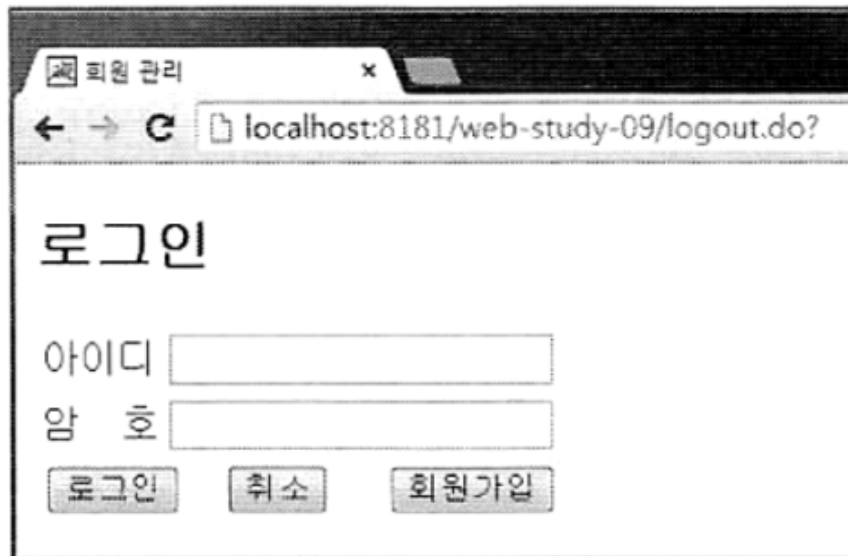
# 데이터베이스를 연동한 회원 관리 시스템

---

## ❖ 회원 관리 시스템

- 로그인

[로그인 페이지 login.jsp]



회원 관리 x

← → ↻ localhost:8181/web-study-09/logout.do?

### 로그인

아이디

암호

# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 회원 관리 시스템

### ○ 회원 가입

[회원 가입 페이지 : join.jsp]

회원 관리

localhost:8181/web-study-09/join.do

### 회원 가입

\* 표시 항목은 필수 입력 항목입니다.

이름  \*

아이디  \*

암호  \*

암호 확인  \*

이메일

전화번호

등급 ☒ 일반회원 ☐ 관리자

[로그인 페이지 : login.jsp]

회원 관리

localhost:8181/web-study-09/join.do

### 로그인

아이디

암호

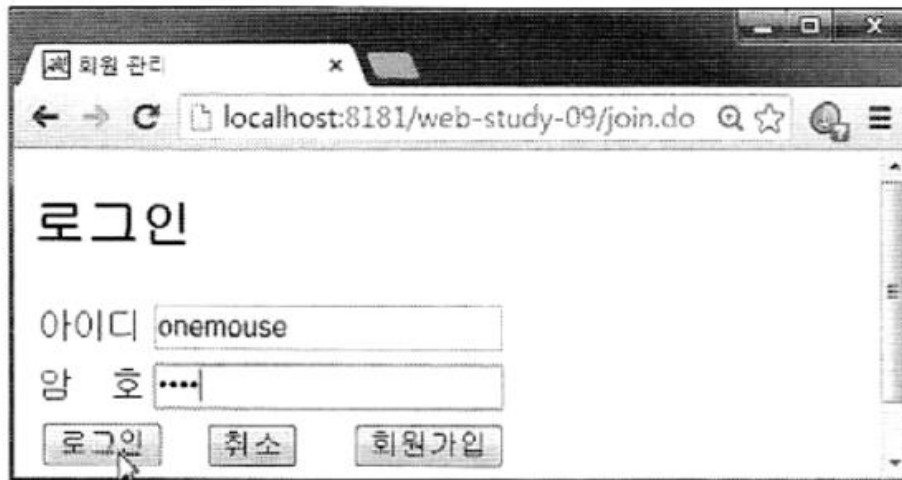
회원가입에 성공했습니다.

# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 회원 관리 시스템

- 회원인증

[로그인 페이지 : login.jsp]



[회원 전용 페이지 : main.jsp]



# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 회원 관리 시스템

- 회원 정보 수정

[회원 정보 수정 페이지 : memberUpdate.jsp]

회원 관리

localhost:8181/web-study-09/memberUpdate.do?userid=onem

### 회원 수정

이름:

아이디:

암호:  \*

암호 확인:  \*

이메일:

전화번호:

등급: ☐ 일반회원 ☒ 관리자

# 데이터베이스를 연동한 회원 관리 시스템

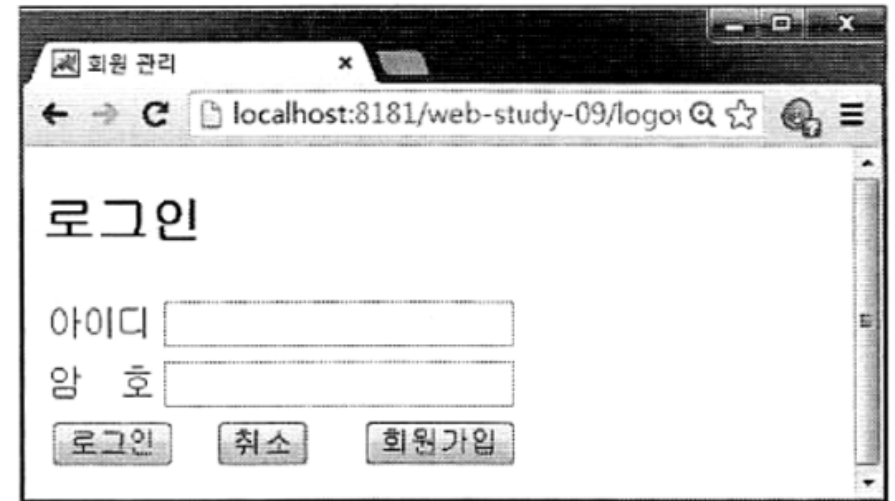
## ❖ 회원 관리 시스템

- 로그아웃

[회원 전용 페이지 : main.jsp]

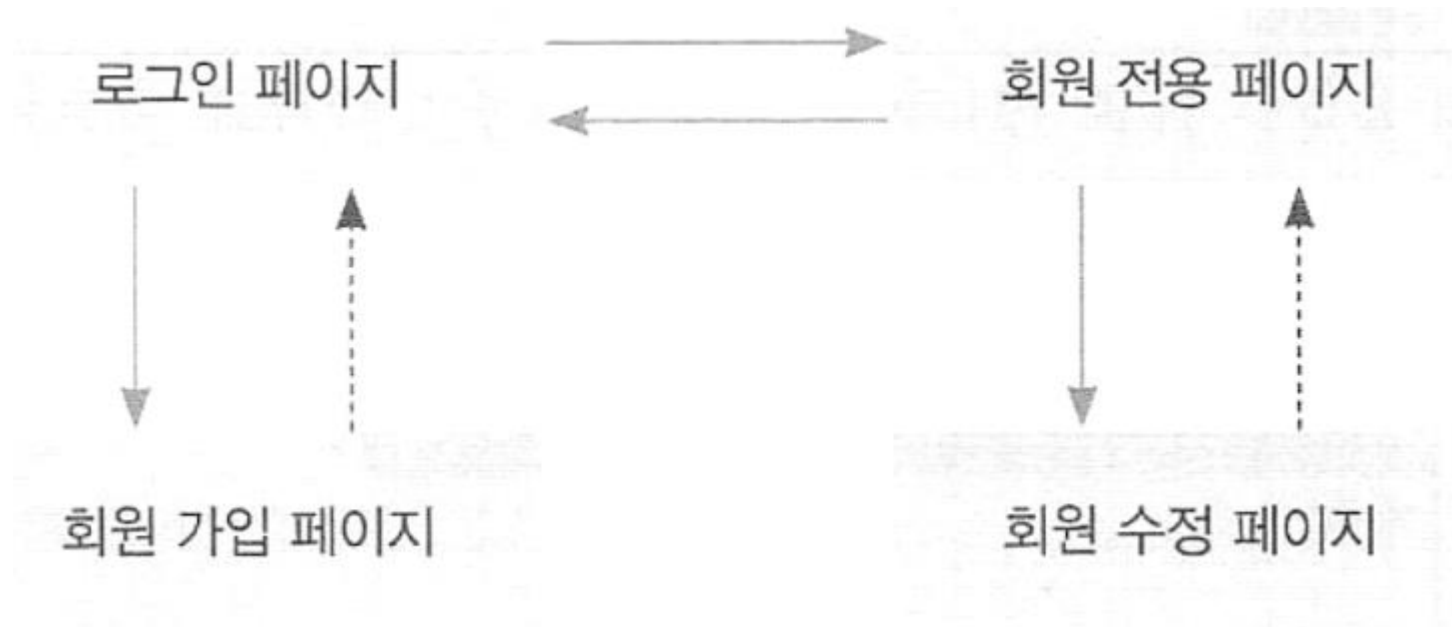


[로그인 페이지 : login.jsp]



# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 사용자 관리 시스템의 전체 구조



# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 사용자 관리 시스템의 전체 구조

[JSP 페이지]

파일	설명	위치
login.jsp	회원인증을 위한 아이디와 비밀번호를 입력받는 폼	WebContent\member
join.jsp	회원 가입을 위해 정보를 입력받는 폼	WebContent\member
main.jsp	회원 인증 확인 후 다양한 서비스를 제공하는 폼	WebContent
memberUpdate.jsp	회원 정보를 수정하기 위한 폼	WebContent\member



# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 사용자 관리 시스템의 전체 구조

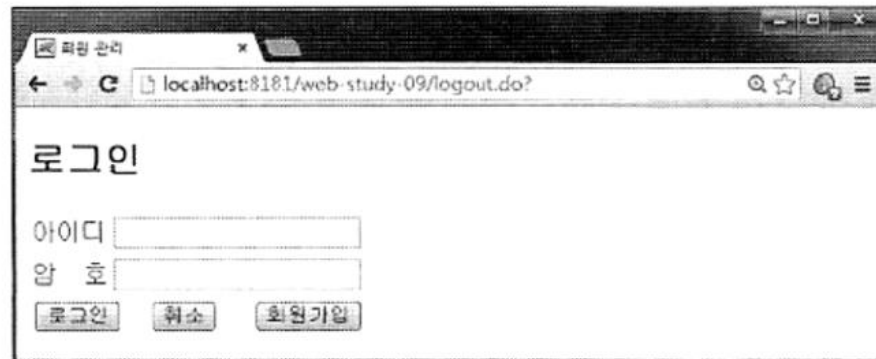
[서블릿 파일(위치 : srcWcomWsaeyanWcontroller)]

파일	설명	URL pattern
JoinServlet.java	입력된 회원 정보로 회원 가입 처리	join.do
LoginServlet.java	회원 인증 처리	login.do
MemberUpdateServlet.java	입력된 회원 정보로 회원 정보 수정	memberUpdate.do
LogoutServlet.java	로그아웃 처리	logout.do

# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 사용자 관리 시스템의 전체 구조

login.jsp

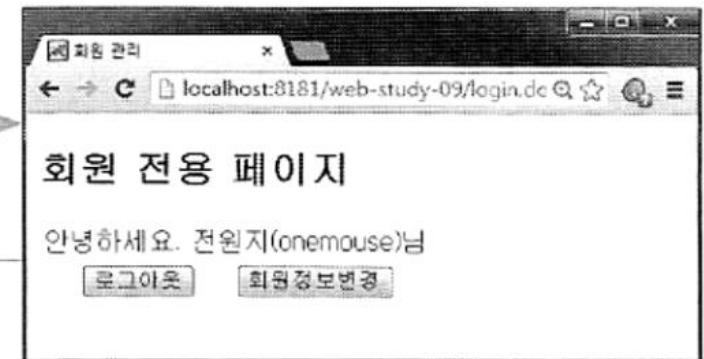


로그인

아이디

암호

main.jsp



회원 전용 페이지

안녕하세요. 전원지(onemouse)님

join.jsp



회원 가입

\* 표시 항목은 필수 입력 항목입니다.

이름  \*

아이디  \*

암호  \*

암호 확인  \*

이메일

전화번호

등급 ☒ 일반회원 ☐ 관리자

memberUpdate.jsp



회원 수정

이름

아이디

암호  \*

암호 확인  \*

이메일

전화번호

등급 ☒ 일반회원 ☐ 관리자

# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 사용자 관리 시스템의 전체 구조

[자바스크립트(위치 : WebContentWscript)]

파일	설명
member.js	폼에 입력된 정보가 올바른지 판단하는 자바스크립트

[VO 클래스 (위치: srcWcomWsaeyanWdto)]

파일	설명
MemberVO.java	회원 정보를 저장하는 클래스

[DAO 클래스 (위치: srcWcomWsaeyanWdao)]

파일	설명
MemberDAO.java	데이터베이스 테이블과 연동해서 작업하는 데이터베이스 처리 클래스

# 데이터베이스를 연동한 회원 관리 시스템

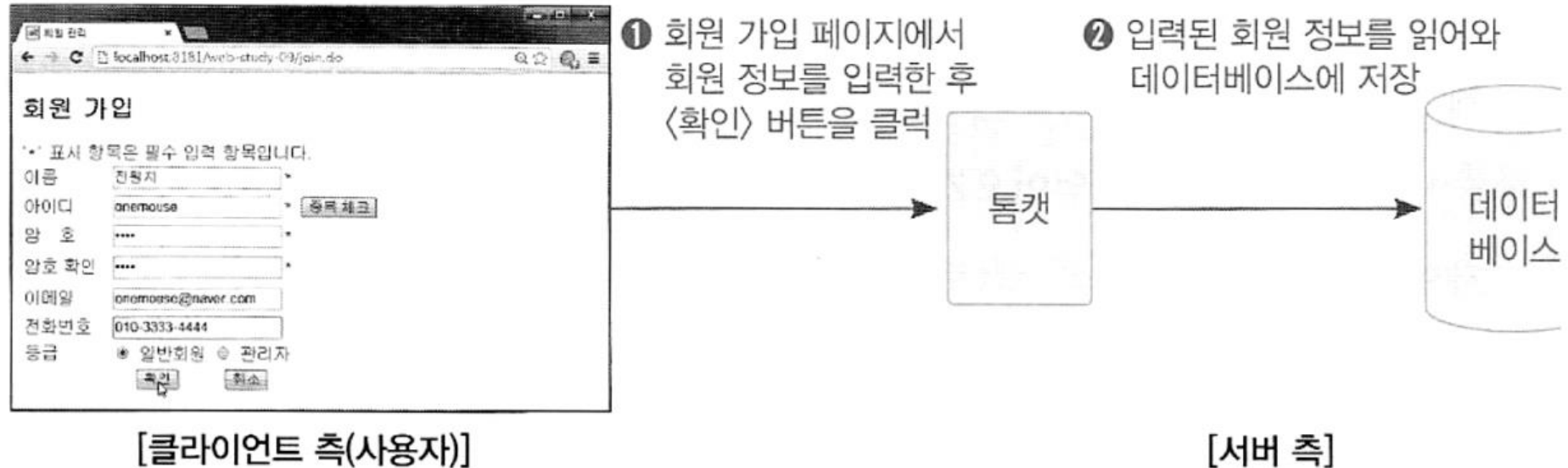
## ❖ 사용자 관리 시스템의 전체 구조

[member 테이블 구조]

컬럼명	크기	설명
name	varchar2(30)	이름
userid	varchar2(30)	아이디
pwd	varchar2(20)	비밀번호
email	varchar2(30)	이메일
phone	char(13)	전화번호
admin	number(1)	관리자 구분 번호(0:사용자, 1:관리자)

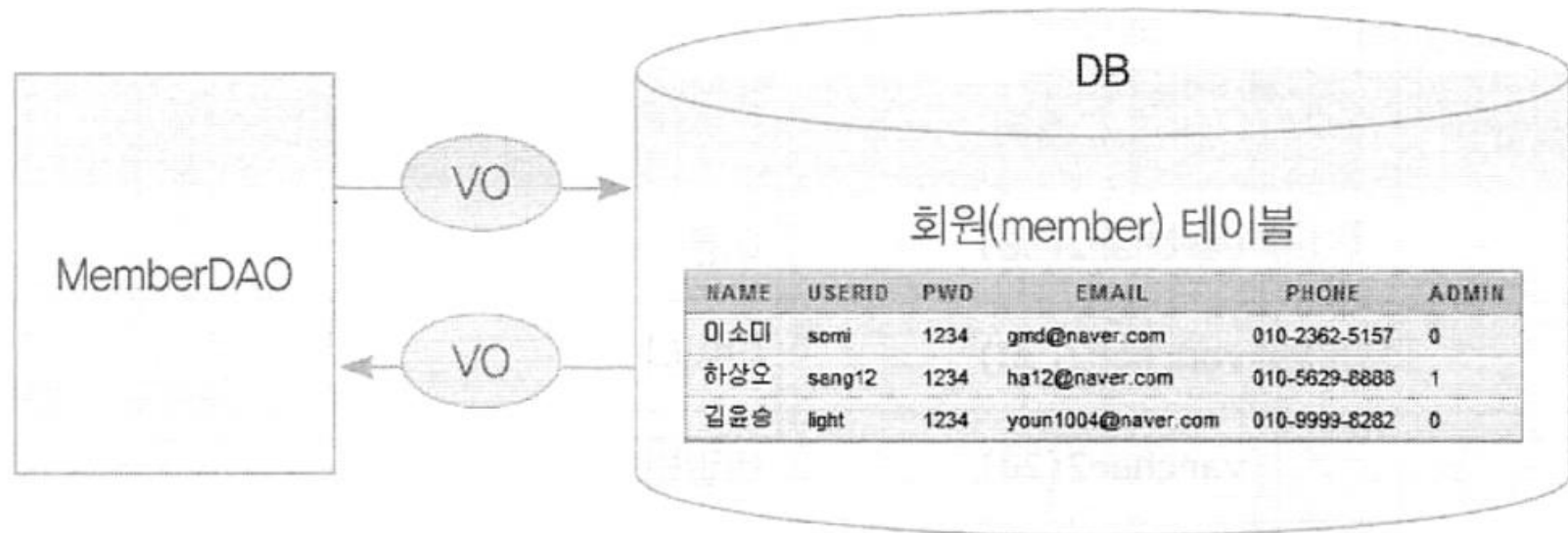
# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 사용자 관리 시스템의 전체 구조



