

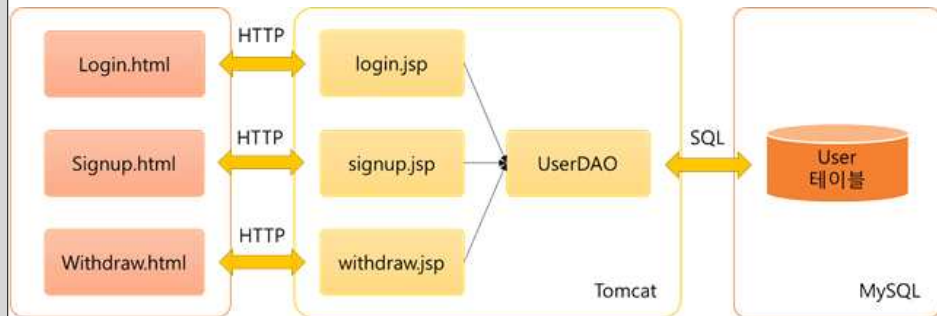
■ 서식 1 : 결과보고서 요약본

Project Based Learning 결과보고서 요약본

교과목명	국문	웹앱프로그래밍		
	영문	Web App Programming		
PBL 관련 능력단위 (능력단위코드)		PBL 관련 능력단위요소 (능력단위요소코드)		
웹프레임워크 (YCS-2001020103_20v1)		HTTP 이해하기		
		웹프레임워크 이용하기		
학년 반	3-1	조원	이은우 외 1명	
프로젝트 주제	모바일 앱 작성하기			
지도교수	심효선	산업체 참가여부	<input type="checkbox"/> 유 <input checked="" type="checkbox"/> 무	
참여학생	이은우, 임현수			
<p>작품 개요 (주제 선정 이유)</p>		<p>기계학습 서비스를 서버를 통해 제공하기 위한 기본 기술중 하나인 서버 프로그래밍을 배울수 있는 것을 목표로 한다. 클라이언트는 휴대폰 이라고 가정하고 서버에서는 휴대폰용의 UI를 제작한다.</p> <p>환경 :</p> <ul style="list-style-type: none"> - JDK - 크롬 브라우저 - 톰캣 - 이클립스 - MySQL <p>기능</p> <ul style="list-style-type: none"> - 회원가입/회원탈퇴 - 로그인/로그아웃 - 블로그등록/수정/삭제 - 블로그 내용 표시 - 블로그 내용에는 사진 포함 - 화면의 UI는 휴대폰 기준 - 데이터 관리는 DB를 사용하여야 함 - 추가기능을 구현하면 가산점 부여 		

작품 구조도
(문제점 제시 및 개선방안)

1) 서버 프로그램의 구조



Browsers

2) UI 구조



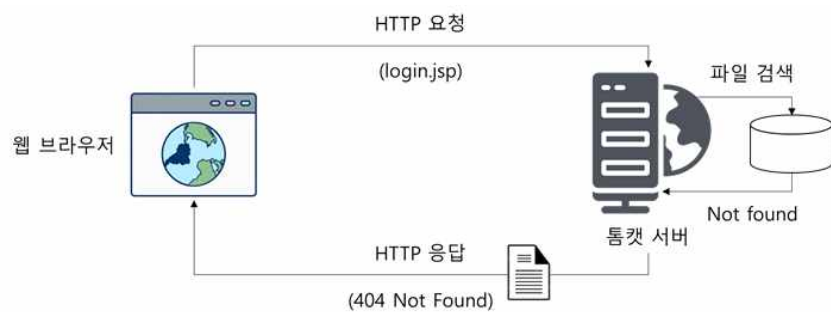
3) 클라이언트 프로그램의 구조 (제공됨)

