

O B J E C T

1. 시스템 개요 (기능 정의)

시스템 개요 및 기능을 1페이지 이내로 간략히 설명함.

2. 시스템 구현 (프로그램 소스)

프로그램 소스 및 커멘트

3. PC 웹 실행 결과 (결과 화면)

PC 웹 실행 결과화면을 캡처하고, 이에 대해 간략히 설명함. 프로그램의 완성도를 입증해 보임.

4. 모바일 웹 실행 결과 (결과 화면)

모바일 웹 실행 결과화면을 캡처하고, 이에 대해 간략히 설명함. 프로그램의 완성도를 입증해 보임.

5. 결 론

본 과제를 통하여 알게 된 내용 및 핵심을 자유롭게 정리함.

과제 #2 안내서

| 과제번호 | | 웹 프로그래밍 과제 #2 | | | | < 2인 1조 과제 > | |
|------|------|---|------|------|-----------------|--------------|--|
| 과제제목 | 유형1 | DB 연동 명함관리시스템의 명함입력 및 조회 프로그램 작성 | | | | | |
| | 유형2 | DB 연동 명함관리시스템의 명함입력 및 조회 프로그램 작성 (+ 모바일 웹 기능 포함) | | | | | |
| 과제일시 | 01분반 | 2022. 11. 29 (화) | 제출기한 | 01분반 | 2022. 12. 14(수) | | |
| | 02분반 | | | 02분반 | | | |
| 담당교수 | | 최 재원 교수 | | | | | |

과제개요

1. 개요

자신의 과제1 확장하여 JDBC를 이용한 DB 연동 명함관리시스템의 명함 입력 및 조회 JSP 프로그램을 구현한다. PC와 모바일 폰에서 웹서비스가 지원되도록 한다.

- 사용자 입력은 과제1을 이용하고, 명함관리 memberTBL 테이블의 스키마는 본인의 명함관리에서의 활용을 고려하여 정의할 것.
- 한글 입출력이 가능하도록 함. 명함 DB에는 과제수행자의 명함을 반드시 입력함.
- 자신의 과제1 프로그램을 기초하여 프로그래밍하고, 프로그램에는 주석을 달 것.
- 모바일 웹 기능구현: 화면크기, 전화, 메일, 맵 서비스 등 (유형2 가점) ※수업중 설명
- 참고: 교과서 11장의 JDBC 회원관리 프로그램을 활용하여 구현함.

2. 형식 및 내용

- 1) 과제 제목 2) 소스 프로그램 및 주석
- 3) 실행 결과: 실행결과 화면 캡처 4) 핵심 정리: 중요핵심, 알게 된 내용정리

3. 레포트 제출

- 1) 레포트는 반드시 출력하여 12/14(목)까지 1부만 레포트함 제출. (미제출 20% 감점)
- 2) 레포트와 소스코드는 홍길동 장보고.zip 압축하여 1부만 구글-레포트 실에 업로드함.

4. 채점 기준 (유의: 프로그램 복제 시 둘 다 혹은 뒤에 제출된 레포트를 복제로 봄)

프로그램과 주석 (50%), 실행결과화면 (20%), 모바일 웹 (25%), 핵심정리 (5%)

- 주1) 실행결과 화면은 프로그램의 정상 동작여부를 입증할 수 있어야 함.
주2) 참여도 표시: 결과보고서의 표지에 과제 참여도를 표시함.

매우우수 (AA) 우수 (A) 보통 (B) 부족 (C) 무성의/거의불참 (D) 불참 (F)

1. 시스템 개요 (기능 정의)

명함 DB연동 JSP / JAVA / DataBase

▶ DataBase

ch11 DB sql 명령(MySql Workbench).sql

JDBC 프로그래밍 : JDBC 드라이브 로드 -> 데이터베이스 연결 -> Statement 생성

-> SQL 문 전송 -> 결과 받기

▶ JSP

InsertTestForm_01.jsp : 명함 등록 화면

InsertTestPro_01.jsp : 테이블에 레코드를 삽입(추가)하는 JSP, 명함 등록 완료

UpdateTestForm_01.jsp : 명함 수정 화면

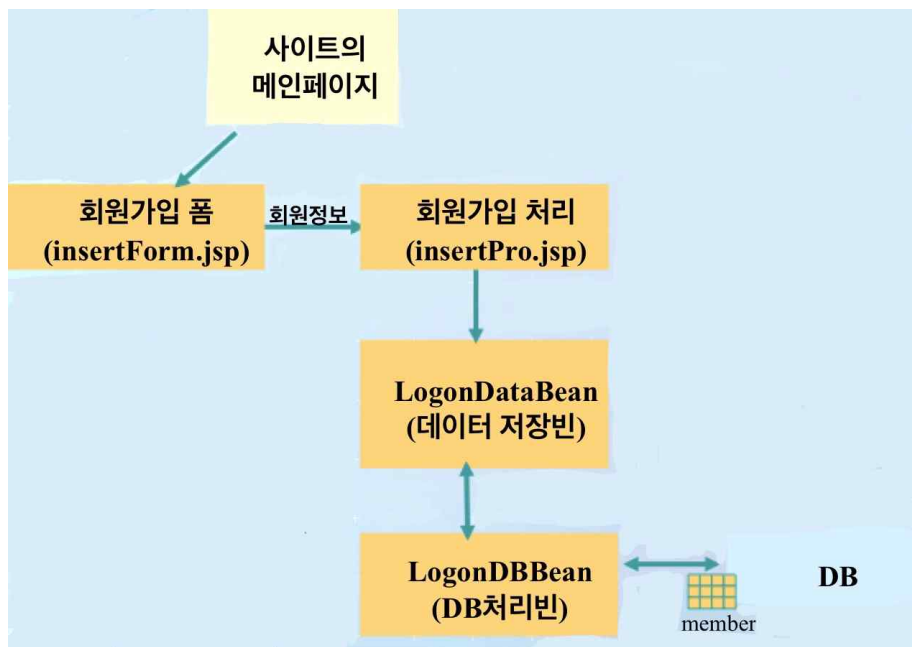
UpdateTestPro_01.jsp : 테이블에 저장되어 있는 레코드를 수정하는 JSP, 명함 수정 완료

DeleteTestForm_01.jsp : 명함 삭제 화면

DeleteTestPro_01.jsp : 테이블에 저장되어 있는 레코드를 삭제하는 JSP, 명함 삭제 완료

SelectTestPro_01.jsp : 테이블에 저장되어 있는 레코드를 검색(조회)하는 JSP, 테이블 목록

usePool_01.jsp : 커넥션풀로부터 커넥션 사용하여 테이블에 저장되어 있는 레코드를 조회하는 JSP



▶ Java

RegisterBean.java : 명함등록을 전달할 자바빈

-과제 1에서 제작한 명함 입력폼을 이용하여 입력된 명함을 관리하는 DB를 생성합니다.

▶ 모바일 웹 기능 구현

inkTestForm.jsp : 모바일 웹 명함 등록 화면

linkTestProc.jsp : 모바일 웹 명함 처리 JSP

2. 시스템 구현 (프로그램 소스)

ch11 DB sql 명령(Mysql Workbench).sql

기존 실습중에 생성한 basicjsp 테이블을 먼저 DROP문으로 삭제 해줍니다.

`/* DB 및 계정 삭제 */`

`DROP DATABASE IF EXISTS basicjsp;`

`DROP USER IF EXISTS jspid@'%';`

`-- DROP USER IF EXISTS jspid@localhost;`

데이터 베이스의 계정을 새로 생성하기 위해 사용자의 ID와 pass를 create로 생성한 후 basicjsp테이블도 생성합니다.

show databases를 입력하여 basicjsp테이블이 데이터 베이스에 생성이 되었는지 확인합니다.

`/* 계정 생성, DB 생성 및 접근권한 부여 */`

`create user jspid@'%' identified with mysql_native_password by 'jsppass';`

`-- create user jspid@localhost identified with mysql_native_password by 'jsppass';`

`create database basicjsp;`

`grant all privileges on basicjsp.* to jspid@'%' with grant option;`

`show databases;`

개별 권한부여

`-- grant select, insert, update, delete, create, drop, alter`

`-- on basicjsp.* to jspid@'%'`

`-- with grant option;`

basicjsp 계정으로 use를 이용하여 접속합니다.

-CMD창에서 데이터 베이스를 접속하는 방법

1. `mysql -u root -p`를 입력합니다.

2. 지정한 password를 입력합니다.

3. `show databases;`를 입력하여 데이터베이스에 접속합니다.

4. `use basicjsp;`를 입력하여 basicjsp테이블로 접속합니다.

`use basicjsp;`

테이블의 속성을 입력하기 위해 우선 test테이블을 생성해봅니다.

membertbl의 테이블을 생성하기 위해 실습 중에 생성한 member테이블의 속성을 그대로 사용하여 생성합니다.

```
create table test (
    num_id int not null primary key auto_increment,
    title varchar(50) not null,
    content text not null
```

```
);
```

```
create table memberTBL (
    id varchar(12) not null primary key,
    passwd varchar(12) not null,
    name varchar(10) not null,
    reg_date datetime not null
```

```
);
```

생성된 테이블을 show tables;을 입력하여 확인하고 desc memberTBL;을 입력하여 membertbl의 항목을 확인합니다.

```
show tables;
desc memberTBL;
```

생성된 membertbl의 테이블에 항목이 입력되는지 insert문을 이용하여 확인합니다

```
/* Table 데이터 입력 */
insert into memberTBL values ('hongkd', 'happy', '홍길동', now());
insert into memberTBL values ('kimkd', 'happy', '김길동', now());
insert into memberTBL values ('choijw', 'happy', '최재원', now());
insert into member1 values ('hongkd', 'happy', '홍길동', '111111', '111111', 'hongkd@ks.ac.kr', '
홍길동의 블로그', now());
/* Table 데이터 조회, 수정, 삭제 */
select * from memberTBL;
select * from memberTBL where id='hongkd';
select * from memberTBL where name like '%길동%';
update memberTBL set id='choejw' where id='choijw';
delete from memberTBL where id='choejw';
/* MySQL 상에서 Auto Commit 해제, 설정, 확인 방법 */
set autocommit=0;           -- 해제: 0, 설정: 1
select @@autocommit;        -- 해제: 0, 설정: 1
begin;                       -- autocommit=1 이라도 rollback 가능
rollback;                    -- 롤백: rollback, 실행: commit
/*참고. JSP 상에서 Auto Commit 해제, 설정 및 트랜잭션 처리 */
/*
conn.setAutoCommit(false);   -- 해제: false, 설정: true
conn.commit();               -- 트랜잭션 실행
conn.rollback();              -- 트랜잭션 철회 */
```

SELECT * FROM basicjsp.membertbl;으로 membertbl의 테이블을 확인하고 insertTestForm에서 명함 입력 폼의 항목을 해당 테이블의 항목으로 입력하기 위해 alter add를 사용하여 부서/직책/이메일/전화번호/주소를 입력합니다.