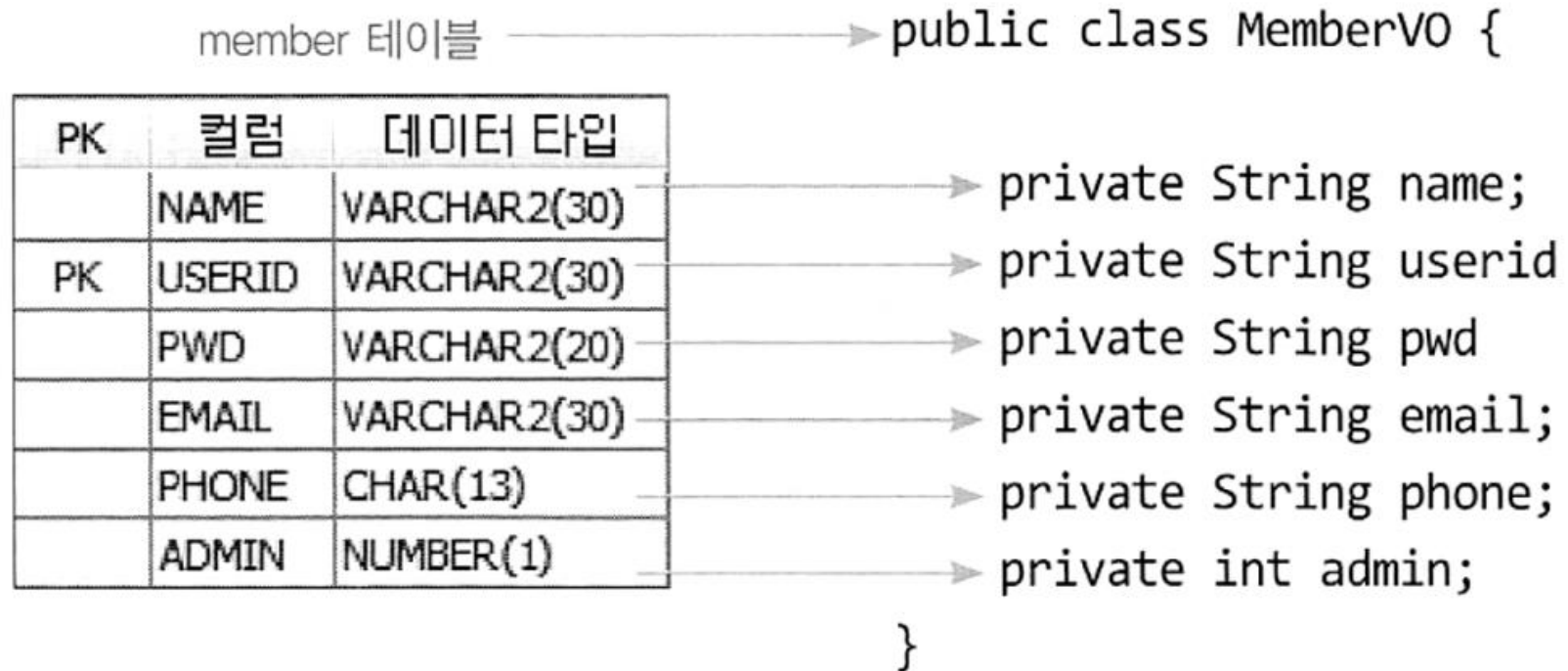
## 데이터베이스를 연동한 회원 관리 시스템

### ❖ 회원 관리 member 테이블과 연동하는 DAO



## 데이터베이스를 연동한 회원 관리 시스템

### ❖ 회원 관리 member 테이블과 연동하는 DAO





# 데이터베이스를 연동한 회원 관리 시스템

---

## ❖ MemberVO.java

```
package com.saeyan.dto;

public class MemberVO {
    private String name;
    private String userid;
    private String pwd;
    private String email;
    private String phone;
    private int admin;

    // 생성자
    // getter/setter
    // toString
}
```



# 데이터베이스를 연동한 회원 관리 시스템

---

## ❖ MemberDAO.java

- 싱글톤(Singleton) 패턴 적용

```
package com.saeyan.dao;
:
public class MemberDAO {
    private MemberDAO() { }

    private static MemberDAO instance = new MemberDAO();

    public static MemberDAO getInstance() {
        return instance;
    }
}
```

# 데이터베이스를 연동한 회원 관리 시스템

---

## ❖ MemberDAO.java

- 싱글톤(Singleton) 객체 얻기

```
MemberDAO mDao = MemberDAO.getInstance();
```

# 데이터베이스를 연동한 회원 관리 시스템

---

## ❖ MemberDAO.java

- DB 커넥션 얻기

```
package com.saeyan.dao;
:
public class MemberDAO {
    :

    public Connection getConnection() throws Exception {
        Connection conn = null;
        Context initContext = new InitialContext();
        Context envContext = (Context) initContext.lookup("java:/comp/env");
        DataSource ds = (DataSource) envContext.lookup("jdbc/myoracle");
        conn = ds.getConnection();
        return conn;
    }
}
```

# 데이터베이스를 연동한 회원 관리 시스템

## ❖ 02\_test.jsp

- 커넥션 획득 테스트

```
<%@page import="java.sql.Connection"%>
<%@page import="com.saeyan.dao.MemberDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <%
        MemberDAO memDao = MemberDAO.getInstance();
        Connection conn = memDao.getConnection();
        out.println("DBCP 연동 성공");
    %>
</body>
</html>
```

## 데이터베이스를 연동한 회원 관리 시스템

### ❖ MemberDAO.java

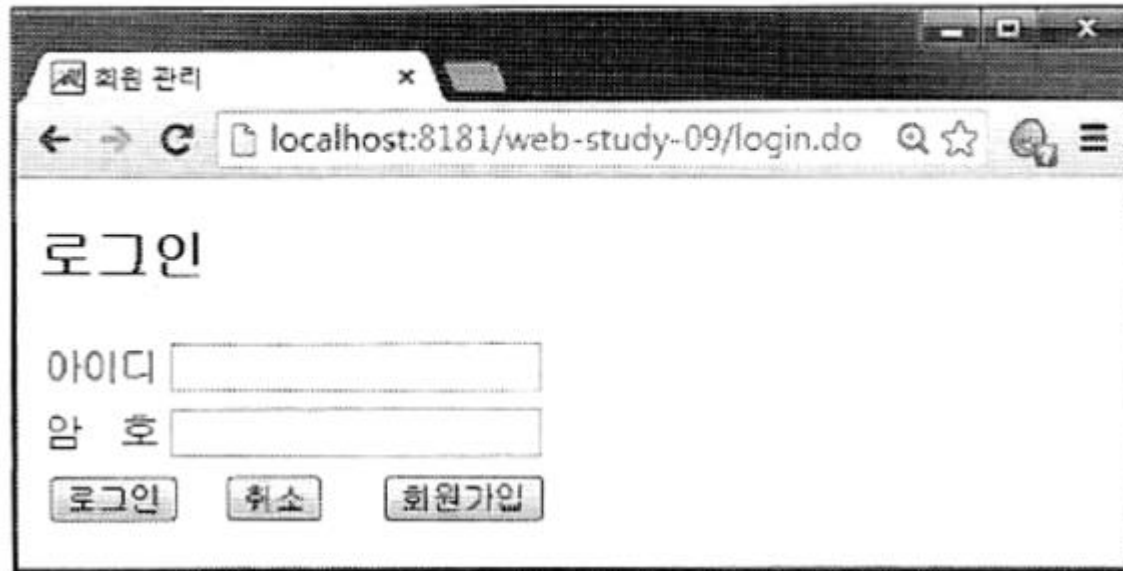
메소드	내용
<code>int userCheck(String userid, String pwd)</code>	사용자 인증시 사용하는 메소드이다. member 테이블에서 아이디와 암호를 비교해서 해당 아이디가 존재하지 않으면 -1을, 아이디만 일치하고 암호가 다르면 0을, 모두 일치하면 1을 리턴한다.
<code>MemberVO getMember(String userid)</code>	member 테이블에서 아이디로 해당 회원을 찾아 회원 정보를 가져온다.
<code>int confirmID(String userid)</code>	회원 가입시 아이디 중복을 확인할 때 사용한다. 해당 아이디가 있으면 1을, 없으면 -1을 리턴한다.
<code>void insertMember(MemberVO mVo)</code>	매개 변수로 받은 VO 객체를 member 테이블에 삽입한다.
<code>void updateMember(MemberVO mVo)</code>	매개 변수로 받은 VO 객체 내의 아이디로 member 테이블에서 검색해서 VO 객체에 저장된 정보로 회원 정보를 수정한다.

## 로그인 인증

---

### ❖ 로그인 인증 처리

[로그인 페이지 : login.jsp]



The screenshot shows a web browser window with the address bar displaying 'localhost:8181/web-study-09/login.do'. The page content includes the title '로그인' (Login) and two input fields: '아이디' (ID) and '암호' (Password). Below the input fields are three buttons: '로그인' (Login), '취소' (Cancel), and '회원가입' (Sign Up).

## ❖ WebContent/member/login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
:
<body>
    <h2>로그인</h2>
    <form action="login.do" method="post" name="frm">
        <table>
            <tr>
                <td>아이디</td>
                <td><input type="text" name="userid" value="${userid}"></td>
            </tr>
            <tr>
                <td>암 호</td>
                <td><input type="password" name="pwd"></td>
            </tr>
        </table>
    </form>
</body>
```

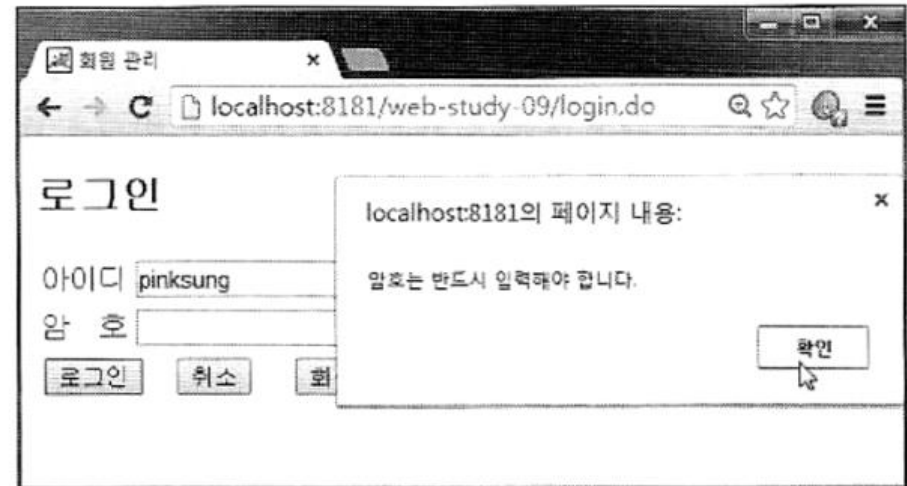
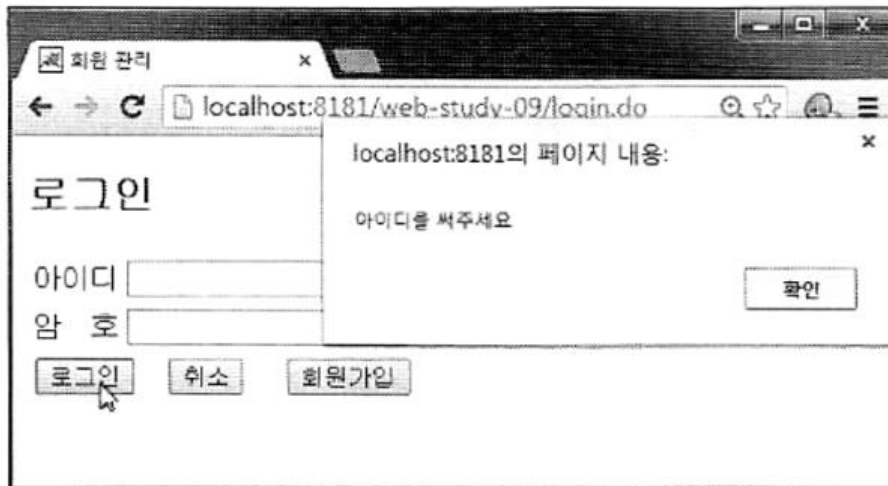
```
<tr>  
    <td colspan="2" align="center">  
        <input type="submit" value="로그인"  
            onclick="return loginCheck()">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <input type="reset" value="취소"> ~~~~~~  
        <input type="button" value="회원 가입"  
            onclick="location.href='join.do'">  
    </td>  
</tr>  
<tr>  
    <td colspan="2">${message}</td>  
</tr>  
</table>  
</form>  
</body>  
</html>
```



## 로그인 인증

### ❖ 유효성 체크를 위한 자바 스크립트 파일

- 아이디와 암호가 모두 입력되었는지 유효성 파악



## 로그인 인증

---

### ❖ 유효성 체크를 위한 자바 스크립트 파일

○ WebContent/script/member.js