WebView 기술을 이용하여 앱처럼 구현하고 IP 주소를 입력할 수 있도록 UI를 구성한다

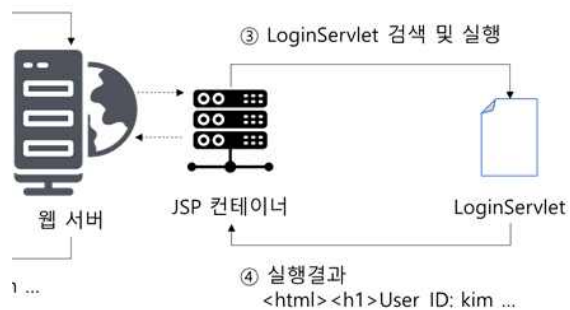


관련 이론

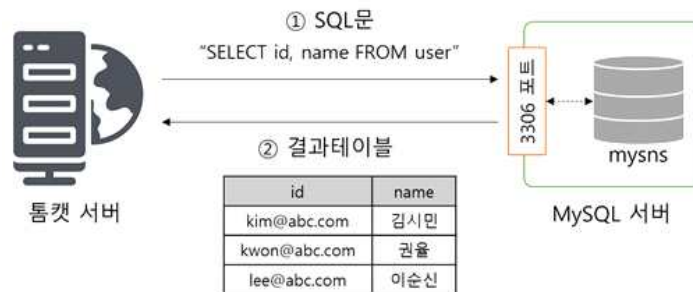
1. HTTP 통신



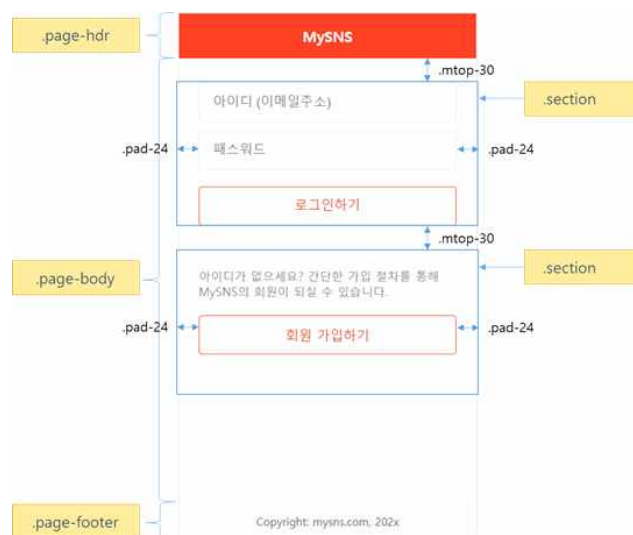
2. Servlet



3. Database



4. Mobile Design



결과물 제작
(문제점 개선사항)

1. USER TABLE에 최소한 한 개의 필드 추가

- desc user; 명령의 실행결과 화면 캡처
- 회원 가입/탈퇴, 로그인/로그아웃, 블로그등록/수정/삭제, 추가기능에 추가된 필드에 해당하는 UI 추가 하고 화면 캡처
- 주요 코드가 있으면 간단히 설명

◆ 필드 phonenum 추가

- 아이디 찾기와 비밀번호 찾기에서 이름만 넣기에는 식상하기에 휴대폰 번호를 이용하여 찾을 수 있게 회원가입시 사용자의 휴대폰 번호도 사용하도록 한다.

▶ desc user;을 사용하여 확인

```
mysql> desc user;
```

Field	Type	Null	Key	Default	Extra
id	varchar(128)	NO	PRI	NULL	
password	varchar(32)	YES		NULL	
name	varchar(32)	YES		NULL	
ts	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
phonenum	varchar(128)	NO		NULL	

5 rows in set (0.01 sec)

```
mysql> ALTER TABLE user ADD COLUMN phonenum varchar(128) NOT NULL;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> select * from user;
```

id	password	name	ts	phonenum
kim@abc.com	111	김시민	2025-05-18 18:45:22	
kwon@abc.com	111	권을	2025-05-18 18:45:22	
lee@abc.com	111	이순신	2025-05-18 18:45:22	

3 rows in set (0.00 sec)

`ALTER TABLE user ADD COLUMN phonenum varchar(128) NOT NULL;`

sql문을 추가하여 MYSQL에 직접 추가하는 방법과 mysns.sql에서 다음과 같이 추가할 컬럼에 대해 sql구문을 작성하는 방법도 있다.

```
CREATE TABLE IF NOT EXISTS user(
    id VARCHAR(128) PRIMARY KEY, -- "email"
    password VARCHAR(32),
    name VARCHAR(32),
    phonenum VARCHAR(128),
    ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

또 이 필드를 추가 하기 위해 아래의 파일들을 수정함.

▶ signup.html = 휴대폰번호 입력란 추가