이전에 테이블에 생성한 id와 passwd, date는 명함 입력폼에 필요가 없는 항목으로 alter drop문으로 삭제합니다

```
SELECT * FROM basicjsp.membertbl;
delete from basicjsp.membertbl;
desc basicjsp.membertbl;
alter table basicjsp.membertbl add department varchar(10) not null;
alter table basicjsp.membertbl add position varchar(20) not null;
alter table basicjsp.membertbl add email varchar(50) not null;
alter table basicjsp.membertbl add tell varchar(20) not null;
alter table basicjsp.membertbl add member_addr varchar(80) not null;
alter table basicjsp.membertbl add detail_addr varchar(80) not null;
alter table basicjsp.membertbl drop id ;
alter table basicjsp.membertbl drop passwd ;
alter table basicjsp.membertbl drop reg_date ;
```

명함입력

InsertTestForm_01.jsp

```
<!--JSP 페이지에 대한 정보를 page 디렉티브의 속성들을 사용하여 정의하였습니다. --%>
<!--JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. --%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
```

```
<html>
<head>
<title>레코드 삭제</title>
```

과제 1에서 제작한 명함 입력 폼의 디자인을 이용하여 inserttestform을 제작합니다.

```
<!-- style 태그를 적용하여 배경 전체 이미지, 테이블과 h2 태그에 디자인을 적용하였습니다.
-->
<style>
```

```
body{margin : 0 auto;
width:100%;} /* 기본적으로 윈도우에 지정되어 있는 여백을 없애준후 가로길이 설정 */
#wrap{
width:90%;
height:90%;
margin : 40px auto;
text-align:center;
background-color:#ffffff;
}/*wrap이라는 id객체를 지정한 상자 스타일
가로 세로 전체 윈도우 크기의 90%
바깥쪽 여백 위아래 40px 지정 좌우 중앙정렬
텍스트 중앙정렬 배경색 흰색
*/
```

```
table{
margin-left:auto;
margin-right:auto;
width:40%;
height:30%;
}/* 표 만들기(명함 등록 폼을 표로 생성)
바깥쪽여백 좌우 auto로 중앙정렬
가로 세로 크기 = 800X500
*/
```

```
tr,td{
border-bottom:4px solid #C4B9F5;
padding-left:10px;
}
/*생성 한 표 내 행과 열 스타일
아래 테두리만 4px 실선으로 지정 색상 지정
안쪽 여백 10px
*/
```

```
input{
width: 300px;
height: 40px;
font-size: 15px;
border: 0;
border-radius: 25px;
outline: none;
padding-left: 10px;
background-color: rgb(233, 233, 233);
}/* input태그 스타일
가로 세로 크기 300X400
테두리 없음(0으로 지정)/모서리 둥글게(25px)
안쪽여백 10px 배경 색지정
*/
input:focus{
background-color: #E2DBFF;
}
/* input태그 상자를 클릭하는 경우(선택자뒤에 ':focuss'를 입력)
배경색 지정
*/
```


InsertTestPro_01.jsp

명함 입력을 한 후 데이터 베이스에 저장한 후 selectTest파일에 저장된 명함들을 모아보기 위함입니다.

내장객체를 이용하여 지정된 name값을 getParameter()로 읽어 옵니다.

데이터베이스를 연결하기 위해 connection을 이용하고 전송하기 위해 preparedStatement를 사용합니다.

```
<%-- JSP 페이지에 대한 정보를 page 디렉티브의 속성들을 사용 --%>
```

```
<%-- JSP 페이지가 생성할 문서의 타입 지정--%>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
```

```
<%@ page import="java.sql.*"%>
```

```
<%-- 웹 브라우저에서 서버로 넘어오는 파라미터 값에 한글이 있는 경우, 한글깨짐 방지(post
방식)--%>
```

```
<%request.setCharacterEncoding("utf-8");%>
```

```
<%-- request 내장객체의 getParameter() 메소드를 사용 프로퍼티 값 읽기 --%>
```

```
<%
```

```
String name = request.getParameter("name");
```

```
String email = request.getParameter("email");
```

```
String tell = request.getParameter("tell");
```

```
String department = request.getParameter("department");
```

```
String position = request.getParameter("position");
```

```
String member_addr = request.getParameter("member_addr");
```

```
String detail_addr = request.getParameter("detail_addr");
```

```
Connection conn = null;//connection은 데이터베이스와 연결하는 객체입니다
```

```
PreparedStatement pstmt = null;// sql문 전송하는 객체인 PreparedStatement
```

```
String str = "";//문자열을 공백으로 출력
```

데이터베이스에 폼에 입력한 정보를 저장하기 위해 각각의 메소드의 순서를 정해줍니다.

연동할 데이터베이스의 url을 입력하고 사용자의 계정과 패스워드를 입력해줍니다.

(계정의 패스워드를 입력하는 이유는 연결된 DB를 생성했을 때 먼저 계정을 생성했습니다.)

드라이버 매니저를 등록한 후 위에서 연결한 객체의 id,pass,url을 가져오고 membertbl 테이블에서 입력하기 위해 프로퍼티 값을 ?로 개수를 표시합니다.

(데이터베이스를 입력하기 위해 해당 함수들은 필수적을 들어가야 합니다.)

```
/*예외 처리 부분입니다.*/
```

```
try {
```

```
/*연동할 DB를 포함한 url*/
```

```
String jdbcUrl = "jdbc:mysql://localhost:3306/basicjsp?serverTimezone=Asia/Seoul";
```

```
String dbId = "jspid";//사용자 계정
```

```
String dbPass="jsppass";//사용자 패스워드
```

```
/*DriverManager 등록*/
```

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

```
/*Connection 객체의 url,id,pass를 얻어옴.*/
```

```
conn = DriverManager.getConnection(jdbcUrl, dbId, dbPass);
```

```
/*sql은 membertbl테이블의 values (?,?,?,?,?)을 레코드 추가
```

```
?에 해당되는 필드의 데이터 타입은 문자열이며 읽어올 프로퍼티 값의 갯수대로 넣기*/
```

```
String sql = "insert into memberTBL values (?,?,?,?,?)";
```

```
/*sql이 실행할 쿼리문 */
```

```
pstmt = conn.prepareStatement(sql);
```


제동된 string 메소드를 순차적으로 입력한 후 올바르게 입력이 되었으면 str문으로 등록이 완료 되었다는 문구가 생성됩니다

```
/*PreparedStatement가 제공하는 setString 메소드
순차적으로 제공*/
pstmt.setString(1, name);
pstmt.setString(2, department);
pstmt.setString(3, position);
pstmt.setString(4, email);
pstmt.setString(5, tell);
pstmt.setString(6, member_addr);
pstmt.setString(7, detail_addr);

pstmt.executeUpdate();

str = "회원 등록이 완료되었습니다 :";
}
```

잘못 입력하면 str문으로 등록이 실패되었다는 문구가 생성됩니다

```
catch (Exception e) { //Exception e가 발생할 경우 작성하는 문구
    e.printStackTrace();
    str = "회원 등록에 실패했습니다";
}
finally { //try 문에서 더이상의 실행이 필요 없을 경우 마지막으로 입력하는 문구
    if (pstmt != null)
        try {
            pstmt.close();
        } catch (SQLException sqle) {}
    if (conn != null)
        try {
            conn.close();
        } catch (SQLException sqle) {}
}
%>
<html>
<head>
<title>레코드 추가</title>

</head>

<body>
<div style=" text-align: center; line-height: 400px; border: 50px solid #C4B9F5; width:
50%; height: 40%; margin: 10% auto; font-size: 40px; font-weight: bold; color: #919191;">
    <%=str %><!-- 상단에 지정한 실패/성공 문구를 출력함 -->
</div>
</body>
</html>
```

명함수정

UpdateTestForm_01.jsp

입력된 명함정보의 부서를 수정하기 위해 이용합니다.

```
<!--JSP 페이지에 대한 정보를 page 디렉티브의 속성들을 사용하여 정의하였습니다. -->
<!--JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. -->
<@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

<html>
<head>
<title>레코드 수정</title>

<!-- style 태그에 사용된 디자인이 적용됩니다. -->
<style>

    body{margin : 0 auto;
width:100%;} /* 기본적으로 윈도우에 지정되어 있는 여백을 없애준후 가로길이 설정 */
    #wrap{
        width:90%;
        height:90%;
        margin : 40px auto;
        text-align:center;
        background-color:#ffffff;
    }/*wrap이라는 id객체를 지정한 상자 스타일
    가로 세로 전체 윈도우 크기의 90%
    바깥쪽 여백 위아래 40px 지정 좌우 중앙정렬
    텍스트 중앙정렬 배경색 흰색
    */
    table{
        margin-left:auto;
        margin-right:auto;
        width:50%;
        height:40%;
    }/* 표 만들기(명함 등록 폼을 표로 생성)
    바깥쪽여백 좌우 auto로 중앙정렬
    가로 세로 크기 = 800X500
    */

    tr,td{
        border-bottom:4px solid #C4B9F5;
        padding-left:10px
    }
    /*생성 한 표 내 행과 열 스타일
    아래 테두리만 4px 실선으로 지정 색상 지정
    안쪽 여백 10px
    */

    input{
        width: 300px;
        height: 40px;
        font-size: 15px;
        border: 0;
        border-radius: 25px;
        outline: none;
        padding-left: 10px;
        background-color: rgb(233, 233, 233);
    }/* input태그 스타일
    가로 세로 크기 300X400
    테두리 없음(0으로 지정)/모서리 둥글게(25px)
    안쪽여백 10px 배경 색지정
    */
    input:focus{
        background-color: #E2DBFF;
    }
    /* input태그 상자를 클릭하는 경우(선택자뒤에 ':focuss'를 입력)
    배경색 지정
    */
```