```
function loginCheck() {  
    if (document.frm.userid.value.length == 0) {  
        alert("아이디를 써주세요");  
        frm.userid.focus();  
        return false;  
    }  
    if (document.frm.pwd.value == "") {  
        alert("암호는 반드시 입력해야 합니다.");  
        frm.pwd.focus();  
        return false;  
    }  
    return true;  
}
```

## 로그인 인증

---

### ❖ 유효성 체크를 위한 자바 스크립트 파일

login.jsp

```
<form action="login.do" method="post" name="frm" >
```

아이디    <input type="text" name="userid" value="\$ {userid}">

member.js

```
document.frm.userid
```

## ❖ WebContent/member/login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Insert title here</title>
    <script type="text/javascript" src="script/member.js"></script>
</head>
<body>
    <h2>로그인</h2>
    :
```

## 로그인 인증

---

### ❖ 로그인 입력 폼을 위한 서블릿

```
package com.saeyan.controller;
:
@WebServlet("/login.do")
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public LoginServlet() {
        super();
    }

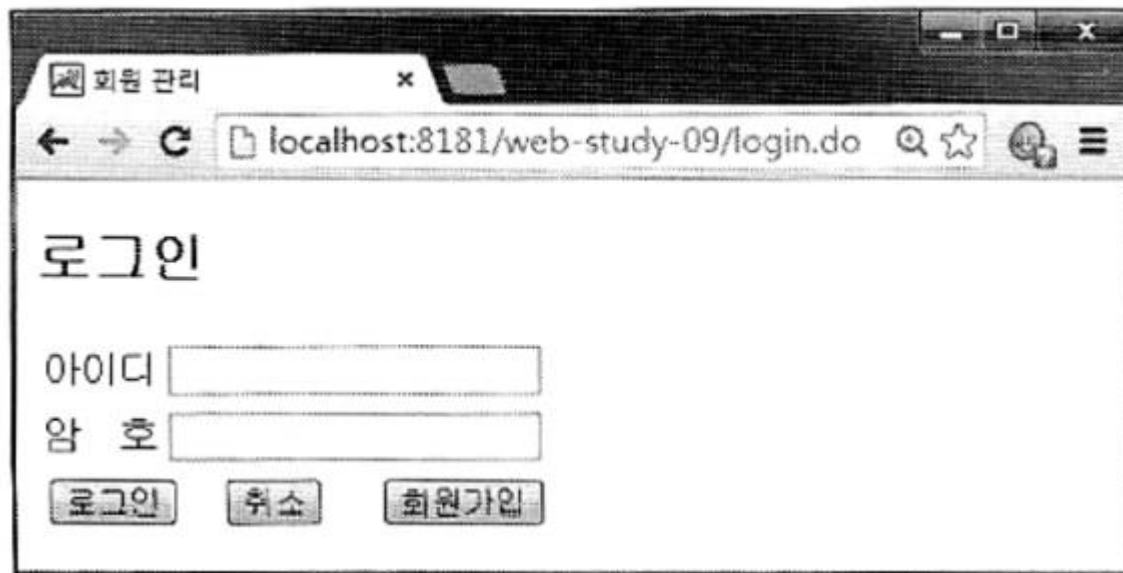
    protected void doGet(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {
        String url = "member/login.jsp";
        RequestDispatcher dispatcher = request.getRequestDispatcher(url);
        dispatcher.forward(request, response);
    }

    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
    }
}
```

## 로그인 인증

### ❖ 로그인 입력 폼 요청 url

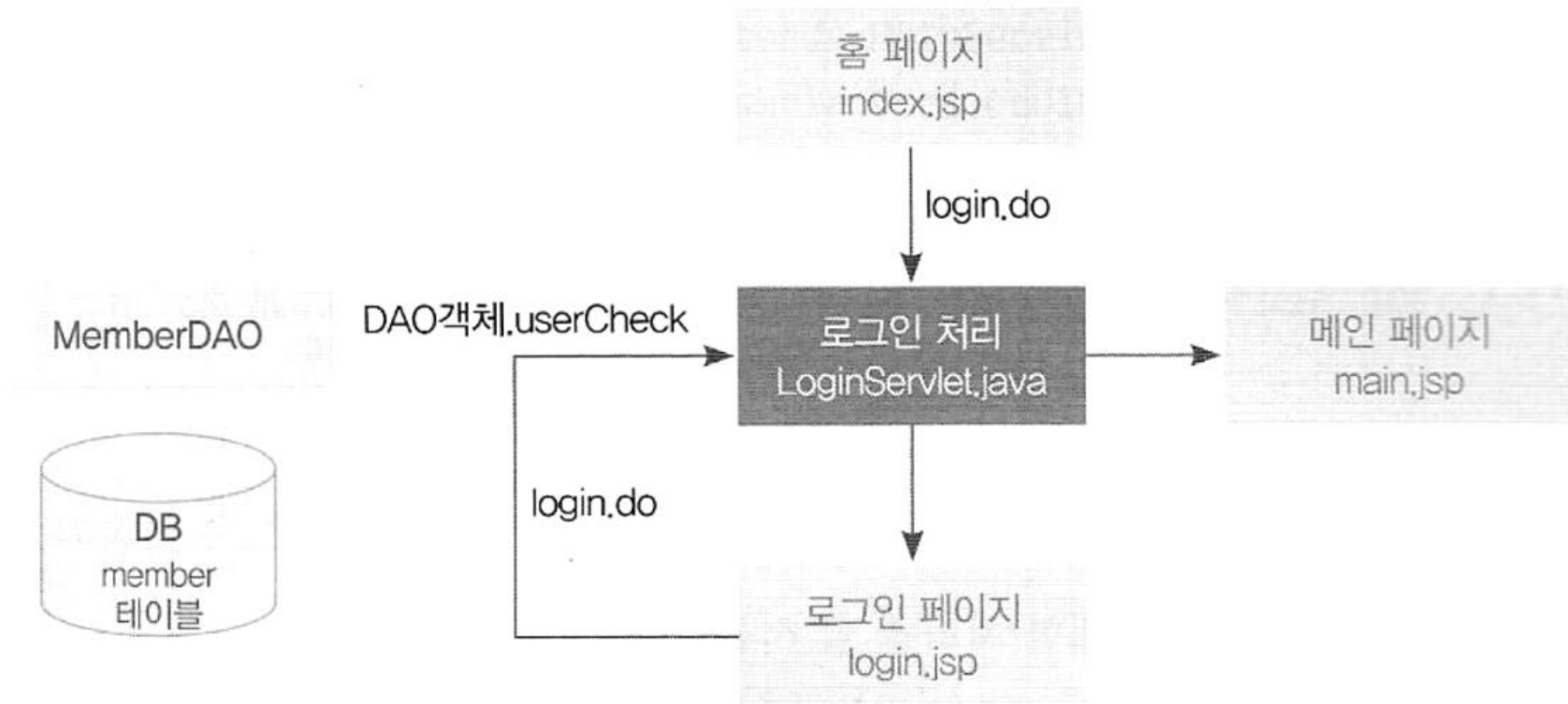
`http://localhost:8181/web-study-09/login.do`



A screenshot of a web browser window. The address bar shows the URL `localhost:8181/web-study-09/login.do`. The page content includes the title "로그인" (Login), two input fields labeled "아이디" (ID) and "암호" (Password), and three buttons: "로그인" (Login), "취소" (Cancel), and "회원가입" (Sign Up).