```
<input name="id" type="text" placeholder="아이디 (이메일주소)" required>
<input name="phonenum" type="text" placeholder="휴대폰번호" required>
<input name="ps" type="password" class="mtop-10" placeholder="패스워드" required>
<input name="ps2" type="password" class="mtop-10" placeholder="패스워드 (확인)">
<input name="name" type="text" class="mtop-10" placeholder="이름">
<input type="submit" class="mtop-20" value="회원가입하기">
```

▶ signup.jsp = uphone이라는 파라미터 저장 변수 생성

```
String uname = request.getParameter("name");
String uphone = request.getParameter("phonenum");

UserDAO dao = new UserDAO();
if (dao.exists(uid)) {
    out.print("이미 가입한 회원입니다.");
    return;
}

if (dao.insert(uid, upass, uname, uphone)) {
    //out.print("회원 가입이 완료되었습니다.");
    session.setAttribute("id", uid);
    response.sendRedirect("main.jsp");
}
```

▶ UserDAO.java = signup.jsp에서 받은 uphone 파라미터 변수를 insert 함수에 추가

```
public boolean insert(String uid, String upass, String uname, String uphone) throws NamingException {
    Connection conn = ConnectionPool.get();
    PreparedStatement stmt = null;
    try {
        String sql = "INSERT INTO user(id, password, name, phonenum) VALUES(?, ?, ?, ?)";
        stmt = conn.prepareStatement(sql);
        stmt.setString(1, uid);
        stmt.setString(2, upass);
        stmt.setString(3, uname);
        stmt.setString(4, uphone);
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        if (stmt != null) stmt.close();
    }
    return true;
}
```

결과 화면

← → ↻ 🏠 localhost:8080/hello-ch14/signup.html

MySNS

아이디 (이메일주소)

휴대폰번호

패스워드

패스워드 (확인)

이름

[회원이입하기](#)

회원 가입시 MySNS의 약관에 동의하신 것으로 간주합니다.

회원정보 입력시 결과 (+ DB user 테이블 상태)

lew@abc.com

010-1234-5678

....

....

이은우

```
mysql> select * from user;
```

id	password	name	ts	phonenum
kim@abc.com	111	김시민	2025-05-18 18:45:22	
kwon@abc.com	111	권율	2025-05-18 18:45:22	
lee@abc.com	111	이순신	2025-05-18 18:45:22	
lew@abc.com	1234	이은우	2025-06-08 01:35:45	010-1234-5678

```
4 rows in set (0.00 sec)
```

2. Feed TABLE에 최소한 한 개의 필드 추가

- desc feed; 명령의 실행결과 화면 캡처
- 회원 가입/탈퇴, 로그인/로그아웃, 블로그등록/수정/삭제, 추가기능에 추가된 필드에 해당하는 UI 추가 하고 화면 캡처
- 주요 코드가 있으면 간단히 설명

◆ 필드 video 추가

- feed를 하나 추가할 때 원래는 id(아이디), content(내용), image(사진)이 있는데, 동영상 파일도 넣기 위해 새로운 필드로 video를 추가하였다.

▶ desc user;을 사용하여 확인

```
mysql> desc feed;
```

Field	Type	Null	Key	Default	Extra
no	int unsigned	NO	PRI	NULL	auto_increment
id	varchar(128)	YES		NULL	
content	varchar(4096)	YES		NULL	
ts	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
images	varchar(1024)	YES		NULL	
video	varchar(1024)	YES		NULL	

6 rows in set (0.00 sec)

```
mysql> select * from feed;
```

no	id	content	ts	images	video
1	kim@abc.com	Hello	2025-06-11 11:01:50	NULL	NULL
2	kwon@abc.com	Aloha	2025-06-11 11:01:50	NULL	NULL
3	lew@abc.com	안녕하세요	2025-06-11 11:31:54	다운로드.jpg	1000012997.mp4

3 rows in set (0.00 sec)

Feed 부분은 video 필드를 추가하기 위해 mysns.sql에서 “video VARCHAR(1024)”를 추가한다.