[illegible]

UpdateTestPro_01.jsp

getParameter 메소드를 이용하여 프로퍼티 값을 읽어온 후 데이터 베이스를 연동합니다.
insertTestPro 파일과 동일하게 데이터 베이스를 연결합니다.

```
<%--JSP 페이지에 대한 정보를 page 디렉티브의 속성들을 사용하여 정의하였습니다. --%>
<%--JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. --%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.sql.*"%>
```

```
<%-- 웹 브라우저에서 서버로 넘어오는 파라미터 값에 한글이 있는 경우, 한글 깨짐을
방지합니다.(post 방식)--%>
<% request.setCharacterEncoding("utf-8"); %>
```

```
<%-- request 내장객체의 getParameter() 메소드를 사용하여 프로퍼티 값을 읽음 --%>
<%
String name= request.getParameter("name");
String tell= request.getParameter("tell");
String position= request.getParameter("position");
```

```
Connection conn=null;//connection은 데이터베이스와 연결하는 객체입니다
PreparedStatement pstmt=null;// sql문 전송하는 객체인 PreparedStatement
ResultSet rs=null;;
```

```
/*예외 처리*/
try{
```

```
    /*연동할 DB를 포함한 url*/
    String
    jdbcUrl="jdbc:mysql://localhost:3306/basicjsp?serverTimezone=Asia/Seoul";
```

```
    String dbId="jspid";//사용자 계정
    String dbPass="jsppass";//사용자 패스워드
```

```
    /*DriverManager에 등록*/
    Class.forName("com.mysql.cj.jdbc.Driver");
    /*Connection 객체의 url,id,pass를 얻어옴*/
    conn=DriverManager.getConnection(jdbcUrl,dbId ,dbPass );
    /*sql은 membertbl테이블의 values (?,?,?,?,?)을 레코드 추가
    ?에 해당되는 필드의 데이터 타입은 문자열이며 읽어올 프로퍼티 값의 갯수대로 넣기*/
    String sql= "select name, tell from memberTBL where name= ?";
    //membertbl의 테이블의 name과 tell의 열이름의 name=?조건을 찾음
    /*sql이 실행할 쿼리문입니다.*/
    pstmt=conn.prepareStatement(sql);//Sql의 Statement 클래스의 기능 향상

    pstmt.setString(1,name);//PreparedStatement 객체가 제공하는 setString 메소드
    rs=pstmt.executeQuery();//객체의 값을 반환 select 구문을 수행할 때 사용하는 함수
```

if문으로 getString메소드를 이용하여 name과 tell을 읽을 수 있게 새로운 변수를 입력합니다

```
if(rs.next()){
    /*PreparedStatement 객체가 제공하는 getString 메소드*/
    String rName=rs.getString("name");
    String rTell=rs.getString("tell");
```

정의된 문자열 rName 과 변수tell의 위에서 정의한 문자열 rTell을 이용하여 membertbl테이블에서 부서와 이름을 수정합니다. 수정된 부서를 sql문으로 입력합니다

```
//변수.equals("상수")는 변수가 null이 들어올 경우 예외를 발생시키게 되고 비정상적인
프로그램 종료
```

```
//변수name의 위에서 정의한 문자열 rName 과 변수tell의 위에서 정의한
문자열 rTell을 선언(&&:and)
```

```
if(name.equals(rName) && tell.equals(rTell)){
    //sql문에서 membertbl테이블의 department= ?을 수정
```

```

        sql= "update memberTBL set department= ? where name= ? ";
        pstmt=conn.prepareStatement(sql);/*sql이 실행할 쿼리문 */
        pstmt.setString(1,position); //PreparedStatement 객체가 제공하는 setString 메소드
        pstmt.setString(2,name); //PreparedStatement 객체가 제공하는 setString 메소드
        pstmt.executeUpdate(); //객체의 값을 반환 select 구문을 수행할 때 사용하는 함수;
    %>
<html>
<head>
<title>레코드 수정</title>
<!-- style 태그에서 배경이미지와 표의 투명도를 설정하였습니다. -->
<style>

</style>
</head>
<body><!-- 회원정보 수정을 알리는 문구를 넣습니다 -->
    <p style="border: 50px solid #C4B9F5;width: 50%;height: 40%;margin: 10%
auto;font-size: 40px;font-weight: bold;color: #919191; line-height: 10; text-align:center;">
    회원 정보를 수정했습니다.</p>
    수정을 성공하면 수정 완료 문구가 뜨고 name과tell의 속성이 다르면 재입력문구가 생성됩니다.
    <%
        }
    else {if(name.equals(rName) && tell.equals(rTell))문의 쿼리 문을 삭제하기 못하면
    실행시킵니다
        //전화번호가 일치하지 않는 경우 재 입력을 요구
    %>
        <table style="border: 50px solid #C4B9F5;width: 50%;height: 40%;margin: 10%
auto;font-size: 40px;font-weight: bold;color: #919191;">
            <tr align="center" bgcolor="white">

                <td>전화번호를 다시 입력해주시길 바랍니다.</td>

            </tr>

        </table>
    <%
        }
    else {if(name.equals(rName) && tell.equals(rTell))문의 쿼리 문을 삭제하기 못하면
    실행시킵니다
        //전화번호가 일치하지 않는 경우 재 입력을 요구
    %>
        <table style="border: 50px solid #C4B9F5;width: 50%;height: 40%;margin: 10%
auto;font-size: 40px;font-weight: bold;color: #919191;">
            <tr align="center" bgcolor="white">
                <td>이름을 다시 입력해주시길 바랍니다.</td>
            </tr>

        </table>
    <%
        }
    }catch(Exception e){//Exception e가 발생할 경우 작성하는 문구
        e.printStackTrace();}
    finally{//try 문에서 더이상의 실행이 필요 없을 경우 마지막으로 입력하는 문구
        if(rs != null)
            try{rs.close();}catch(SQLException sqle){}
        if(pstmt != null)
            try{pstmt.close();}catch(SQLException sqle){}
        if(conn != null)
            try{conn.close();}catch(SQLException sqle){}
    }
    %>
</body>
</html>

```

명 함 삭 제

DeleteTestForm_01.jsp

입력된 명함의 정보를 삭제하기 위해 사용합니다.

```
<!--JSP 페이지에 대한 정보를 page 디렉티브의 속성들을 사용하여 정의하였습니다. -->
<!--JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

<html>
<head>
<title>레코드 삭제</title>

<!-- style 태그를 적용하여 배경 전체 이미지, 테이블과 h2 태그에 디자인을 적용하였습니다.
-->
<style>

    body{margin : 0 auto;
        width:100%;} /* 기본적으로 윈도우에 지정되어 있는 여백을 없애준후 가로길이 설정 */
    #wrap{
        width:90%;
        height:90%;
        margin : 40px auto;
        text-align:center;
        background-color:#ffffff;
    }/*wrap이라는 id객체를 지정한 상자 스타일
    가로 세로 전체 윈도우 크기의 90%
    바깥쪽 여백 위아래 40px 지정 좌우 중앙정렬
    텍스트 중앙정렬 배경색 흰색
    */
    table{
        margin-left:auto;
        margin-right:auto;
        width:50%;
        height:30%;
    }/* 표 만들기(명함 등록 폼을 표로 생성)
    바깥쪽여백 좌우 auto로 중앙정렬
    가로 세로 크기 = 800X500
    */

    tr,td{
        border-bottom:4px solid #C4B9F5;
        padding-left:10px
    }

    /*생성 한 표 내 행과 열 스타일
    아래 테두리만 4px 실선으로 지정 색상 지정
    안쪽 여백 10px
    */
    input{
        width: 300px;
        height: 40px;
        font-size: 15px;
        border: 0;
        border-radius: 25px;
        outline: none;
        padding-left: 10px;
        background-color: rgb(233, 233, 233);
    }/* input태그 스타일
    가로 세로 크기 300X400
    테두리 없음(0으로 지정)/모서리 둥글게(25px)
    안쪽여백 10px 배경 색지정
    */
    input:focus{
        background-color: #E2DBFF;
    }
    /* input태그 상자를 클릭하는 경우(선택자뒤에 ':foucus'를 입력)
    배경색 지정
    */
```


DeleteTestPro_01.jsp

```
<!-- JSP 페이지에 대한 정보를 page 디렉티브의 속성들을 사용하여 정의하였습니다. -->
<!-- JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.sql.*"%>
```

```
<!-- 웹 브라우저에서 서버로 넘어오는 파라미터 값에 한글이 있는 경우, 한글 깨짐을
방지합니다.(post 방식)-->
<%request.setCharacterEncoding("utf-8");%>
```

```
<%
//request.getParameter(" "); HTTP요청의 파라미터 값을 얻기 위해 사용함
String name= request.getParameter("name");
String tell= request.getParameter("tell");
```

```
Connection conn=null;//connection은 데이터베이스와 연결하는 객체입니다
PreparedStatement pstmt=null;// sql문 전송하는 객체인 PreparedStatement
ResultSet rs=null;
```

```
/* 예외 처리 */
```

```
try{
    /*연동할 DB를 포함한 url*/
    String jdbcUrl="jdbc:mysql://localhost:3306/basicjsp?serverTimezone=Asia/Seoul";
```

```
String dbId="jspid";//사용자 계정
String dbPass="jsppass";//사용자 패스워드
```

```
/*DriverManager에 등록 */
Class.forName("com.mysql.cj.jdbc.Driver");
```

```
/*Connection 객체의 url,id,pass를 얻어옴.*/
conn=DriverManager.getConnection(jdbcUrl,dbId ,dbPass );
```

```
String sql= "select name, tell from memberTBL where name= ?";//membertbl의
테이블의 name과 tell의 열이름의 name=?조건을 찾음
pstmt=conn.prepareStatement(sql);//Sql의 Statement 클래스의 기능 향상
```

if문으로 getString메소드를 이용하여 name과 tell을 읽을 수 있게 새로운 변수를 입력합니다

```
pstmt.setString(1,name);//PreparedStatement 객체가 제공하는 setString 메소드
rs=pstmt.executeQuery();//객체의 값을 반환 select 구문을 수행할 때 사용하는 함수
```

```
if(rs.next()){
    /*PreparedStatement 객체가 제공하는 getString 메소드*/
    String rName=rs.getString("name");
    String rTell=rs.getString("tell");
```

```
//변수.equals("상수")는 변수가 null이 들어올 경우 예외를 발생시키게 되고 비정상적인
프로그램 종료
//변수name의 위에서 정의한 문자열 rName 과 변수tell의 위에서 정의한 문자열 rTell을
선언(&&and)
```