## 로그인 인증

### ❖ 로그인 처리



## 로그인 인증

---

### ❖ 홈(프론트) 페이지

- WebContent/index.jsp
  - 로그인 페이지로 이동 링크 제시

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <a href="login.do"> 로그인 페이지로 이동 </a>
</body>
</html>
```

http://localhost:8181/web-study-09



## 로그인 인증

---

### ❖ 로그인 인증 처리를 위한 프로그래밍

```
<form action="login.do" method="post" name="frm">  
...  
</form>
```

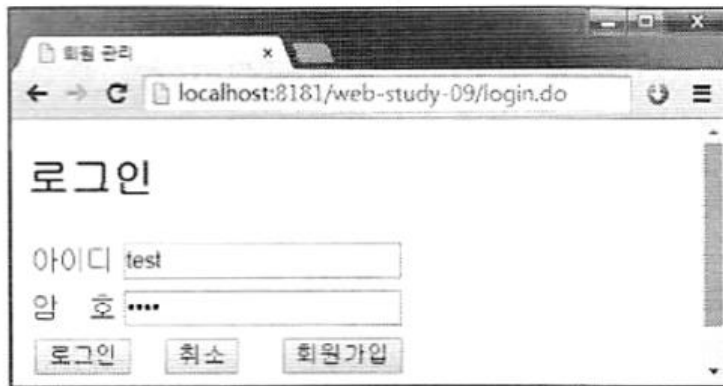
- 로그인 버튼 클릭
  - login.do의 POST 방식으로 요청
  - LoginServlet의 doPost에서 로그인 처리
  - 성공하는 경우와 실패하는 경우 고려하여 구현

## 로그인 인증

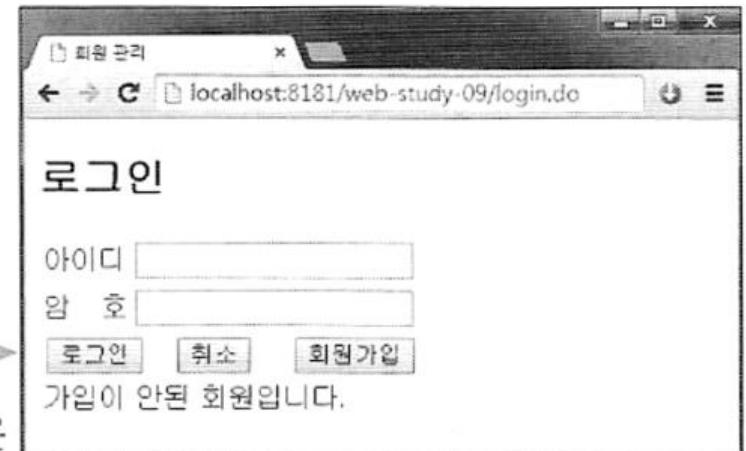
### ❖ 로그인 인증 처리를 위한 프로그래밍

- 로그인 실패의 경우
  - 다시 로그인 페이지로 이동
  - 실패 이유 설명

[로그인 페이지 : login.jsp]



[로그인 페이지 : login.jsp]



인증 처리에 실패한 회원은  
다시 로그인 작업을 하기 위해서  
login.jsp 페이지로 이동

## 로그인 인증

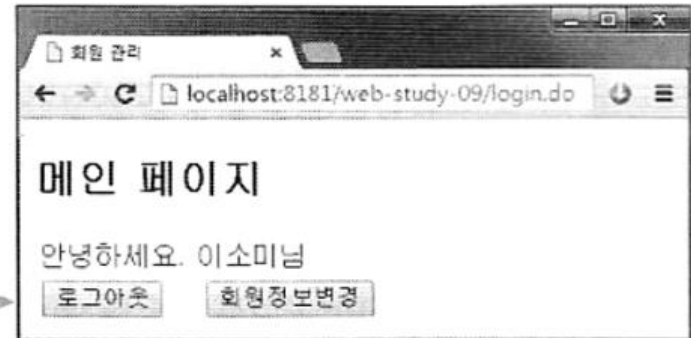
### ❖ 로그인 인증 처리를 위한 프로그래밍

- 로그인 성공의 경우
  - main.jsp로 이동

[로그인 페이지 : login.jsp]



[회원 전용 페이지 : main.jsp]



회원 인증 처리에 성공한 회원은  
main.jsp 페이지로 이동

## 로그인 인증

---

### ❖ 회원 인증을 위한 MemberDAO 메서드 추가

- userCheck 메서드 추가 : id/password 검사

```
public class MemberDAO {  
:  
    // 사용자 인증시 사용하는 메소드  
    public int userCheck(String userid, String pwd) {  
        int result = -1;  
        String sql = "select pwd from member where userid=?";  
        Connection conn = null;  
        PreparedStatement pstmt = null;  
        ResultSet rs = null;
```

## 로그인 인증

---

### ❖ 회원 인증을 위한 MemberDAO 메서드 추가

- userCheck 메서드 추가 : id/password 검사

```
try {
    conn = getConnection();
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, userid);
    rs = pstmt.executeQuery();
    if (rs.next()) {
        if (rs.getString("pwd") != null
            && rs.getString("pwd").equals(pwd)) {
            result = 1;
        } else {
            result = 0;
        }
    } else {
        result = -1;
    }
}
```

## 로그인 인증

---

### ❖ 회원 인증을 위한 MemberDAO 메서드 추가

- userCheck 메서드 추가 : id/password 검사

```
        catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                if (rs != null)
                    rs.close();
                if (pstmt != null)
                    pstmt.close();
                if (conn != null)
                    conn.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return result;
    }
}
```

## 로그인 인증

---

### ❖ 회원 인증을 위한 MemberDAO 메서드 추가

- getMember() 메서드 추가 : id로 회원 정보 가져오는 메서드

```
// 아이디로 회원 정보 가져오는 메소드
public MemberVO getMember(String userid) {
    MemberVO mVo = null;
    String sql = "select * from member where userid=?";
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
```

## 로그인 인증

---

### ❖ 회원 인증을 위한 MemberDAO 메서드 추가

- getMember() 메서드 추가 : id로 회원 정보 가져오는 메서드

```
try {
    conn = getConnection();
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, userid);
    rs = pstmt.executeQuery();
    if (rs.next()) {
        mVo = new MemberVO();
        mVo.setName(rs.getString("name"));
        mVo.setUserId(rs.getString("userid"));
        mVo.setPwd(rs.getString("pwd"));
        mVo.setEmail(rs.getString("email"));
        mVo.setPhone(rs.getString("phone"));
        mVo.setAdmin(rs.getInt("admin"));
    }
}
```



## 로그인 인증

---

### ❖ 회원 인증을 위한 MemberDAO 메서드 추가

- getMember() 메서드 추가 : id로 회원 정보 가져오는 메서드

```
        catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                if (rs != null)
                    rs.close();
                if (pstmt != null)
                    pstmt.close();
                if (conn != null)
                    conn.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return mVo;
    }
}
```

## 로그인 인증

---

### ❖ 회원 인증을 위한 서블릿 클래스

```
package com.saeyan.controller;
:
@WebServlet("/login.do")
public class LoginServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {
        String url = "member/login.jsp";

        String userid = request.getParameter("userid");
        String pwd = request.getParameter("pwd");

        MemberDAO mDao = MemberDAO.getInstance();
    }
}
```

## 로그인 인증

### ❖ 회원 인증을 위한 서블릿 클래스

```
int result = mDao.userCheck(userid, pwd);
if (result == 1) {
    MemberVO mVo = mDao.getMember(userid);

    HttpSession session = request.getSession();
    session.setAttribute("loginUser", mVo);
    request.setAttribute("message", "회원 가입에 성공했습니다.");
    url = "main.jsp";

} else if (result == 0) {
    request.setAttribute("message", "비밀번호가 맞지 않습니다.");
} else if (result == -1) {
    request.setAttribute("message", "존재하지 않는 회원입니다.");
}

RequestDispatcher dispatcher = request.getRequestDispatcher(url);
dispatcher.forward(request, response);
}
```

## 로그인 인증

### ❖ 회원 인증된 사용자에게 제공하는 JSP 페이지

- WebContent/main.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<c:if test="${empty loginUser}">
    <jsp:forward page='login.do' />
</c:if>
:
<body>
    <h2>회원 전용 페이지</h2>
    <form action="logout.do">
        <table>
            <tr>
                <td>안녕하세요. ${loginUser.name}(${loginUser.userid})님</td>
            </tr>
```

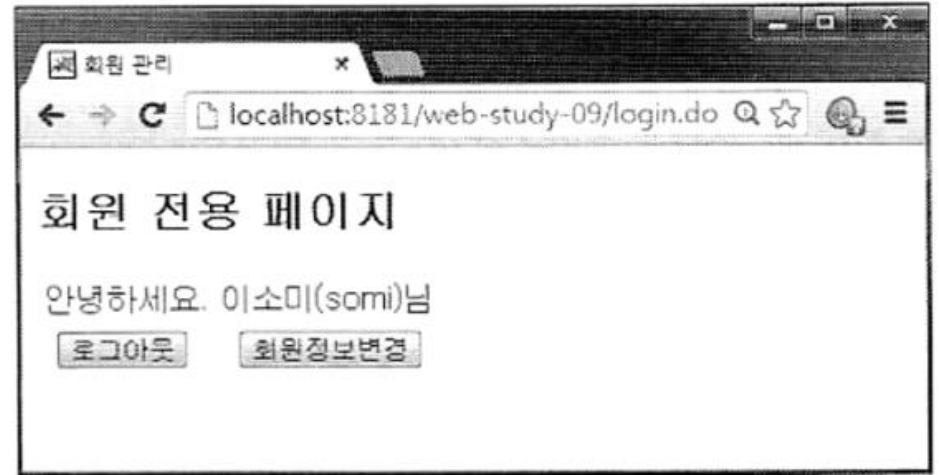
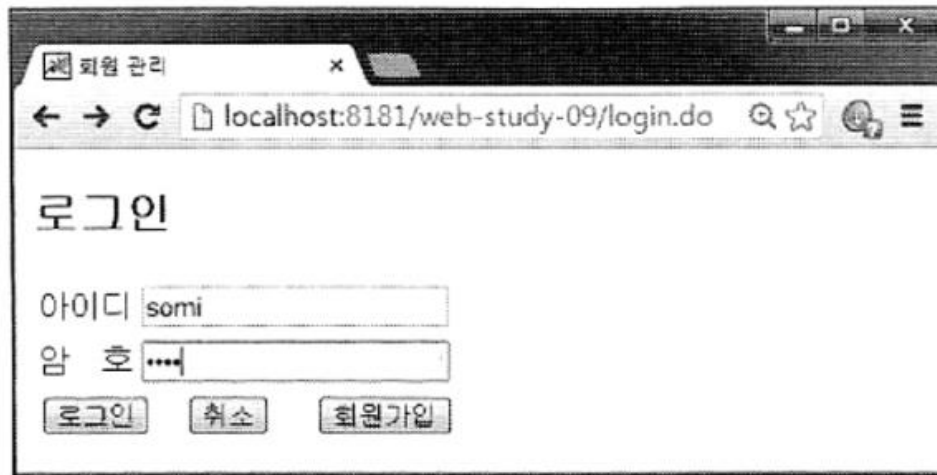
## ❖ 회원 인증된 사용자에게 제공하는 JSP 페이지