```
CREATE TABLE IF NOT EXISTS feed(  
    no INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    id VARCHAR(128), -- "email"  
    content VARCHAR(4096),  
    ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    images VARCHAR(1024),  
    video VARCHAR(1024)  
);
```

또 이 필드를 추가 하기 위해 아래의 파일들을 수정함.

▶ feedAdd.html = 비디오파일 첨부 추가

```
<form action="/jsp/feedAdd4.jsp" method="post" enctype="multipart/form-data">  
    <div class="page-hdr">MySNS</div>  
    <div class="page-body">  
        <div class="section pad-24 mtop-30">  
            <input name="id" type="text" placeholder="아이디 (이메일주소)">  
            <textarea name="content" class="mtop-10" placeholder="이곳에 글을 작성해 주세요."></textarea>  
            <div class="section pad-4 mtop-30">  
                <div class="desc">아래에서 이미지를 선택해주세요.</div>  
                <input type="file" name="image">  
            </div>  
            <div class="section pad-4 mtop-30">  
                <div class="desc">비디오를 선택해주세요.</div>  
                <input type="file" name="video" accept="video/*">  
            </div>  
            <input type="submit" class="mtop-30" value="업로드하기">  
        </div>  
    </div>  
    <div class="page-footer">Copyright: mysns.com, 202x</div>  
</form>
```

▶ feedAdd4.jsp = 비디오 첨부 처리 코드 [이미지 파일 처리와 동일]

```
else if (name.equals("video")) {
    vfname = item.getName();
    if (vfname != null && !vfname.equals("")) {
        Path pv = Paths.get(vfname);
        String vfileName = pv.getFileName().toString();
        vfile = item.get();
        String root = application.getRealPath(java.io.File.separator);
        FileUtil.saveImage(root, vfileName, vfile); // 영상도 저장!
    }
}

String videofile = "";
if (vfname != null && !vfname.trim().isEmpty()) {
    Path pv = Paths.get(vfname);
    videofile = pv.getFileName() != null ? pv.getFileName().toString() : "";
}
```

업로드한 비디오 파일이 있을 때 vfname에서 진짜 “파일명”만 뽑아 videofile 변수에 저장하는 코드 추가

```
FeedDAO dao = new FeedDAO();
boolean result;
try {
    result = dao.insert(uid, ucon, file, videofile); // 이미지/비디오 모두 DB 저장
} catch (Exception e) {
    result = false;
    out.print("DB 오류: " + e.getMessage());
}

if (result) {
    response.sendRedirect("main.jsp");
} else {
    out.print("작성 글의 업로드 중 오류가 발생하였습니다.");
}
<>
```

FeedDAO.java에서 insert() 메소드를 불러올 때 첨부한 비디오파일도 받기 위해 videofile 변수 추가함.

▶ FeedDAO.java = feedAdd4.jsp에서 받은 video파라미터 변수들을 insert 함수에 추가

```
public boolean insert(String uid, String ucon, String uimages, String video) throws Na
Connection conn = ConnectionPool.get();
PreparedStatement stmt = null;
try {
    String sql = "INSERT INTO feed(id, content, images, video) VALUES(?, ?, ?, ?)";
    stmt = conn.prepareStatement(sql);
    stmt.setString(1, uid);
    stmt.setString(2, ucon);
    stmt.setString(3, uimages);
    stmt.setString(4, video);
}
```

▶ FeedObj.java = FeedObj클래스에도 video 필드를 추가해준다.