정의된 문자열 rName 과 변수tell의 위에서 정의한 문자열 rTell을 이용하여
membertbl테이블에서 부서와 이름을 수정합니다. 수정된 부서를 sql문으로 입력합니다

```
if(name.equals(rName) && tell.equals(rTell)){
    //sql문에서 membertbl테이블의 name= ?을 삭제
    sql= "delete from memberTBL where name= ? ";
    pstmt=conn.prepareStatement(sql); /*sql이 실행할 쿼리문 */
    pstmt.setString(1,name);//PreparedStatement 객체가 제공하는 setString 메소드
    pstmt.executeUpdate();//객체의 값을 반환 select 구문을 수행할 때 사용하는 함수
```

```
%>
```



```
<html>
<head>
<title>레코드 삭제</title>
```

```
</head>
<body>
```

```
<!-- 회원 탈퇴를 알리는 문구를 넣습니다 -->
```

```
<p style="border: 50px solid #C4B9F5;width: 50%;height: 40%;margin: 10% auto;font-size: 40px;font-weight: bold;color: #919191; line-height: 10; text-align:center;">
회원 탈퇴가 완료되었습니다.</p>
```

이름과 전화번호를 올바르게 입력하면 탈퇴완료 문구가 생성됩니다.

전화번호가 일치하지않는 경우 전화번호 재입력 문구가 생성되고

이름이 일치하지 않는 경우 이름을 재입력 문구가 생성됩니다.

```
<%
```

```
    }
```

```
    else {if(name.equals(rName) && tell.equals(rTell))문의 쿼리 문을 삭제하기
못하면 실행시킵니다
```

```
        //전화번호가 일치하지 않는 경우 재 입력을 요구
```

```
%>
```

```
        <table style="border: 50px solid #C4B9F5;width: 50%;height: 40%;margin: 10% auto;font-size: 40px;font-weight: bold;color: #919191;">
        <tr align="center" bgcolor="white">
```

```
            <td>전화번호를 다시 입력해주시길 바랍니다.</td>
```

```
        </tr>
```

```
    </table>
```

```
<%
```

```
    }
```

```
    else {if(name.equals(rName) && tell.equals(rTell))문의 쿼리 문을 삭제하기 못하면
실행시킵니다
```

```
        //전화번호가 일치하지 않는 경우 재 입력을 요구
```

```
%>
```

```
        <table style="border: 50px solid #C4B9F5;width: 50%;height: 40%;margin: 10% auto;font-size: 40px;font-weight: bold;color: #919191;">
        <tr align="center" bgcolor="white">
```

```
            <td>이름을 다시 입력해주시길 바랍니다.</td>
```

```
        </tr>
```

```
    </table>
```

```
<%
```

```
    }
```

```
}
catch(Exception e){e.printStackTrace();}//Exception e가 발생할 경우 작성하는 문구
finally{//try 문에서 더이상의 실행이 필요 없을 경우 마지막으로 입력하는 문구
```

```
    if(rs != null)
```

```
        try{rs.close();}catch(SQLException sqle){}
```

```
    if(pstmt != null)
```

```
        try{pstmt.close();}catch(SQLException sqle){}
```

```
    if(conn != null)
```

```
        try{conn.close();}catch(SQLException sqle){}
```

```
}
```

```
%>
```

```
</body>
```

```
</html>
```

usePool_01.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!-- JSP 페이지에 대한 정보를 page 디렉티브의 속성들을 사용하여 정의하였습니다. --%>
<!-- JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. --%>
<%@ page import = "java.sql.*, javax.sql.*, javax.naming.*" %>

<html>
<head>
<title>커넥션 풀_등록결과</title>
<!-- 생성된 표 스타일 생성 -->
<style>
body{margin : 0 auto;
width:100%;} /* 기본적으로 윈도우에 지정되어 있는 여백을 없애준후 가로길이 설정 */
#wrap{
width:90%;
height:90%;
margin : 40px auto;
text-align:center;
background-color:#ffffff;
}/*wrap이라는 id객체를 지정한 상자 스타일
가로 세로 전체 윈도우 크기의 90%
바깥쪽 여백 위아래 40px 지정 좌우 중앙정렬
텍스트 중앙정렬 배경색 흰색
*/
table{
margin-left:auto;
margin-right:auto;
width:70%;
}/* 표 만들기(명함 등록 폼을 표로 생성)
바깥쪽여백 좌우 auto로 중앙정렬
가로 세로 크기 = 800X500
*/

tr,td{
border-bottom:4px solid #C4B9F5;
padding-left:10px
}
/*생성 한 표 내 행과 열 스타일
아래 테두리만 4px 실선으로 지정 색상 지정
안쪽 여백 10px
*/
</style>
</head>


등록된 폼의 어떤 정보를 입력하는지 표로 타이틀을 입력합니다.
<body bgcolor="#C4B9F5"><!-- 배경 색상 -->
<DIV id="wrap"><!-- 전체 상자를 wrap을 id선택자로 입력. -->
<h2 style="padding:40px;"><font size="9" color="gray">등 록 결 과</font></h2>
<!-- h2태그를 이용하여 제목을 입력. 안쪽 여백을 40px을 주고 font태그로 글자 크기를 9
회색으로 지정. -->
<TABLE border="0" align="center">
<!-- 표만들기 테두리를 0으로 지정
입력된 명함을 한번에 보기위한 표를 생성합니다-->
<TR><!-- 행 정렬 -->
<TH width="200">이름</TH>
<TH width="200">부서</TH>
<TH width="200">직책</TH>
<TH width="500">이메일(E-mail)</TH>
<TH width="500">전화번호</TH>
<TH width="900">주소</TH>
<TH width="900">상세주소</TH>

</TR>
</DIV>


```

등록된 명함 폼을 확인하기 위해 저장된 데이터 베이스를 가져와야 합니다.

데이터 베이스를 가져오기 전 connnetion으로 연결하고 sql문을 전송하기 위해 preparedStatement 함수를 사용합니다.

basicjsp 테이블을 연동 시키고 membertbl의 전체 조건을 찾습니다.

제공되는 getString메소드를 지정하고 표입력창에 지정한 메소드의 값을 읽어 오면 등록한 회원 정보를 볼 수 있습니다.

<%

```
Connection conn=null;//connection은 데이터베이스와 연결하는 객체입니다
PreparedStatement pstmt=null;// sql문 전송하는 객체인 PreparedStatement
ResultSet rs=null;
```

/* 예외 처리 부분.*/

```
try{
    Context initCtx = new InitialContext();
    Context envCtx = (Context) initCtx.lookup("java:comp/env");
    DataSource ds = (DataSource)envCtx.lookup("jdbc/basicjsp");
    /*Connection 객체를 얻음 */
    conn = ds.getConnection();
```

//membertbl의 테이블의 전체 조건을 찾음

```
String sql= "select * from memberTBL";
//Sql의 Statement 클래스의 기능 향상
pstmt=conn.prepareStatement(sql);
//객체의 값을 반환 select 구문을 수행할 때 사용
rs=pstmt.executeQuery();
```

```
while(rs.next()){
    /*PreparedStatement 객체가 제공하는 getString 메소드*/
    String name= rs.getString("name");
    String department= rs.getString("department");
    String position= rs.getString("position");
    String email= rs.getString("email");
    String tell= rs.getString("tell");
    String member_addr= rs.getString("member_addr");
    String detail_addr= rs.getString("detail_addr");
```

%>

<!-- request 내장객체의 getParameter() 메소드를 사용하여 읽어온 id 프로퍼티 값을 표현식을 사용하여 화면에 출력. -->

```
<TR>
    <TD width="100"><%=name%></TD>
    <TD width="250"><%=department%></TD>
    <TD width="250"><%=position%></TD>
    <TD width="100"><%=email%></TD>
    <TD width="100"><%=tell%></TD>
    <TD width="250"><%=member_addr%></TD>
    <TD width="250"><%=detail_addr%></TD>
```

```
</TR>
```

<%

```
    }
}
```

catch(Exception e){out.println("Error! " + e.toString());} //Exception e가 발생할 경우 작성하는 문구

```
finally{//try 문에서 더이상의 실행이 필요 없을 경우 마지막으로 입력하는 문구
    if(rs != null) try{rs.close();}catch(SQLException sqle){}
    if(pstmt != null) try{pstmt.close();}catch(SQLException sqle){}
    if(conn != null) try{conn.close();}catch(SQLException sqle){}
}
```

%>

```
</table>
</body>
</html>
```

selectTest_01

```

<!--JSP 페이지에 대한 정보를 page 디렉티브의 속성들을 사용하여 정의하였습니다. -->
<!--JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.sql.*"%>

<html>
<head>
<title>레코드 표시</title>
<!-- 생성된 표 스타일 생성 -->
<style>
body{margin : 0 auto;
width:100%;} /* 기본적으로 윈도우에 지정되어 있는 여백을 없애준후 가로길이 설정 */
#wrap{
width:90%;
height:90%;
margin : 40px auto;
text-align:center;
background-color:#fffff;
}/*wrap이라는 id객체를 지정한 상자 스타일
가로 세로 전체 윈도우 크기의 90%
바깥쪽 여백 위아래 40px 지정 좌우 중앙정렬
텍스트 중앙정렬 배경색 흰색
*/
table{
margin-left:auto;
margin-right:auto;
width:70%;
}/* 표 만들기(명함 등록 폼을 표로 생성)
바깥쪽여백 좌우 auto로 중앙정렬
가로 세로 크기 = 800X500
*/

tr,td{
border-bottom:4px solid #C4B9F5;
padding-left:10px
}
/*생성 한 표 내 행과 열 스타일
아래 테두리만 4px 실선으로 지정 색상 지정
안쪽 여백 10px
*/
</style>
</head>