## ❖ 회원 인증된 사용자에게 제공하는 JSP 페이지

- ```
<tr>  
    <td colspan="2" align="center">  
        <input type="submit" value="로그아웃">   <br>  
        <input type="button" value="회원정보변경"  
onclick="location.href='memberUpdate.do?userid=${loginUser.userid}' ">  
    </td>  
</tr>  
</table>  
</form>
```
-

## 로그인 인증

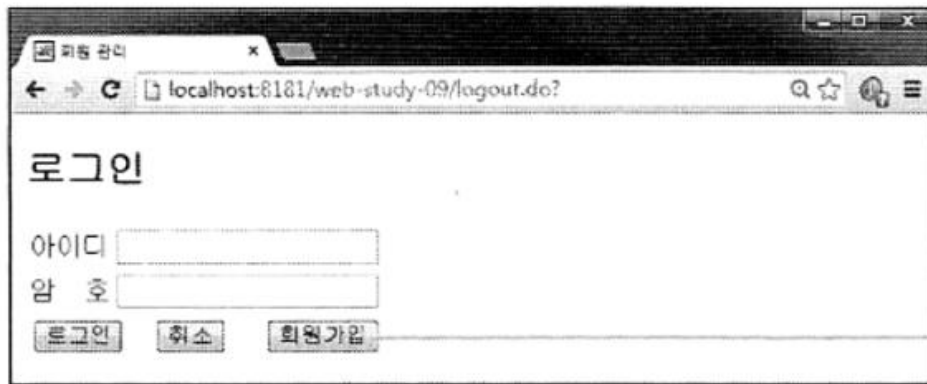
### ❖ 회원 인증된 사용자에게 제공하는 JSP 페이지



## 회원 가입

### ❖ 회원 가입을 위한 프로그래밍

[로그인 페이지 : login.jsp]



로그인

아이디

암 호

로그인 페이지의 <회원 가입> 버튼이  
클릭되면 join.do가 요청되어 join.jsp  
페이지로 이동

[회원 가입 페이지 : join.jsp]



회원 가입

\* 표시 항목은 필수 입력 항목입니다.

이름  \*

아이디  \*

암 호  \*

암호 확인  \*

이메일

전화번호

등급 ☒ 일반회원 ☐ 관리자

```
<input type="button" value="회원 가입" onclick="location.href='join.do'">
```

o join.do를 처리할 서블릿 구현

## 회원 가입

---

### ❖ 회원 정보 입력 폼을 위한 서블릿 클래스 : JoinServlet

- join.do의 GET 메서드 처리

```
package com.saeyan.controller;
:
@WebServlet("/join.do")
public class JoinServlet extends HttpServlet {
    :
    protected void doGet(HttpServletRequest request,
                          HttpServletResponse response)
        throws ServletException, IOException {
        RequestDispatcher dispatcher = request
            .getRequestDispatcher("member/join.jsp");
        dispatcher.forward(request, response);
    }

    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response)
        throws ServletException, IOException {
    }
}
```



## 회원 가입

### ❖ 회원 정보 입력 폼



The screenshot shows a web browser window with the title '회원 관리' and the address bar displaying 'localhost:8181/web-study-09/join.do'. The main heading is '회원 가입'. Below it, a note states: '\* 표시 항목은 필수 입력 항목입니다.' (Items marked with \* are required input items).

The form contains the following fields and controls:

- 이름** (Name): Text input with '전원지' entered. A '\*' indicates it is required.
- 아이디** (ID): Text input with 'onemouse' entered. A '\*' indicates it is required. To the right is a '중복 체크' (Check for duplicates) button.
- 암 호** (Password): Text input with four dots '....' entered. A '\*' indicates it is required.
- 암호 확인** (Confirm Password): Text input with four dots '....' entered. A '\*' indicates it is required.
- 이메일** (Email): Text input with 'onemouse@naver.com' entered.
- 전화번호** (Phone Number): Text input with '010-3333-4444' entered.
- 등급** (Grade): Radio buttons for '일반회원' (General Member) and '관리자' (Administrator). '일반회원' is selected.

At the bottom, there are two buttons: '확인' (Confirm) and '취소' (Cancel). A mouse cursor is pointing at the '확인' button.

# 회원 가입

## ❖ 회원 정보 입력 폼

- WebContent/member/join.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
:
<script type="text/javascript" src="script/member.js"></script>
</head>
<body>

    <h2>회원 가입</h2>
    '*' 표시 항목은 필수 입력 항목입니다.

    <form action="join.do" method="post" name="frm">
        <table>
            <tr>
                <td>이름</td>
                <td><input type="text" name="name" size="20">*</td>
            </tr>
```

## 회원 가입

### ❖ 회원 정보 입력 폼

○ WebContent/member/join.jsp

```
<tr>
  <td>아이디</td>
  <td>
    <input type="text" name="userid" size="20" id="userid">*
    <input type="hidden" name="reid" size="20">
    <input type="button" value="중복 체크"
      onclick="idCheck()">
  </td>
</tr>
<tr>
  <td>암 호</td>
  <td><input type="password" name="pwd" size="20">*</td>
</tr>
<tr height="30">
  <td width="80">암호 확인</td>
  <td><input type="password" name="pwd_check" size="20">*</td>
</tr>
```

## 회원 가입

### ❖ 회원 정보 입력 폼

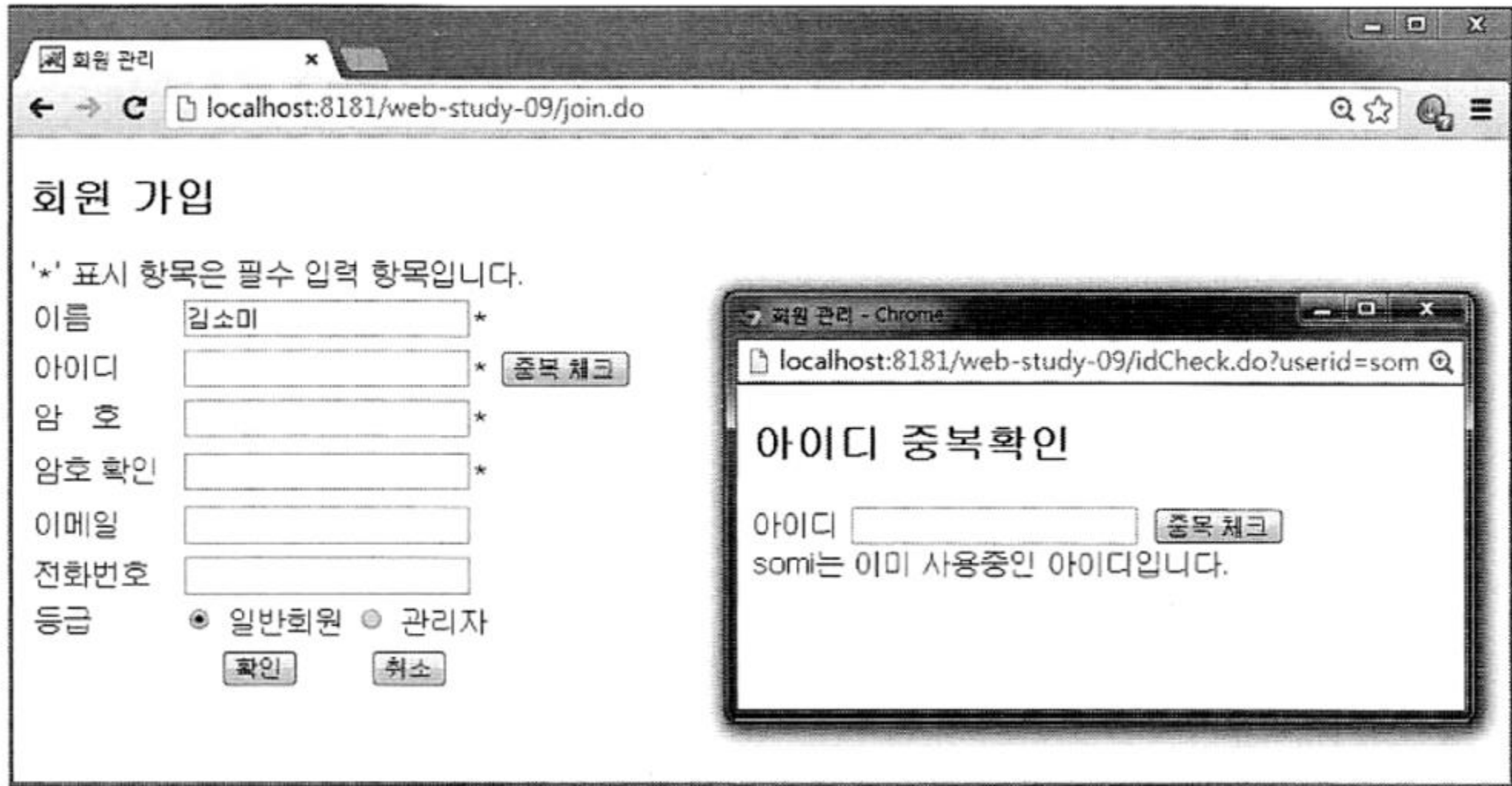
○ WebContent/member/join.jsp

```
<tr>
  <td>이메일</td>
  <td><input type="text" name="email" size="20"></td>
</tr>
<tr>
  <td>전화번호</td>
  <td><input type="text" name="phone" size="20"></td>
</tr>
<tr>
  <td>등급</td>
  <td>
    <input type="radio" name="admin" value="0"
      checked="checked"> 일반회원
    <input type="radio" name="admin" value="1"> 관리자</td>
</tr>
```



## 회원 가입

### ❖ ID 중복 체크 페이지 팝업



## 회원 가입

---

### ❖ ID 중복 체크 페이지를 팝업으로 띄우기 위한 자바스크립트

○ member.js

