

PhotoWeb:

JSP와 MariaDB를 활용한 백엔드 프로그래밍 미니프로젝트

대한상공회의소 서울기술교육원
자바기반 빅데이터 시각화 시스템 과정
심윤경

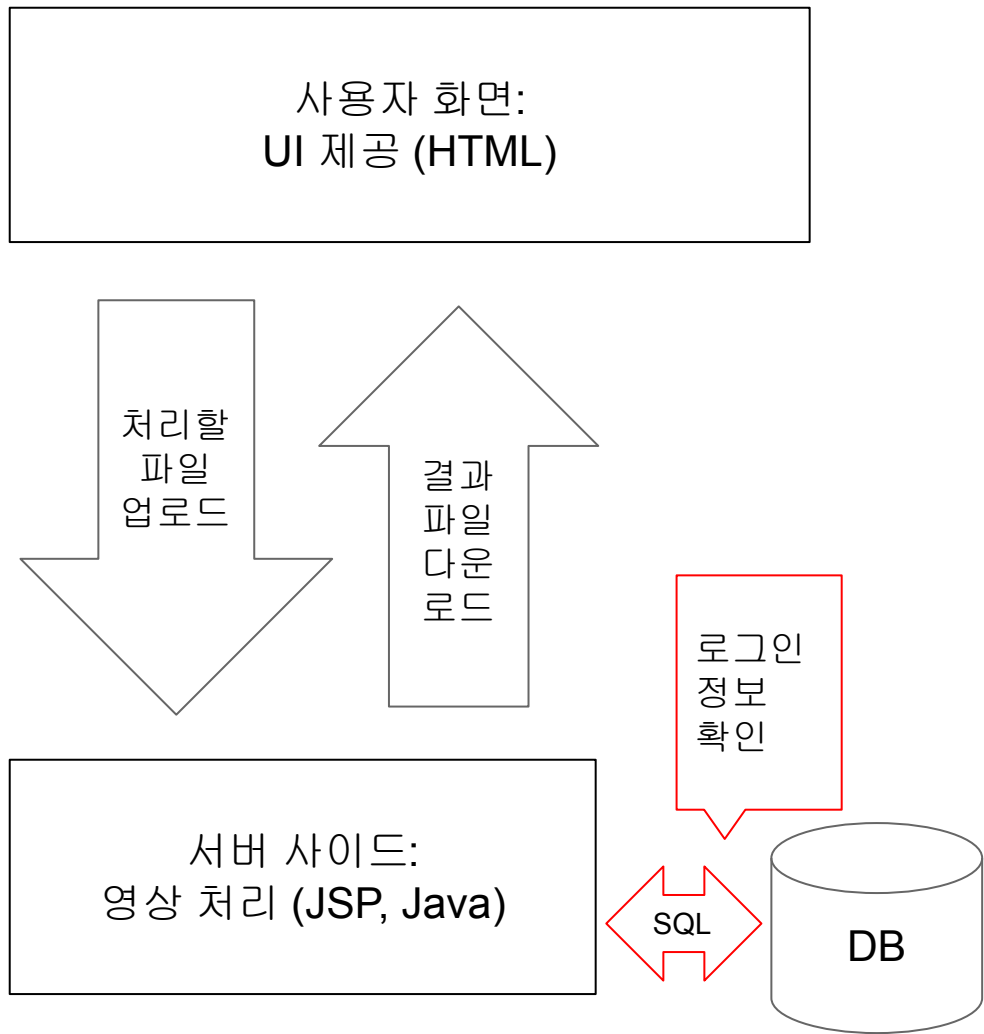
프로젝트 개요 & 학습 목표