등록된 폼의 어떤 정보를 입력하는지 표로 타이틀을 입력합니다.
<body bgcolor="#C4B9F5"><!-- 배경 색상 -->
<div id="wrap"><!-- 전체 상자를 wrap을 id선택자로 입력. -->
<h2 style="padding:40px;"><font size="9" color="gray">등 록 결 과</font></h2>
<!-- h2태그를 이용하여 제목을 입력. 안쪽 여백을 40px을 주고 font태그로 글자 크기를 9
회색으로 지정. -->
<table border="0" align="center">
<!-- 표만들기 테두리를 0으로 지정
입력된 명함을 한번에 보기위한 표를 생성합니다-->
<tr><!-- 행 정렬 -->
<th width="200">이름</th>
<th width="300">부서</th>
<th width="300">직책</th>
<th width="500">이메일(E-mail)</th>
<th width="500">전화번호</th>
<th width="900">주소</th>
<th width="900">상세주소</th>
</tr>
</div>


```

<%

```

Connection conn=null;//connection은 데이터베이스와 연결하는 객체
PreparedStatement pstmt=null;// sql문 전송하는 객체인 PreparedStatement
ResultSet rs=null;

```

```

/*예외 처리 부분*/
try{

```

```

    /*연동할 DB를 포함한 url*/

```

```

    String

```

```

jdbcUrl="jdbc:mysql://localhost:3306/basicjsp?serverTimezone=Asia/Seoul";

```

```

    String dbId="jspid";//사용자 계정

```

```

    String dbPass="jsppass";//사용자 패스워드

```

```

/*DriverManager에 등록*/

```

```

Class.forName("com.mysql.cj.jdbc.Driver");

```

```

/*Connection 객체의 url,id,pass를 얻어옴.*/

```

```

conn=DriverManager.getConnection(jdbcUrl,dbId ,dbPass );

```

```

//membertbl의 테이블의 전체 조건을 찾음

```

```

String sql= "select * from memberTBL";

```

```

//Sql의 Statement 클래스의 기능 향상

```

```

pstmt=conn.prepareStatement(sql);

```

```

//객체의 값을 반환 select 구문을 수행할 때 사용하는 함수

```

```

rs=pstmt.executeQuery();

```

```

while(rs.next()){

```

```

    /*PreparedStatement 객체가 제공하는 getString 메소드*/

```

```

String name= rs.getString("name");

```

```

String department= rs.getString("department");

```

```

String position= rs.getString("position");

```

```

String email= rs.getString("email");

```

```

String tell= rs.getString("tell");

```

```

String member_addr= rs.getString("member_addr");

```

```

String detail_addr= rs.getString("detail_addr");

```

%>

```

<!-- request 내장객체의 getParameter() 메소드를 사용하여 읽어온 id 프로퍼티 값을 표현식을 사용하여 화면에 출력. -->

```

```

<tr>

```

```

    <td width="100"><%=name%></td>

```

```

    <td width="100"><%=department%></td>

```

```

    <td width="100"><%=position%></td>

```

```

    <td width="200"><%=email%></td>

```

```

    <td width="200"><%=tell%></td>

```

```

    <td width="200"><%=member_addr%></td>

```

```

    <td width="200"><%=detail_addr%></td>

```

```

</tr>

```

<%

```

}

```

```

}catch(Exception e){e.printStackTrace();} //Exception e가 발생할 경우 작성하는

```

문구

```

finally{//try 문에서 더이상의 실행이 필요 없을 경우 마지막으로 입력하는 문구

```

```

    if(rs != null)

```

```

        try{rs.close();}catch(SQLException sqle){}

```

```

    if(pstmt != null)

```

```

        try{pstmt.close();}catch(SQLException sqle){}

```

```

    if(conn != null)

```

```

        try{conn.close();}catch(SQLException sqle){}

```

```

}

```

%>

```

</table>

```

```

</body>

```

```

</html>

```

모바일 웹 폼 실행

linkTestForm.jsp

```
<!--JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. --%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<!--meta 태그: 브라우저에게 웹 페이지 관련 부가 정보를 제공하는 태그-->
<!-- 뷰포트 선언 -->
<!--content="width=device-width;= 화면 뷰 포트의 너비
    initial-scale=1.0;=초기 화면 확대비율
    minimum-scale=1.0;=화면 축소 최소비율
    maximum-scale=1.0;=화면 확대 최대 비율
    user-scalable=yes;"=사용자의 확대 혹은 축소 가능 여부
-->
```

meta 태그를 이용하여 뷰 포트를 선언하기 위해 head태그 안에 meta태그를 포함 시킵니다.
device-width는 화면 뷰포트의 너비를 지정하며 초기 화면 확대 비율을 initial-scale의 속성을
이용합니다.minimum은 화면 축소 최소 비율을 maimum은 화면 확대 최대 비율을 나타냅니다.

```
<meta
    charset="UTF-8"

    name="viewport"
    content="width=device-width;
    initial-scale=1.0;
    minimum-scale=1.0;
    maximum-scale=1.0;
    user-scalable=yes;"
/>

<!-- 모바일 입력 폼의 스타일을 지정합니다 -->

<style>

table{

    background: #fff;
    margin-left:auto;
    margin-right:auto;

    /* 표 만들기(명함 등록 폼을 표로 생성)
    바깥쪽여백 좌우 auto로 중앙정렬
    가로 세로 크기 = 800X500
    */

    tr,td{
        border-bottom:4px solid #fff;
        padding-left:10px
    }/*생성 한 표 내 행과 열 스타일
    아래 테두리만 4px 실선으로 지정 색상 지정
    안쪽 여백 10px
    */
```

```

input{
    height:35px;
    font-size: 15px;
    border: 0;
    border-radius: 25px;
    outline: none;
    padding-left: 10px;
    background-color: rgb(233, 233, 233);
    margin-top: 10px;
}/* input태그 스타일
가로 세로 크기 300X400
테두리 없음(0으로 지정)/모서리 둥글게(25px)
안쪽여백 10px 배경 색지정
*/
input:focus{
    background-color: #E2DBFF;
}/* input태그 상자를 클릭하는 경우(선택자뒤에 ':foucus'를 입력)
배경색 지정
*/

.sub{
    margin:10px 60px;
    color: #000;
    border: 1px solid #C4B9F5;
    border-radius: 25px;
    font-weight: bold;
    font-size: 12PX;
    width:200px;
    height:50px;
    background-color:#fff;
    cursor:pointer;
}/* sub라는 class객체를 지정한 스타일(명함 입력,수정,조회 버튼)
가로 세로 길이 150X50 배경색상 흰색
글자크기 16px 굵게
테두리 둥글게(25px)
버튼에 커서를 가져다 대면 포인터로 바꿈
*/

.sub:hover{
    color: #fff;
    background-color:#C4B9F5;
}/* .sub에 :hover를 같이 입력하면 마우스 커서를 올리면 배경색상과 폰트 색상을 변경함
*/

```

미디어 스크린의 크기의 최대 값을 320px로 지정, 세로보기 모드일 때 가로 최솟값이 321px, 가로보기 모드일 때 가로 최소값을 321px로 지정하고 배경색과 글자색을 지정합니다.

```

@media only screen and (max-width: 320px) {
    * { background: #C4B9F5; color: white; }
}

/* 세로보기모드 */
@media only screen and (min-width: 321px) and (orientation: portrait) {
    * { background: #C4B9F5; color: black; }
}

/* 가로보기모드 */
@media only screen and (min-width: 321px) and (orientation: landscape) {
    * { background: #C4B9F5; color: black; }
}
</style>
<title>외부 프로그램 링크</title>
</head>

```

모바일로 입력할 폼을 지정합니다.

input태그를 사용하여 입력 폼을 생성하고 placeholder속성을 이용하여 힌트를 주었습니다.

```
<body>
<h2 style="text-align:center; padding:40px;"><font size="6" >모 바 일 등 록</font></h2>
  <form method="post" action="linkTestProc.jsp" >

    <!-- 표만들기 테두리를 0으로 지정 cellpadding은 표의 여백을 2로 지정 -->
    <table border="0" cellspacing="0" cellpadding="2" align="center">

      <!--이메일 입력-->
      <tr> <!-- 행 정렬 -->
        <td ><label for="email">EMAIL</label></td><!-- 데이터 셀 지정, 열 정렬-->
        <td> <!-- 데이터 셀 지정, 열 정렬/가로 행 합치기 -->
          <input id="email" name="email" type="email" size="20"
placeholder="email@address.com" autofocus required><p>
          </td>
        </tr>

        <!--전화번호 입력(전화)-->
        <tr>
          <td><label for="phone">PHONE</label></td>
          <td>
            <input id="phone" name="phone" type="number" size="10"
placeholder="010-1234-1234"><p>
          </td>
        </tr>

        <!--전화번호 입력(SMS)-->
        <tr>
          <td><label for="sms">SMS</label></td>
          <td>
            <input id="sms" name="sms" type="number" size="10"
placeholder="010-1234-1234"><p>
          </td>
        </tr>

        <!--주소 입력-->
        <tr>
          <td><label for="addr">ADDRESS</label></td>
          <td>
            <input id="addr" name="addr" type="text" size="25" placeholder="주소를
입력해주세요."><p>
          </td>
        </tr>

        <tr>
          <td colspan="2">
            <input type="submit" value="S U B M I T" class="sub"></input>
          </td>
        </tr>

      </table>

    </form>

</body>
</html>
```


linkTestProc.jsp

```

<!-- JSP 페이지가 생성할 문서의 타입을 아래와 같이 지정하였습니다. -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- meta 태그: 브라우저에게 웹 페이지 관련 부가 정보를 제공하는 태그 -->
<!-- 뷰포트 선언 -->
<!-- 뷰포트 선언 -->
<!-- content="width=device-width;= 화면 뷰 포트의 너비
        initial-scale=1.0;=초기 화면 확대비율
        minimum-scale=1.0;=화면 축소 최소비율
        maximum-scale=1.0;=화면 확대 최대 비율
        user-scalable=yes;"=사용자의 확대 혹은 축소 가능 여부
        -->

<meta
    charset="UTF-8"

    name="viewport"
    content="width=device-width;
        initial-scale=1.0;
        minimum-scale=1.0;
        maximum-scale=1.0;
        user-scalable=no;"
/>

<% request.setCharacterEncoding("utf-8");%>

<style>

    img{
        width:40px;
    }

    table{

        background: #fff;
        margin-left:auto;
        margin-right:auto;