```
function idCheck() {  
    if (document.frm.userid.value == "") {  
        alert('아이디를 입력하여 주십시오.');        document.formm.userid.focus();  
        return;  
    }  
  
    var url = "idCheck.do?userid=" + document.frm.userid.value;  
  
    window.open(url, "_blank_1",  
        "toolbar=no, menubar=no, scrollbars=yes, resizable=no, width=450,  
height=200");  
}
```

## 회원 가입

---

### ❖ ID 중복 체크 MemberDAO 메서드 추가

리턴형	메소드
<code>int confirmID(String userid)</code>	회원 가입시 아이디 중복을 확인할 때 사용한다. 해당 아이디가 있으면 1을, 없으면 -1을 리턴한다.



## 회원 가입

---

### ❖ ID 중복 체크 MemberDAO 메서드 추가

```
public int confirmID(String userid) {
    int result = -1;
    String sql = "select userid from member where userid=?";
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, userid);
        rs = pstmt.executeQuery();
        if (rs.next()) {
            result = 1;
        } else {
            result = -1;
        }
    }
}
```

### ❖ ID 중복 체크 MemberDAO 메서드 추가

```
        catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                if (rs != null)
                    rs.close();
                if (pstmt != null)
                    pstmt.close();
                if (conn != null)
                    conn.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return result;
    }
}
```

## 회원 가입

---

### ❖ ID 중복 체크 서블릿

- `var url = "idCheck.do?userid=" + document.frm.userid.value;`
  - 요청 URL : `idCheck.do`
  - 요청 메서드 : `GET`
  - 요청 파라미터 : `userid`

## 회원 가입

---

### ❖ ID 중복 체크 서블릿

```
package com.saeyan.controller;
:
@WebServlet("/idCheck.do")
public class IdCheckServlet extends HttpServlet {
    :
    protected void doGet(HttpServletRequest request,
                        HttpServletResponse response)
                        throws ServletException, IOException {

        String userid = request.getParameter("userid");

        MemberDAO mDao = MemberDAO.getInstance();
        int result = mDao.confirmID(userid);

        request.setAttribute("userid", userid);
        request.setAttribute("result", result);

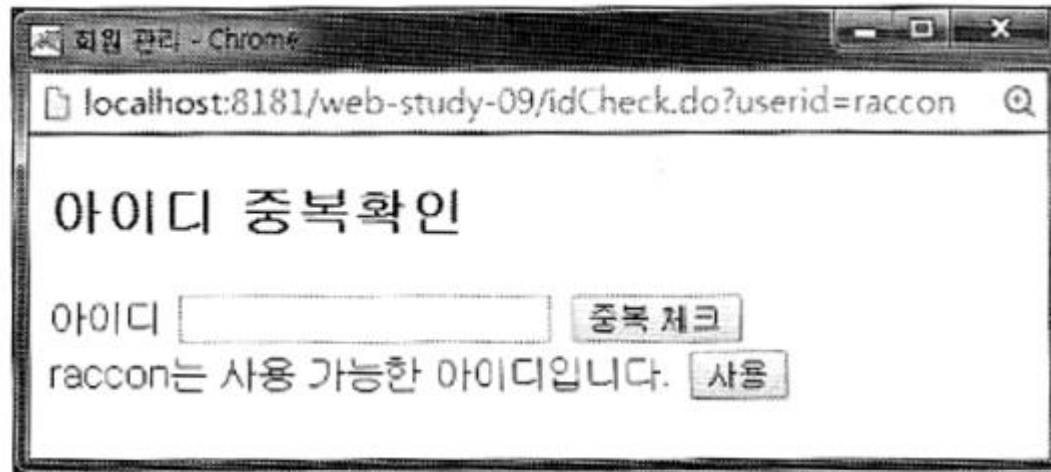
        RequestDispatcher dispatcher = request
            .getRequestDispatcher("member/idcheck.jsp");
        dispatcher.forward(request, response);
    }
}
```

## 회원 가입

---

### ❖ ID 중복 체크 팝업 JSP 페이지

- WebContent/member/idcheck.jsp



## 회원 가입

### ❖ ID 중복 체크 팝업 JSP 페이지

- WebContent/member/idcheck.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
:
<script type="text/javascript" src="script/member.js"></script>
</head>
<body>
    <h2>아이디 중복확인</h2>

    <form action="idCheck.do" method="get" name="frm">
        아이디 <input type="text" name="userid" value="${userid}">
        <input type="submit" value="중복 체크"> <br>
        <c:if test="${result == 1}">
            <script type="text/javascript">
                opener.document.frm.userid.value = "";
            </script>
            ${userid}는 이미 사용 중인 아이디입니다.
        </c:if>
```

## 회원 가입

---

### ❖ ID 중복 체크 팝업 JSP 페이지

- WebContent/member/idcheck.jsp

```
<c:if test="${result==-1}">
    ${userid}는 사용 가능한 아이디입니다.
    <input type="button" value="사용" class="cancel"
        onclick="idok('${userid}')">
</c:if>
</form>
</body>
</html>
```

## 회원 가입

---

### ❖ ID 중복 체크 완료 처리 자바스크립트 함수

```
<input type="button" value="사용" class="cancel"
      onclick="idok('${userid}')">
```

○ WebContent/script/member.js

```
function idok(userid) {
    opener.frm.userid.value = document.frm.userid.value;
    opener.frm.reid.value = document.frm.userid.value;
    self.close();
}
```



## 회원 가입

### ❖ 회원 정보의 유효성을 체크하기 위한 자바스크립트 함수

```
<input type="submit" value="확인" onclick="return joinCheck()">
```

```
function joinCheck() {  
    if (document.frm.name.value.length == 0) {  
        alert("이름을 써주세요.");  
        frm.name.focus();  
        return false;  
    }  
  
    if (document.frm.userid.value.length == 0) {  
        alert("아이디를 써주세요");  
        frm.userid.focus();  
        return false;  
    }  
  
    if (document.frm.userid.value.length < 4) {  
        alert("아이디는 4글자이상이어야 합니다.");  
        frm.userid.focus();  
        return false;  
    }  
}
```

## 회원 가입

### ❖ 회원 정보의 유효성을 체크하기 위한 자바스크립트 함수

```
if (document.frm.pwd.value == "") {  
    alert("암호는 반드시 입력해야 합니다.");  
    frm.pwd.focus();  
    return false;  
}  
  
if (document.frm.pwd.value != document.frm.pwd_check.value) {  
    alert("암호가 일치하지 않습니다.");  
    frm.pwd.focus();  
    return false;  
}  
  
if (document.frm.reid.value.length == 0) {  
    alert("중복 체크를 하지 않았습니다.");  
    frm.userid.focus();  
    return false;  
}  
return true;  
}
```

## 회원 가입

---

### ❖ 회원 정보를 DB에 추가하기 위한 메서드 추가

#### ○ MemberDAO.java

```
public int insertMember(MemberVO mVo) {
    int result = -1;
    String sql = "insert into member values(?, ?, ?, ?, ?, ?)";
    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, mVo.getName());
        pstmt.setString(2, mVo.getUserid());
        pstmt.setString(3, mVo.getPwd());
        pstmt.setString(4, mVo.getEmail());
        pstmt.setString(5, mVo.getPhone());
        pstmt.setInt(6, mVo.getAdmin());
        result = pstmt.executeUpdate();
    }
}
```

## 회원 가입

---

### ❖ 회원 정보를 DB에 추가하기 위한 메서드 추가

#### ○ MemberDAO.java

```
        catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                if (pstmt != null)
                    pstmt.close();
                if (conn != null)
                    conn.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return result;
    }
}
```

## 회원 가입

---

### ❖ 회원 가입 처리 서블릿

```
<form action="join.do" method="post" name="frm">
```

- join.do 처리 서블릿
- doPost() 구현

# 회원 가입

---

## ❖ 회원 가입 처리 서블릿

```
package com.saeyan.controller;
:
@WebServlet("/join.do")
public class JoinServlet extends HttpServlet {
    :
    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {

        request.setCharacterEncoding("UTF-8");

        String name = request.getParameter("name");
        String userid = request.getParameter("userid");
        String pwd = request.getParameter("pwd");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");
        String admin = request.getParameter("admin");
```