```

public ArrayList<FeedObj> getList2() throws NamingException, SQLException {
    Connection conn = ConnectionPool.get();
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        String sql = "SELECT * FROM feed ORDER BY ts DESC";
        stmt = conn.prepareStatement(sql);
        rs = stmt.executeQuery();

        ArrayList<FeedObj> feeds = new ArrayList<FeedObj>();
        while(rs.next()) {
            feeds.add(new FeedObj(rs.getString("id"), rs.getString("content"), rs.getString("ts"),
                rs.getString("images"), rs.getString("video")));
        }
        return feeds;
    } finally {
        if (rs != null) rs.close();
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
    }
}

package dao;

public class FeedObj {
    private String id, content, ts, images, video;

    public FeedObj(String id, String content, String ts, String images, String video) {
        this.id = id;
        this.content = content;
        this.ts = ts;
        this.images = images;
        this.video = video;
    }

    public String getId() { return this.id; }
    public String getContent() { return this.content; }
    public String getTs() { return this.ts; }
    public String getImages() { return this.images; }
    public String getVideo() { return video; }
}

```

▶ main.jsp = feed 테이블에 저장된 데이터들, 그러니까 등록된 게시물들을 보여주는 main.jsp 화면에 새로 추가된 필드인 video도 보여주기 위해 추가적으로 수정한다.

```

for (FeedObj feed : feeds) {
    String img = feed.getImages();
    String video = feed.getVideo();
    String videostr = "";

    if (img != null && !img.trim().isEmpty()) {
        imgstr = "<img src='../images/' + img + \"' width=240>\";
    }

    if (video != null && !video.trim().isEmpty()) {
        videostr = "<video width='240' controls><source src='../images/' + video + \"'></video>\";
    }

    str += "<tr><td colspan=2><hr></td></tr>\";
    str += "<tr>\";
    str += "<td><small>\" + feed.getId() + "</small></td>\";
    str += "<td><small>&nbsp;\" + feed.getTs() + "</small></td>\";
    str += "</tr>\";
    str += "<tr><td colspan=2>\" + imgstr + videostr + "</td></tr>\";
    str += "<tr><td colspan=2>\" + feed.getContent() + "</td></tr>\";
}

```

추가적으로 로그인한 사용자가 누구인지를 식별하기 위해

```

str += "<tr><td colspan='2' align='right'><b>로그인한 사용자: \" + uid +
\"</b></td></tr>\";

```

를 통해 로그인한 사용자의 id도 출력하게 해준다.

```
String str = "<table align=center>";
str += "<tr height=40><td><b>작성글 리스트</b></td>";
str += "<td align=right>";
str += "<a href='../feedAdd.html'><button>글쓰기</button></a>";
str += "<td align=right>";
str += "<a href='logout.jsp'><button>로그아웃</button></a>";
str += "</td></tr>";
str += "<tr><td colspan='2' align='right'><b>로그인한 사용자: " + uid + "</b></td></tr>";|
```

작성글 리스트

글쓰기

로그아웃

로그인한 사용자: lew@abc.com

이렇게 하면 login.html에서 로그인을 하여 정상적인 로그인 이 되면 main.jsp를 통해 feed테이블에 등록된 게시물의 내용(id, content, images, video, ts)들을 볼 수 있다.

MySNS

lew@abc.com

안녕하세요

아래에서 이미지를 선택해주세요.

파일 선택 다운로드.jpg

아래에서 비디오를 선택해주세요.

파일 선택 1000012997.mp4

업로드하기

Copyright: mysns.com, 202x

글쓰기를 누르면 작성자 id를 적는 칸, comment내용을 적는 칸, 이미지를 첨부하는 파일 선택 버튼, 비디오를 첨부하는 파일 선택 버튼이 존재 하고, 이들을 선택하여 업로드하게 되면,

다음과 같이 게시글이 업로드 된다.

동시에 DB로도 등록이 되었음을 확인할 수 있다.