        /* 표 만들기(명함 등록 폼을 표로 생성)
        바깥쪽여백 좌우 auto로 중앙정렬
        가로 세로 크기 = 800X500
        */

        tr,td{
            border-bottom:4px solid #fff;
            padding-left:10px
        }/*생성 한 표 내 행과 열 스타일
        아래 테두리만 4px 실선으로 지정 색상 지정
        안쪽 여백 10px
        */

        input{
            height:40px;
            font-size: 15px;
            border: 0;
            border-radius: 25px;
            outline: none;
            padding-left: 10px;
            background-color: rgb(233, 233, 233);
            margin-top: 10px;
        }/* input태그 스타일
        가로 세로 크기 300X400
        테두리 없음(0으로 지정)/모서리 둥글게(25px)
        안쪽여백 10px 배경 색지정
        */

        input:focus{
            background-color: #E2DBFF;
        }/* input태그 상자를 클릭하는 경우(선택자뒤에 ':foucus'를 입력)
        배경색 지정
        */

```

```

.sub{
    margin:10px 60px;
    color: #000;
    border: 1px solid #C4B9F5;
    border-radius: 25px;
    font-weight: bold;
    font-size: 12PX;
    width:200px;
    height:50px;
    background-color:#fff;
    cursor:pointer;
}/* sub라는 class객체를 지정한 스타일(명함 입력,수정,조회 버튼)
가로 세로 길이 150X50 배경색상 흰색
글자크기 16px 굵게
테두리 둥글게(25px)
버튼에 커서를 가져다 대면 포인터로 바꿈
*/
.sub:hover{
    color: #fff;
    background-color:#C4B9F5;
}/* .sub에 :hover를 같이 입력하면 마우스 커서를 올리면 배경색상과 폰트 색상을 변경함
*/

@media only screen and (max-width: 320px) {
    * { background: #C4B9F5; color: white; }
}

/* 세로보기모드 */
@media only screen and (min-width: 321px) and (orientation: portrait) {
    * { background: #C4B9F5; color: black; }
}

/* 가로보기모드 */
@media only screen and (min-width: 321px) and (orientation: landscape) {
    * { background: #C4B9F5; color: black; }
}
</style>
<!-- request 내장객체의 getParameter() 메소드를 사용하여 읽어온 id 프로퍼티 값을 표현식을
사용하여 화면에 출력. -->

<%
    String email = request.getParameter("email");
    String phone = request.getParameter("phone");
    String sms = request.getParameter("sms");
    String addr = request.getParameter("addr");
%>

```

입력 폼의 input태그의 name속성에서 지정한 이름들을 getparameter값으로 읽어 온 후 폼에서
보일수 있게 합니다.

```

<body>

<h2 style="text-align:center; padding:40px;"><font size="6" >모 바 일 등 록 확
인</font></h2>
    <form method="post" action="linkTestForm.jsp" >

        <!-- 표만들기 테두리를 0으로 지정 cellpadding은 표의 여백을 2로 지정 -->
        <table border="0" cellspacing="0" cellpadding="2" align="center">

            <!--이메일 입력-->
            <tr> <!-- 행 정렬 -->
                <td> <!-- 데이터 셀 지정, 열 정렬/가로 행 합치기 -->
                    <!-- 이메일 링크 -->
                    <a href="mailto:<%=email %>">
                        
                    </a><p>
                </td>
            </tr>

```

```

<td><label for="email" style="font-weight:bold">EMAIL</label></td><!-- 데이터
셀 지정, 열 정렬-->
<td> <%=email%></td><!-- 데이터 셀 지정, 열 정렬/가로 행 합치기 -->

```

```

</tr>

```

a 태그 안에 href속성 값을 특정 응용프로그램으로 이동하는 값을 삽입하여 자동으로 관련 프로그램음을 호출하게 하였으며 target속성을 _blank를 이용하여 새 탭에서 열릴 수 있게 입력하였습니다. 이미지로 삽입한 아이콘을 누르면 해당 프로그램을 호출합니다.

request 내장 객체의 getParameter()메소드를 이용해 읽어온 id값을 입력하여 화면에 출력할 수 있게 합니다