## 회원 가입

### ❖ 회원 가입 처리 서블릿

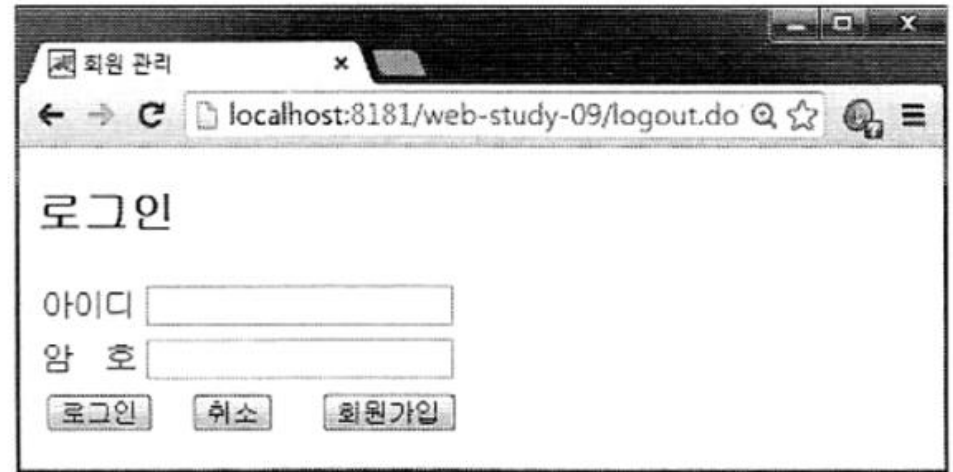
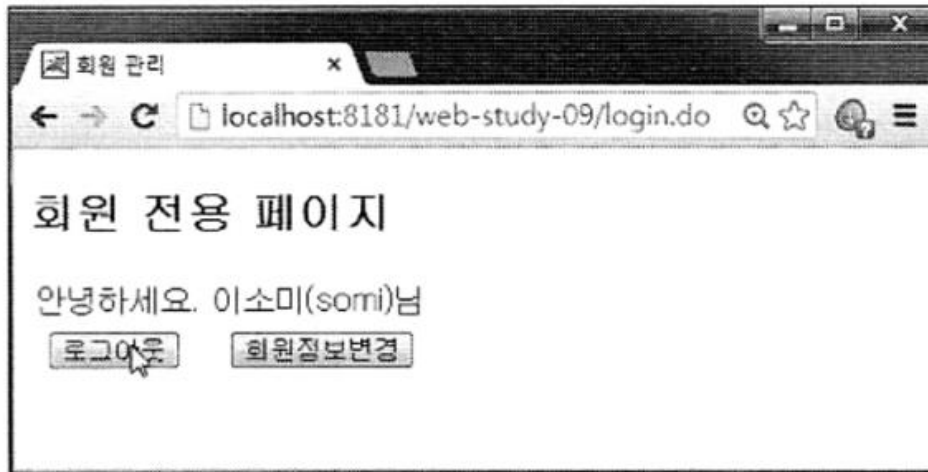
```
MemberVO mVo = new MemberVO();
mVo.setName(name);
mVo.setUserid(userid);
mVo.setPwd(pwd);
mVo.setEmail(email);
mVo.setPhone(phone);
mVo.setAdmin(Integer.parseInt(admin));

MemberDAO mDao = MemberDAO.getInstance();
int result = mDao.insertMember(mVo);
HttpSession session = request.getSession();
if (result == 1) {
    session.setAttribute("userid", mVo.getUserid());
    request.setAttribute("message", "회원 가입에 성공했습니다.");
} else {
    request.setAttribute("message", "회원 가입에 실패했습니다.");
}
RequestDispatcher dispatcher = request
    .getRequestDispatcher("member/login.jsp");
dispatcher.forward(request, response);
}
```

# 로그아웃

## ❖ 로그아웃

- 요청 URL : logout.do
- 요청 메서드 : GET
- 로그아웃 프로그래밍
  - 세션에서 로그인 정보 삭제(무효화)





# 로그아웃

---

## ❖ 로그아웃 서블릿

```
package com.saeyan.controller;
:
@WebServlet("/logout.do")
public class LogoutServlet extends HttpServlet {
    :
    protected void doGet(HttpServletRequest request,
                        HttpServletResponse response)
                        throws ServletException, IOException {

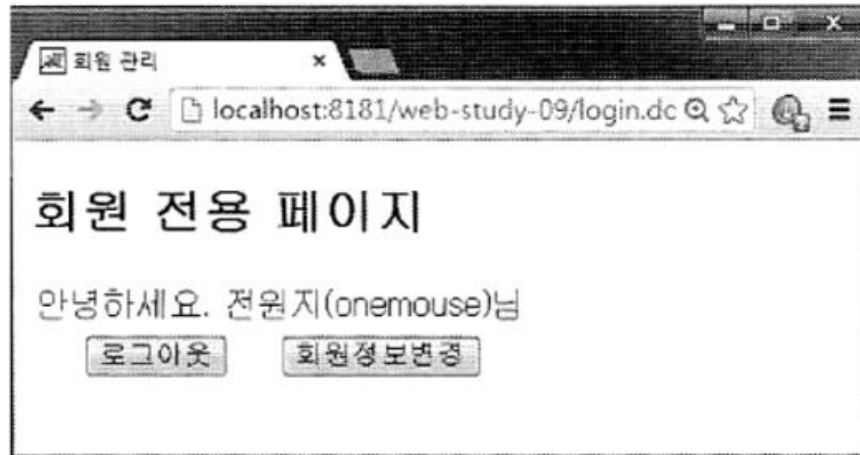
        HttpSession session = request.getSession();
        session.invalidate();

        RequestDispatcher dispatcher = request
            .getRequestDispatcher("member/login.jsp");
        dispatcher.forward(request, response);

    }
}
```

## 회원 정보 수정

### ❖ 회원 정보 수정



## 회원 정보 수정

---

### ❖ 회원정보 수정

```
<input type="button" value="회원정보변경"  
      onclick="location.href='memberUpdate.do?userid=${loginUser.userid}'">
```

- 요청 URL : memberUpdate.do
- 요청 메서드
  - GET : 회원 정보 수정 폼 제시
    - 회원 정보를 DB에서 추출하여 설정
  - POST : 회원 정보 수정 수행

## 회원 정보 수정

---

### ❖ 회원정보 수정 폼 처리 서블릿

- MemberUpdateServlet.java

```
package com.saeyan.controller;
:
@WebServlet("/memberUpdate.do")
public class MemberUpdateServlet extends HttpServlet {
    :
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException
    {
        String userid = request.getParameter("userid");

        MemberDAO mDao = MemberDAO.getInstance();

        MemberVO mVo = mDao.getMember(userid);
        request.setAttribute("mVo", mVo);

        RequestDispatcher dispatcher = request
            .getRequestDispatcher("member/memberUpdate.jsp");
        dispatcher.forward(request, response);
    }
}
```

## 회원 정보 수정

### ❖ 회원 정보 수정 페이지

○ WebContent/member/memberUpdate.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
:
<script type="text/javascript" src="script/member.js"></script>
</head>
<body>
    <h2>회원 수정</h2>
    <form action="memberUpdate.do" method="post" name="frm">
        <table>
            <tr>
                <td>이름</td>
                <td><input type="text" name="name" size="20"
                    value="${mVo.name}" readonly></td>
            </tr>
            <tr>
                <td>아이디</td>
                <td><input type="text" name="userid" size="20"
                    value="${mVo.userid}" readonly></td>
            </tr>
        </table>
    </form>
</body>
```

## 회원 정보 수정

---

### ❖ 회원 정보 수정 페이지

○ WebContent/member/memberUpdate.jsp

```
<tr>
    <td>암 &nbsp; 호</td>
    <td><input type="password" name="pwd" size="20">*</td>
</tr>
<tr height="30">
    <td width="80">암호 확인</td>
    <td><input type="password" name="pwd_check" size="20">*</td>
</tr>
<tr>
    <td>이메일</td>
    <td><input type="text" name="email" size="20"
        value="${mVo.email}"></td>
</tr>
<tr>
    <td>전화번호</td>
    <td><input type="text" name="phone" size="20"
        value="${mVo.phone}"></td>
</tr>
```

## 회원 정보 수정

### ❖ 회원 정보 수정 페이지

○ WebContent/member/memberUpdate.jsp

```
<tr>
  <td>등급</td>
  <td><c:choose>
    <c:when test="${mVo.admin==0}">
      <input type="radio" name="admin" value="0"
        checked="checked"> 일반회원
      <input type="radio" name="admin" value="1"> 관리자
    </c:when>
    <c:otherwise>
      <input type="radio" name="admin" value="0"> 일반회원
      <input type="radio" name="admin" value="1"
        checked="checked"> 관리자
    </c:otherwise>
  </c:choose>
</td>
</tr>
```





### ❖ 회원 정보 수정을 위한 MemberDAO 메서드 추가

```
public int updateMember(MemberVO mVo) {
    int result = -1;
    String sql = "update member set pwd=?, email=?,"
        + "phone=?, admin=? where userid=?";
    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, mVo.getPwd());
        pstmt.setString(2, mVo.getEmail());
        pstmt.setString(3, mVo.getPhone());
        pstmt.setInt(4, mVo.getAdmin());
        pstmt.setString(5, mVo.getUserid());
        result = pstmt.executeUpdate();
    }
}
```

## 회원 정보 수정

---

### ❖ 회원 정보 수정을 위한 MemberDAO 메서드 추가

```
        catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                if (pstmt != null)
                    pstmt.close();
                if (conn != null)
                    conn.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return result;
    }
}
```

## 회원 정보 수정

---

### ❖ 회원정보 수정 처리 서블릿

```
package com.saeyan.controller;
:
@WebServlet("/memberUpdate.do")
public class MemberUpdateServlet extends HttpServlet {
    :
    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {

        request.setCharacterEncoding("UTF-8"); // 한글 깨짐을 방지

        // 폼에서 입력한 회원 정보 얻어오기
        String userid = request.getParameter("userid");
        String pwd = request.getParameter("pwd");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");
        String admin = request.getParameter("admin");
```

## 회원 정보 수정

---

### ❖ 회원정보 수정 처리 서블릿

```
// 회원 정보를 저장할 객체 생성
MemberVO mVo = new MemberVO();
mVo.setUserid(userid);
mVo.setPwd(pwd);
mVo.setEmail(email);
mVo.setPhone(phone);
mVo.setAdmin(Integer.parseInt(admin));

MemberDAO mDao = MemberDAO.getInstance();
mDao.updateMember(mVo);

response.sendRedirect("login.do");
}

}
```

## 회원 정보 수정

---

### ❖ 회원정보 수정 처리

- 정보 수정 후 redirect로 login.jsp 이동
- 이미 로그인 된 상태이므로 login 페이지로 이동하는 것이 아니라
- main.jsp로 이동하도록 처리
  - login.do의 GET 요청에서 세션 정보 확인
  - 세션 정보가 있다면 main.jsp로 이동
  - 세션 정보가 없다면 login.jsp로 이동

## 회원 정보 수정

---

### ❖ 로그인 정보에 따라 이동 페이지 결정

#### ○ LoginServlet.java

```
@WebServlet("/login.do")
public class LoginServlet extends HttpServlet {
    :
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException
    {
        String url = "member/login.jsp";
        HttpSession session = request.getSession();

        // 이미 로그인 된 사용자이면
        if (session.getAttribute("loginUser") != null) {
            url = "main.jsp"; // 메인 페이지로 이동한다.
        }
        RequestDispatcher dispatcher = request.getRequestDispatcher(url);
        dispatcher.forward(request, response);
    }
    :
}
```