```
mysql> select * from feed;
```

no	id	content	ts	images	video
1	kim@abc.com	Hello	2025-06-15 12:28:45	NULL	NULL
2	kwon@abc.com	Aloha	2025-06-15 12:28:45	NULL	NULL
3	lew@abc.com	안녕하세요	2025-06-15 14:14:37	다운로드.jpg	1000012997.mp4
4	lew@abc.com	테스트입니다.	2025-06-15 14:18:24	다운로드.jpg	1000012997.mp4

동시에 main.jsp 페이지에서 로그아웃 버튼을 누르게 되면

lew@abc.com

(2025-06-16 14:40:55)



안녕하세요

작성글 리스트

글쓰기

로그아웃

로그인한 사용자: lew@abc.com

로그아웃을 완료하였습니다.

로그인하기

다시 로그인하려 갈 수 있는 버튼을 만들었다.

4. 새로운 테이블을 추가한 추가기능 (가산점)

테이블을 추가한 추가기능은 아니나. 새로운 html, jsp, dao 파일 추가 클래스를 이용하여 잃어버린 ID 및 PW를 찾는 기능을 구현한다.

■ id, pw 찾기 기능 추가

◆findID(아이디 찾기)

- **UserDAO.java** = findID라는 클래스를 생성. 먼저 findID.html에서 이름과 휴대폰번호를 이용하여 잃어버린 ID를 찾는데 이때 입력 값이 findID.jsp 파일에서 요청하여 응답할 때 이용된다.

Mysns DB에서 user테이블에 접근하여 입력받은 이름을 검색하여 일치하는 유저가 있는지 확인하고, 유저가 존재하면 그 유저의 id값을 넘겨준다.

```
public String findId(String name, String phonenum) throws NamingException, SQLException {
    String sql = "SELECT id FROM user WHERE name=? AND phonenum = ?";
    Connection conn = ConnectionPool.get();
    PreparedStatement stmt = null;
    try {
        stmt = conn.prepareStatement(sql);
        stmt.setString(1, name);
        stmt.setString(2, phonenum);
        ResultSet rs = stmt.executeQuery();
        if(rs.next()) {
            return rs.getString("id");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
    }
    return null;
}
```

- **findID.html** = 먼저 CSS 설정, 그 외 인코딩 파일 설정은 다른 html 파일과 동일하다. 눈여겨 볼 것은 이름을 입력하는 것과 휴대전화 번호를 입력하고 아이디 찾기 버튼이다. 그리고 밑 부분에 비밀번호도 바로 찾을 수 있게, findPW.html로 이동하는 버튼을 추가하였다.

```

<form action="/jsp/findID.jsp" method="post">
  <div class="page-hdr">MySNS</div>
  <div class="page-body">
    <div class="section pad-24 mtop-30">
      <input name="name" type="text" placeholder="이름" required>
      <input name="phonenum" type="text" class="mtop-10" placeholder="휴대폰 번호" required>
      <input type="submit" class="mtop-20" value="아이디 찾기">
    </div>
    <div class="section pad-24 mtop-30">
      <div class="desc mtop-20 mbot-20">
        비밀번호도 잊으셨나요? 아래의 버튼을 통해 찾아보세요!
      </div>
      <a href="/findPW.html"><div class="button">비밀번호 찾기</div></a>
    </div>
  </div>
  <div class="page-footer">Copyright: mysns.com, 202x</div>

```

이름

휴대폰 번호

아이디 찾기

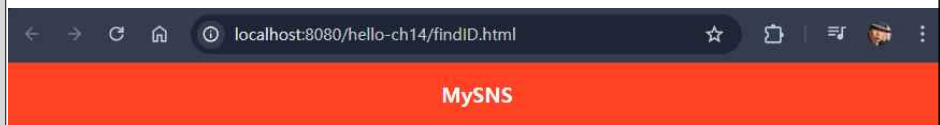
- findID.jsp = UserDao.java파일에 추가된 findID 클래스와 연관이 있다. 먼저 String name = request.getParameter("name"); 은 클라이언트가 html 폼을 통해 제출한 값 중 name(이름)에 입력된 파라미터 값을 가져오고, 그 값을 name이라는 java 변수에 저장함. 마찬가지로 phonenum(폰번호)에 입력된 파라미터 값을 가져오고 phonenum이라는 java변수에 저장. 그리고 UserDao dao = new UserDao(); 는 사용자 정보를 다루는 UserDao의 객체를 생성하여 DB 접근 메서드를 사용함. String userId = dao.findId(name, phonenum);를 통해 앞에서 입력받은 name과 phonenum을 인자로 넘겨 findId() 메서드를 호출한다.