```

<!--전화번호 입력(전화)-->

```

```

<tr>

```

```

    <td>

```

```

        <!-- 전화번호 링크 -->

```

```

        <a href="tel:<%=phone %> " target="_blank">

```

```

```

```

        </a><p>

```

```

    </td>

```

```

    <td><label style="font-weight:bold">Phone</label></td>

```

```

    <td> <%=phone%></td><!-- 데이터 셀 지정, 열 정렬/가로 행 합치기 -->

```

```

</tr>

```

```

<!--전화번호 입력(SMS)-->

```

```

<tr>

```

```

    <td>

```

```

        <!-- SMS문자링크 -->

```

```

        <a href="sms:<%=sms %> " target="_blank">

```

```

```

```

        </a><p>

```

```

    </td>

```

```

    <td><label style="font-weight:bold">SMS</label></td>

```

```

    <td> <%=sms%></td><!-- 데이터 셀 지정, 열 정렬/가로 행 합치기 -->

```

```

</tr>

```

```

<!--주소 입력-->

```

```

<tr>

```

```

    <td>

```

```

        <!-- 주소링크 -->

```

```

        <a href="http://maps.google.com/maps?q=<%=addr %> ">

```

```

```

```

        </a><p>

```

```

    </td>

```

```

    <td><label style="font-weight:bold">Adress</label></td>

```

```

    <td> <%=addr%></td><!-- 데이터 셀 지정, 열 정렬/가로 행 합치기 -->

```

```

</tr>

```

```

<tr>

```

```

    <td colspan="3">

```

```

        <input type="submit" value="S U B M I T" class="sub"></input>

```

```

    </td>

```

```

</tr>

```

```

</table>

```

```

</body>

```

3. PC 웹 실행 결과 (결과 화면)

등록

명함 등록

이름 장보고

회사명 경성대학교 직책 대학원생

이메일 ks0000@ks.co.kr

전화번호 010-1234-2222

우편번호 48434 검색

주소 부산 남구 수영로 309

상세주소 결성대학교

명함 입력 명함 수정 명함 조회

회원 등록이 완료되었습니다 :

명함 조회 버튼 클릭

| 이름 | 회사명 | 직책 | 이메일(email) | 전화번호 | 주소 | 상세주소 |
|-----|-------|------|-----------------|---------------|-------------------|-------|
| 장보고 | 경성대학교 | 대학원생 | ks0000@ks.co.kr | 010-1234-2222 | 부산 남구 수영로 309 | 결성대학교 |
| 홍길동 | 결성대학교 | 학생 | email@email.com | 010-1111-1234 | 부산 남구 수영로 309 | 결성대학교 |
| 고영희 | 부산대학교 | 학생 | ks@email.com | 010-0000-1234 | 부산 남구 수영로 309 | 부산대학교 |
| 김민준 | 동원대학교 | 학생 | dong@email.com | 010-7777-8888 | 부산 부산진구 영정로 12-17 | 동원대학교 |

명함 수정 버튼 클릭

명함 등록

이름 장보고

회사명 경성대학교 직책 대학원생

이메일 ks0000@ks.co.kr

전화번호 010-1234-2222

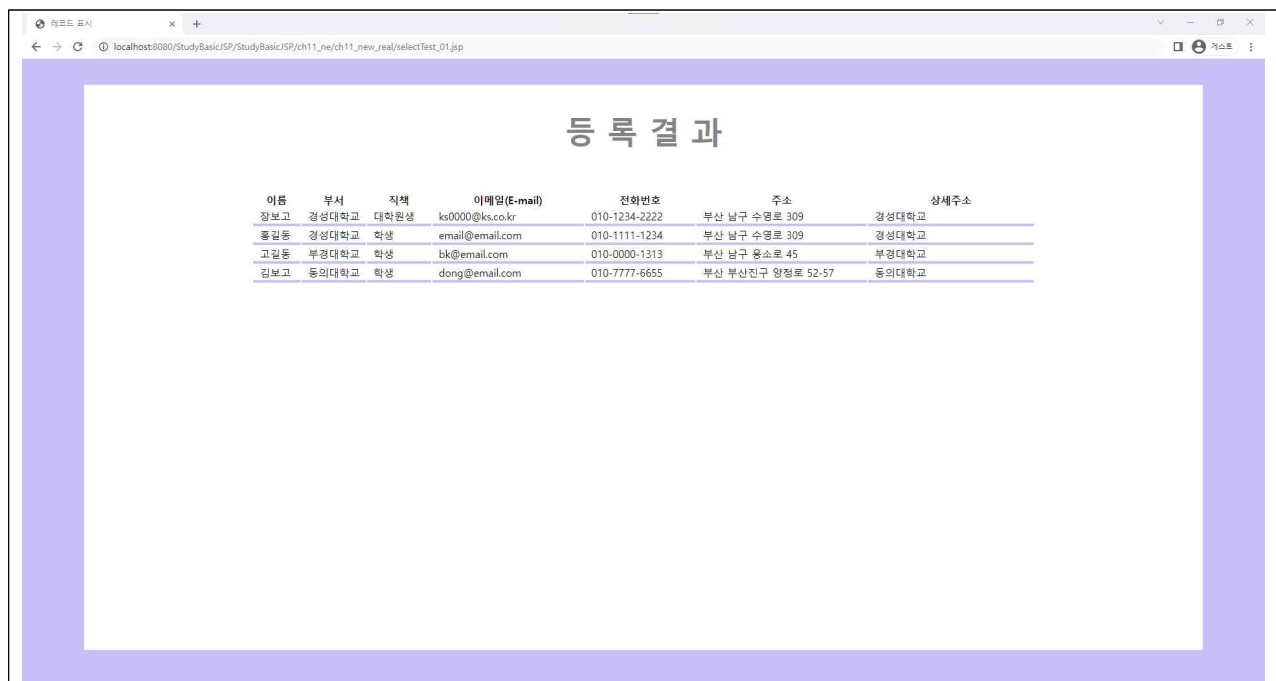
우편번호 48434 검색

주소 부산 남구 수영로 309

상세주소 결성대학교

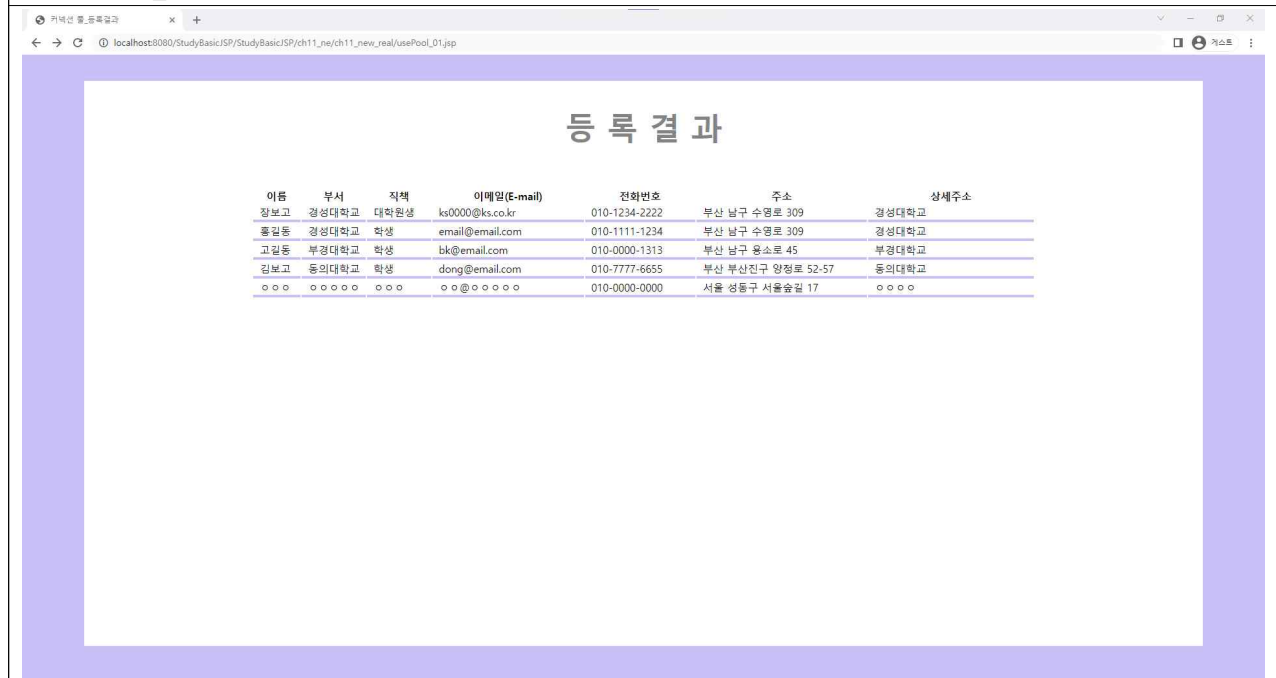
명함 입력 명함 수정 명함 조회

등록 결과



| 이름 | 부서 | 직책 | 이메일(E-mail) | 전화번호 | 주소 | 상세주소 |
|-----|-------|------|-----------------|---------------|-------------------|-------|
| 장보고 | 경성대학교 | 대학원생 | ks0000@ks.co.kr | 010-1234-2222 | 부산 남구 수영로 309 | 경성대학교 |
| 홍길동 | 경성대학교 | 학생 | email@email.com | 010-1111-1234 | 부산 남구 수영로 309 | 경성대학교 |
| 고길동 | 부경대학교 | 학생 | bk@email.com | 010-0000-1313 | 부산 남구 용소로 45 | 부경대학교 |
| 김보고 | 동의대학교 | 학생 | dong@email.com | 010-7777-6655 | 부산 부산진구 양정로 52-57 | 동의대학교 |

selectTest_01



| 이름 | 부서 | 직책 | 이메일(E-mail) | 전화번호 | 주소 | 상세주소 |
|-----|-------|------|-----------------|---------------|-------------------|-------|
| 장보고 | 경성대학교 | 대학원생 | ks0000@ks.co.kr | 010-1234-2222 | 부산 남구 수영로 309 | 경성대학교 |
| 홍길동 | 경성대학교 | 학생 | email@email.com | 010-1111-1234 | 부산 남구 수영로 309 | 경성대학교 |
| 고길동 | 부경대학교 | 학생 | bk@email.com | 010-0000-1313 | 부산 남구 용소로 45 | 부경대학교 |
| 김보고 | 동의대학교 | 학생 | dong@email.com | 010-7777-6655 | 부산 부산진구 양정로 52-57 | 동의대학교 |
| ㅇㅇㅇ | ㅇㅇㅇㅇㅇ | ㅇㅇㅇ | ㅇㅇ@ㅇㅇㅇㅇㅇ | 010-0000-0000 | 서울 성동구 서울숲길 17 | ㅇㅇㅇㅇ |

usePool_01

수정

회원 정보 변경

이름

이름을 입력해주세요

전화번호

010-0000-1111

직책

직책을 입력해주세요

입력 완료

다시 작성

회원 정보 변경

이름

홍보국

전화번호

010-1234-2222

직책

부인사장

입력 완료

다시 작성

회원 정보를 수정했습니다.

회원 정보 변경

이름

홍보국

전화번호

010-1234-2222

직책

교수

입력 완료

다시 작성

이름을 다시 입력해주세요길 바랍니다.

회원 정보 변경

이름

홍보국

전화번호

010-1234-2222

직책

교수

입력 완료

다시 작성

전화번호를 다시 입력해주세요길 바랍니다.

등록 결과

| 이름 | 성 | 직책 | 이메일주소(email) | 전화번호 | 주소 | 생년월일 |
|-----|-------|------|-----------------|---------------|------------------|-------|
| 홍보국 | 홍보국 | 대학원생 | ku0000@ku.ac.kr | 010-1234-2222 | 부산 남구 수영로 309 | 경원대학교 |
| 홍길동 | 경원대학교 | 학생 | email@email.com | 010-1111-1234 | 부산 남구 수영로 309 | 경원대학교 |
| 고길동 | 부산대학교 | 학생 | ku@gmail.com | 010-0000-1111 | 부산 남구 통도사 45 | 부산대학교 |
| 김보경 | 홍익대학교 | 학생 | dong@net.com | 010-7777-6665 | 부산 북진로 52-57 | 홍익대학교 |
| 문정미 | 부산대학교 | 교수 | | 010-0000-0000 | 서울 영등포구 서울영등길 17 | 부산대학교 |

삭제

회원 탈퇴

이름

이름을 입력해주세요

전화번호

010-0000-1111

입력 완료

다시작성

회원 탈퇴

이름

이름을 입력해주세요

전화번호

010-0000-0000

입력 완료

다시작성

회원 정보를 수정했습니다.

회원 탈퇴

이름

이름을 다시 입력해주세요

전화번호

010-0000-0000

입력 완료

다시작성

이름을 다시 입력해주세요 바랍니다.

회원 탈퇴

이름

이름을 다시 입력해주세요

전화번호

010-0000-0000

입력 완료

다시작성

이름을 다시 입력해주세요 바랍니다.

회원 탈퇴

이름

이름을 다시 입력해주세요

전화번호

010-0000-0000

입력 완료

다시작성

전화번호를 다시 입력해주세요 바랍니다.

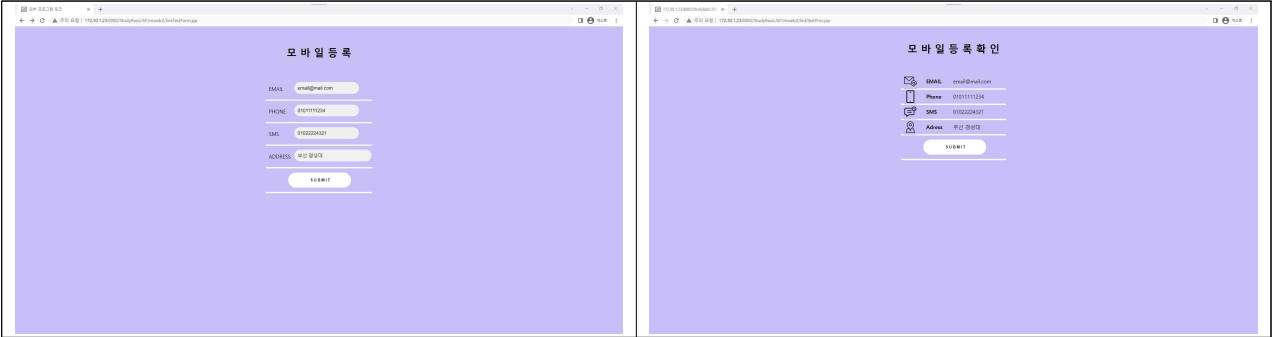
등록 결과

| 이름 | 부서 | 직책 | 이메일(email) | 전화번호 | 주소 | 생년월일 |
|-----|-----|-----|-----------------|---------------|---------------|------|
| 김보경 | 부서 | 직책 | ks0000@ks.co.kr | 010-1234-2222 | 부산 남구 수영로 309 | 김보경 |
| 홍정호 | 홍정호 | 홍정호 | email@email.com | 010-1111-1234 | 부산 남구 수영로 309 | 홍정호 |
| 김보경 | 부서 | 직책 | ks0000@ks.co.kr | 010-0000-1111 | 부산 남구 수영로 309 | 김보경 |
| 김보경 | 부서 | 직책 | email@email.com | 010-7777-6655 | 부산 남구 수영로 309 | 김보경 |

| | | |
|----------------------------------|------|---------|
| DB 연동 명함관리시스템의 명함입력 및 조회 프로그램 작성 | 문서분류 | 공학/보고서 |
| | 작성자 | 강정미/김보경 |

4. 모바일 웹 실행 결과 (결과 화면)


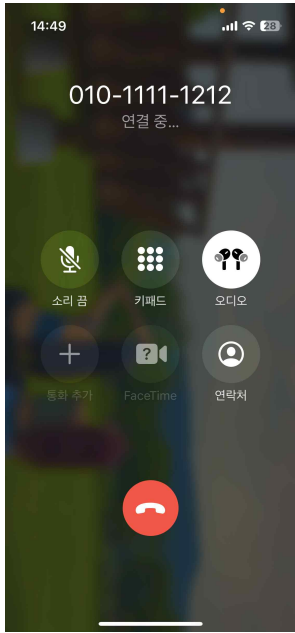

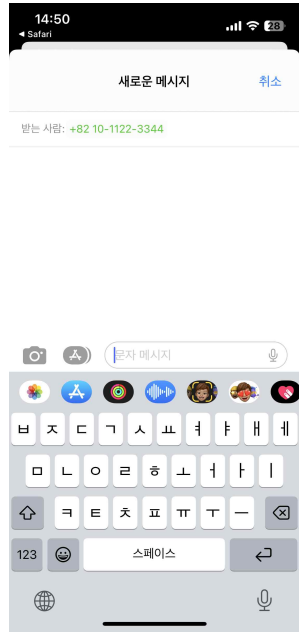
웹으로 실행




모바일로 실행



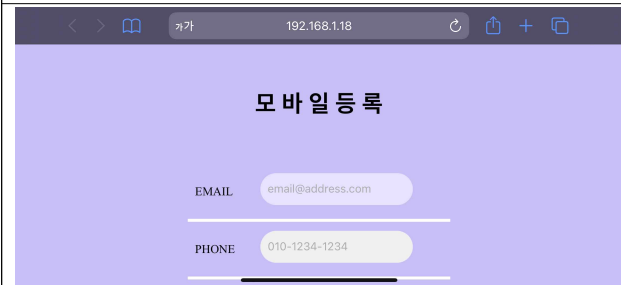
| | | |
|----------------------------------|------|---------|
| DB 연동 명함관리시스템의 명함입력 및 조회 프로그램 작성 | 문서분류 | 공학/보고서 |
| | 작성자 | 강정미/김보경 |


| | | | |
|---|---|--|---|
| 전화 아이콘을 클릭시 통화 앱으로 실행 | 전화가 바로 걸린다 | 메시지 아이콘을 클릭 시 메시지 앱으로 실행 | 문자 보낼수 있게 실행 된다 |
|  |  |  |  |

| |
|---|
| 지도 아이콘을 클릭하 면 바로 구글 지도로 실행됩니다 |
|  |

가로모드 실행 화면

표의 기본적인 크기가 커서 스크롤을 해야 합니다





5. 결 론

모바일 웹 연동

제작한 웹을 연동하기 위하여 해당 윈도우의 IPv4주소를 알아야 합니다.

1. cmd창에서 ipconfig를 입력하여 IPv4 주소를 알아냅니다.

```
C:\Users\kjm80>ipconfig

Windows IP 구성

무선 LAN 어댑터 로컬 영역 연결* 1:

    미디어 상태 . . . . . : 미디어 연결 끊김
    연결별 DNS 접미사 . . . . . :

무선 LAN 어댑터 Wi-Fi:

    연결별 DNS 접미사 . . . . . :
    링크-로컬 IPv6 주소 . . . . . :
    IPv4 주소 . . . . . :
    서브넷 마스크 . . . . . :
    기본 게이트웨이 . . . . . :
```

2. 웹에서 모바일을 연동하기 위해 톰캣 서버를 이용합니다.(startup 파일 실행)

```
Tomcat

??덱?? ?웁?? ?쥬--[C:\Wapache-tomcat-8.5.69\webapps\src]??덱? ?덱?? ?(5)??
??덱?? ?웁?? ?쥬--[C:\Wapache-tomcat-8.5.69\webapps\src]??덱? ?덱?? ?(5)??
13-Dec-2022 05:21:49.695 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory ???덱?? ?(5)??
??덱?? ?웁?? ?쥬--[C:\Wapache-tomcat-8.5.69\webapps\src]??덱? ?덱?? ?(5)??
13-Dec-2022 05:21:49.695 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory ???덱?? ?(5)??
??덱?? ?웁?? ?쥬--[C:\Wapache-tomcat-8.5.69\webapps\src]??덱? ?덱?? ?(5)??
```

3. 톰캣 서버가 실행이 되면 <http://ip주소:8080/StudyBasicJSP/mwep/linkTestForm.jsp>를 입력하여 웹에서 해당 파일이 실행이 되는지 확인합니다.

4. 웹에서 실행이 된 것을 확인하면 핸드폰에서 상단에 있는 주소를 그대로 입력하여 모바일에서도 실행이 되는지 확인을 합니다.

모바일에서 보이기 위해 <meta>태그에 뷰 포트를 입력하여 화면크기와 해상도가 다양하고 PC화면과 비슷한 해상도로 표시되므로 인식하기 어려울 정도의 작은 글씨로 표시되는 문제점을 해결합니다.

해당 태그 안에 너비 높이 확대 최대 최소 비율의 속성을 입력하여 화면의 크기를 조절합니다.

| 뷰포트 속성 | 속성값 | 의미 |
|---------------|--------------------------------------|-----------------------------|
| width | 예) 480px, 720px, 980px, device-width | 화면(뷰포트)의 너비(200px~10,000px) |
| height | 예) 800px, 1280px, device-height | 화면(뷰포트)의 높이 |
| initial-scale | 예) 1.0 | 초기 화면 확대 비율(0~10.0) |
| minimum-scale | 예) 0.5 | 화면 축소 최소 비율(0~10.0) |
| maximum-scale | 예) 2.0 | 화면 확대 최대 비율(0~10.0) |
| user-scalable | 예) yes(기본값), no | 사용자의 확대 혹은 축소 가능 여부 |

| | | |
|----------------------------------|------|---------|
| DB 연동 명함관리시스템의 명함입력 및 조회 프로그램 작성 | 문서분류 | 공학/보고서 |
| | 작성자 | 강정미/김보경 |

모바일에서 외부 프로그램 연결하기 위해 <a>태그의 href속성에 외부프로그램 주소를 입력합니다.

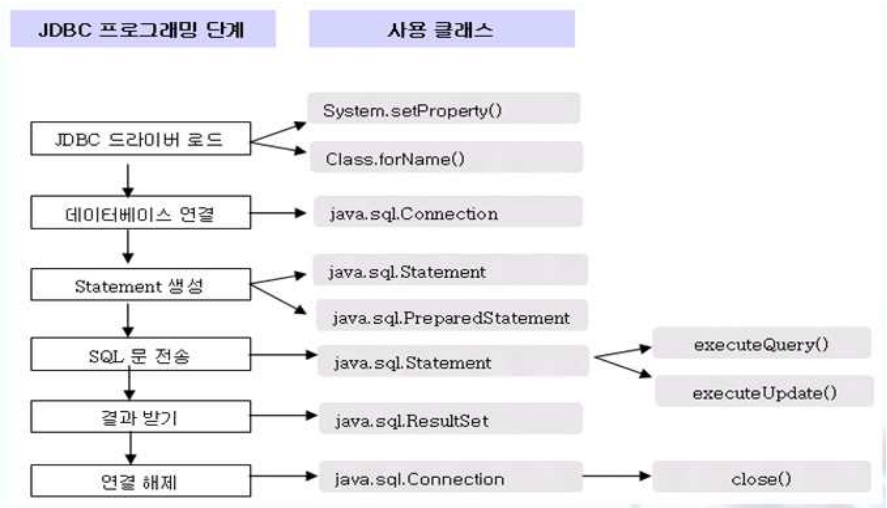
| | |
|-------|--------------------------------|
| 이메일 | mailto |
| 전화번호 | tel |
| sms문자 | sms |
| 지도 | http://maps.google.com/maps?q= |

JDBC

JDBC : 자바 프로그램의 표준화된 데이터 베이스 접근 인터페이스를 제공한다

응용프로그램이 JDBC드라이버에서 제공하는 API 함수를 이용하여 DBMS의 접근 제어가 가능해진다.

DBMS가 변해도 응용프로그램은 수정하지 않아도 된다(독립성 보장) DB변경시 JDBC 드라이버만 교체
프로그래밍 단계



DBMS : 데이터 베이스를 구성하고 운영하는 시스템 소프트웨어로 Mysql을 이용합니다.

SQL : 데이터베이스의 생성부터 레코드 검색 등의 작업을 수행합니다.

| | |
|--------------|-------------------------|
| 데이터 정의문(DDL) | CREATE, ALTER, DROP |
| 데이터 제어문(DCL) | GRANT, REVOKE |
| 데이터 조작문(DML) | UPDATE, INSERT, DELETE, |
| 쿼리(Query) | SELECT |
| 트랜잭션 처리 | COMMIT, ROLLBACK |

테이블 생성 create - create table tese{속성 값};

테이블 삭제 drop - drop table tese;

레코드 삽입 insert - insert into member values 값

레코드 검색 select - select * form member

레코드 수정 update - update member set passwd=1111 where id='aaa';

레코드 삭제 delete - delete from member where id = 'abc';

| | | |
|----------------------------------|------|---------|
| DB 연동 명함관리시스템의 명함입력 및 조회 프로그램 작성 | 문서분류 | 공학/보고서 |
| | 작성자 | 강정미/김보경 |

context 웹프로그래밍을 위한 물리적인 디렉토리 구조입니다

servers파일 내의 server.xml파일을 먼저 백업파일을 하나 만든 후 해당 파일에 StudyBasicJSP파일의 경로를 입력합니다

```
<!-- Eclipse 실행 -->
<Context docBase="StudyBasicJSP" path="/StudyBasicJSP" reloadable="true" source="org.eclipse.jst.jee.server:StudyBasicJSP">

<!-- Target 실행
<Context path="/StudyBasicJSP" docBase="c:\apache-tomcat-8.5.69\webapps\StudyBasicJSP">
-->
    <Resource name="jdbc/basicjsp"
        auth="Container"
        type="javax.sql.DataSource"
        driverClassName="com.mysql.cj.jdbc.Driver"
        username="jspid"
        password="jsppass"
        url="jdbc:mysql://localhost:3306/basicjsp?serverTimezone=Asia/Seoul"
        maxWait="5000"
    />
</Context>

</Host>
</Engine>
</Service>
</Server>
```

mysql을 실행하기 위해 해당 마일도 라이브러리에 꼭 넣어줍니다



http 통신 과정

클라이언트(브라우저) > request(get/post) > 웹서버(tomcat/eclips) >
response(get/post) > 클라이언트(브라우저)

JSP 한글처리

웹 페이지에 입력한 내용은 웹서버의 영향을 받습니다

웹 서버에서 지정한 값으로 인코딩 값을 결정하는데 contentType을 결정하고 charset으로 UTF-8방식을 지정해야 한글을 입력하였을 때 글자 깨짐 현상이 나타나지 않습니다.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.sql.*"%>
```