```

<%@ page contentType="text/html" pageEncoding="utf-8" %>
<%@ page import="dao.UserDao" %>
<%
    request.setCharacterEncoding("utf-8");
    String name = request.getParameter("name");
    String phonenum = request.getParameter("phonenum");
    UserDao dao = new UserDao();
    String userId = dao.findId(name, phonenum);
    if(userId != null) {
        out.println("입력된 이름: " + name + "<br>");
        out.println("입력된 폰번호: " + phonenum + "<br>");
        out.print("당신의 아이디는: " + userId);
    } else {
        out.println("입력된 이름: " + name + "<br>");
        out.println("입력된 폰번호: " + phonenum + "<br>");
        out.print("일치하는 회원 정보가 없습니다.");
    }
%>

```

◆ 결과 화면



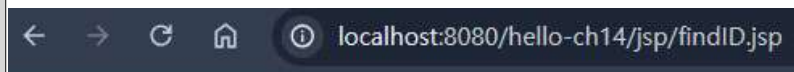
이은우

010-1234-5678

아이디 찾기

비밀번호도 잊으셨나요? 아래의 버튼을 통해 찾아보세요!

비밀번호 찾기



입력된 이름: 이은우
입력된 폰번호: 010-1234-5678
당신의 아이디는: lew@abc.com

◆findPW(비밀번호 찾기)

- **UserDAO.java** = checkUser와 updatePassword 클래스를 생성.
이번에는 findPW.html 에서 ID와 이름, 휴대폰 번호를 입력하여
잃어버린 PW를 찾는데 이때 입력 값이 findPW.jsp 파일에서 요청하여
응답할 때 이용된다. 위의 ID찾기와 동일하며 Mysns DB에서
user테이블에 접근하여 입력받은 이름, 휴대폰번호, 아이디를 검색하여
일치하는 유저가 있는지 확인하고, 유저가 존재하면 그 유저의 PW를
변경하기 위해 updatePassword클래스를 사용하여 비밀번호를 변경한다.
"SELECT * FROM user WHERE id=? AND name=? AND phonenum=?";
그리고 updatePassword에서는 PW를 변경할 유저의 id를 파악하고, 그
유저의 pw를 입력한 값으로 변경한다.
"UPDATE user SET password=? WHERE id=?";

- **findPW.html** = CSS 설정, 그 외 인코딩 파일 설정은 다른 html
파일과 동일하다. 눈여겨 볼 것은 사용자의 아이디와 이름, 휴대전화
번호를 입력하고 비밀번호 찾기 버튼이다. 그리고 밑 부분에 로그인 화면(
login.html)으로 이동하는 버튼을 추가하였다.


```

public boolean checkUser(String uid, String uname, String phonenum) throws NamingException, SQLException {
    String sql = "SELECT * FROM user WHERE id=? AND name=? AND phonenum=?";
    Connection conn = ConnectionPool.get();
    PreparedStatement stmt = null;
    try {
        stmt = conn.prepareStatement(sql);
        stmt.setString(1, uid);
        stmt.setString(2, uname);
        stmt.setString(3, phonenum);
        ResultSet rs = stmt.executeQuery();
        return rs.next();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
    }
    return false;
}

public boolean updatePassword(String id, String newPw) throws NamingException, SQLException {
    String sql = "UPDATE user SET password=? WHERE id=?";
    Connection conn = ConnectionPool.get();
    PreparedStatement stmt = null;
    try {
        stmt = conn.prepareStatement(sql);
        stmt.setString(1, newPw);
        stmt.setString(2, id);
        return stmt.executeUpdate() == 1;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}
}

<form action="jsp/findPW.jsp" method="post">
    <div class="page-hdr">MySNS</div>
    <div class="page-body">
        <div class="section pad-24 mtop-30">
            <input name="id" type="text" placeholder="아이디 (이메일주소)" required>
            <input name="name" type="text" class="mtop-10" placeholder="이름" required>
            <input name="phonenum" type="text" class="mtop-10" placeholder="휴대폰 번호" required>
            <input type="submit" class="mtop-20" value="비밀번호 찾기">
        </div>
        <div class="section pad-24 mtop-30">
            <div class="desc mtop-20 mbot-20">
                <a href="login.html"><div class="button">로그인하러가기</div></a>
            </div>
        </div>
    </div>
    <div class="page-footer">Copyright: mysns.com, 202x</div>
</form>

```

- findPW.jsp = UserDao.java파일에 추가된 checkuser 클래스와 연관이 있다. 먼저 `String id = request.getParameter("id");` 는 클라이언트가 html 폼을 통해 제출한 값 중 id에 입력된 파라미터 값을 가져오고, 그 값을 id이라는 java 변수에 저장함. 마찬가지로 `name`(이름), `phonenum`(폰번호)에 입력된 파라미터 값을 가져오고 `name`, `phonenum`이라는 java변수에 저장. 그리고 `UserDAO dao = new UserDao();` 는 사용자 정보를 다루는 UserDao의 객체를 생성하여 DB 접근 메서드를 사용함. [위의 findID.jsp와 같은 방식]

boolean valid = dao.checkUser(id, name, phonenum);를 통해 앞에서 입력받은 id, name, phonenum을 인자로 넘겨 cheackUser() 메서드를 호출한다.

```
<%@ page contentType="text/html" pageEncoding="utf-8" %>
<%@ page import="dao.UserDAO" %>
<%
    request.setCharacterEncoding("utf-8");
    String id = request.getParameter("id");
    String name = request.getParameter("name");
    String phonenum = request.getParameter("phonenum");
    UserDAO dao = new UserDAO();
    boolean valid = dao.checkUser(id, name, phonenum);
    if(valid) {
%>
<form action="resetPW.jsp" method="post">
    새 비밀번호: <input type="password" name="newPw"><br>
    <input type="hidden" name="id" value="<%= id %>">
    <button type="submit">비밀번호 변경</button>
</form>
<%
    } else {
        out.println("입력된 이메일: " + phonenum + "<br>");
        out.println("입력된 이름: " + name + "<br>");
        out.println("입력된 전화번호: " + phonenum + "<br>");
        out.print("입력하는 회원 정보가 없습니다.");
    }
%>
```

- resetPW.jsp = findPW.jsp를 통해 PW를 변경할 유저 id값을 가져오고, 그 유저의 pw를 새로 변경하기 위해 resetPW.jsp를 불러온다. 파라미터를 불러오고 저장하는 내용은 findPW.jsp와 동일하며, **boolean** result = dao.updatePassword(id, newPw);를 통해 id와 newPW값을 인자로 넘겨받아 updatParameter() 메서드를 호출한다. 성공시 입력된 id, newPW 값을 보여주며 변경 성공, 실패 여부를 보여준다.

```
<%@ page contentType="text/html" pageEncoding="utf-8" %>
<%@ page import="dao.UserDAO" %>
<%
    String id = request.getParameter("id").trim();
    String newPw = request.getParameter("newPw");
    UserDAO dao = new UserDAO();
    boolean result = dao.updatePassword(id, newPw);
    if(result) {
        out.println("입력된 id: " + id + "<br>");
        out.println("입력된 newPw: " + newPw + "<br>");
        out.print("비밀번호가 성공적으로 변경되었습니다.");
    } else {
        out.println("입력된 id: " + id + "<br>");
        out.println("입력된 newPw: " + newPw + "<br>");
        out.print("비밀번호 변경에 실패했습니다.");
    }
%>
```

- 비밀번호 변경시 화면

	<div><div>localhost8080/hello-ch14/jsp/</div><div>localhost8080/hello-ch14/jsp/findPW.jsp</div><div>새 비밀번호:</div><div>비밀번호 변경</div><div>- 성공시 화면</div><div><div>localhost8080/hello-ch14/jsp/resetPW.jsp</div><div>입력된 id: lew@abc.com 입력된 newPw: 12345 비밀번호가 성공적으로 변경되었습니다.</div><div>- 실패시 화면</div><div><div>localhost8080/hello-ch14/jsp/resetPW.jsp</div><div>비밀번호 변경에 실패했습니다.</div></div></div></div>
기대 효과 및 활용방안	<p>기계학습서비스 구현에 필요한 서버구축기술을 익힘으로써 서버 프런트엔트, 백엔드 분야로 진출이 가능하고, 서버 프로그래밍 기술을 기반으로 인공지능의 서비스를 웹을 통해서 제공할 수 있는 서버기술을 제공할 수 있게 된다.</p>
2025 년 6 월 20 일	
지도교수 심 효 선 (인)	