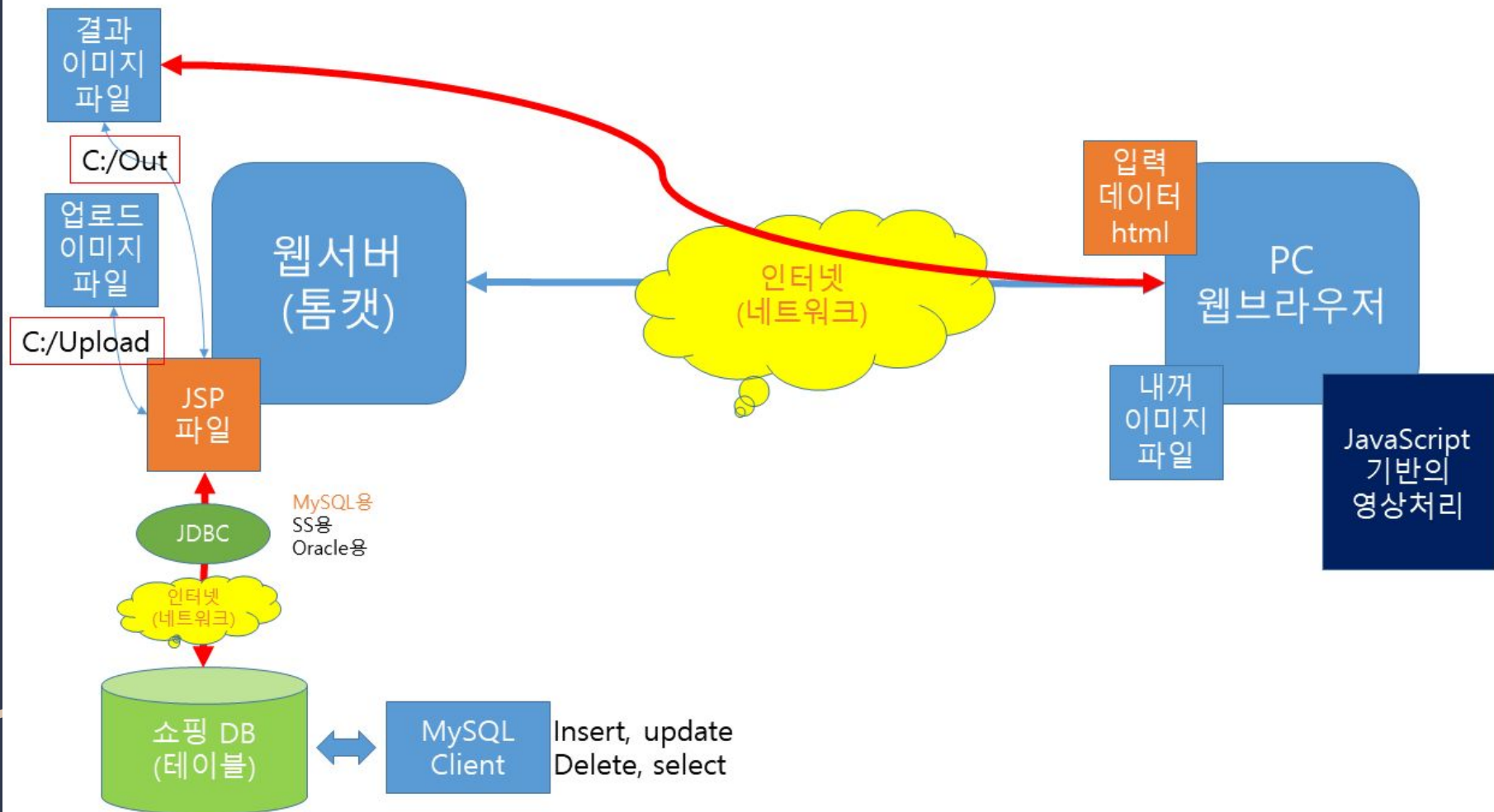
- JSP와 MariaDB를 이용한 백엔드 프로그래밍 익히기
 - 백엔드 프로그래밍을 위한 주요 도구 익히기
 - 백엔드 프로그래밍을 위한 환경 구성하기
- 이전에 배운 프론트엔드 프로그래밍과 연동하여 작동하는 영상처리 웹페이지 만들기

프로젝트 환경

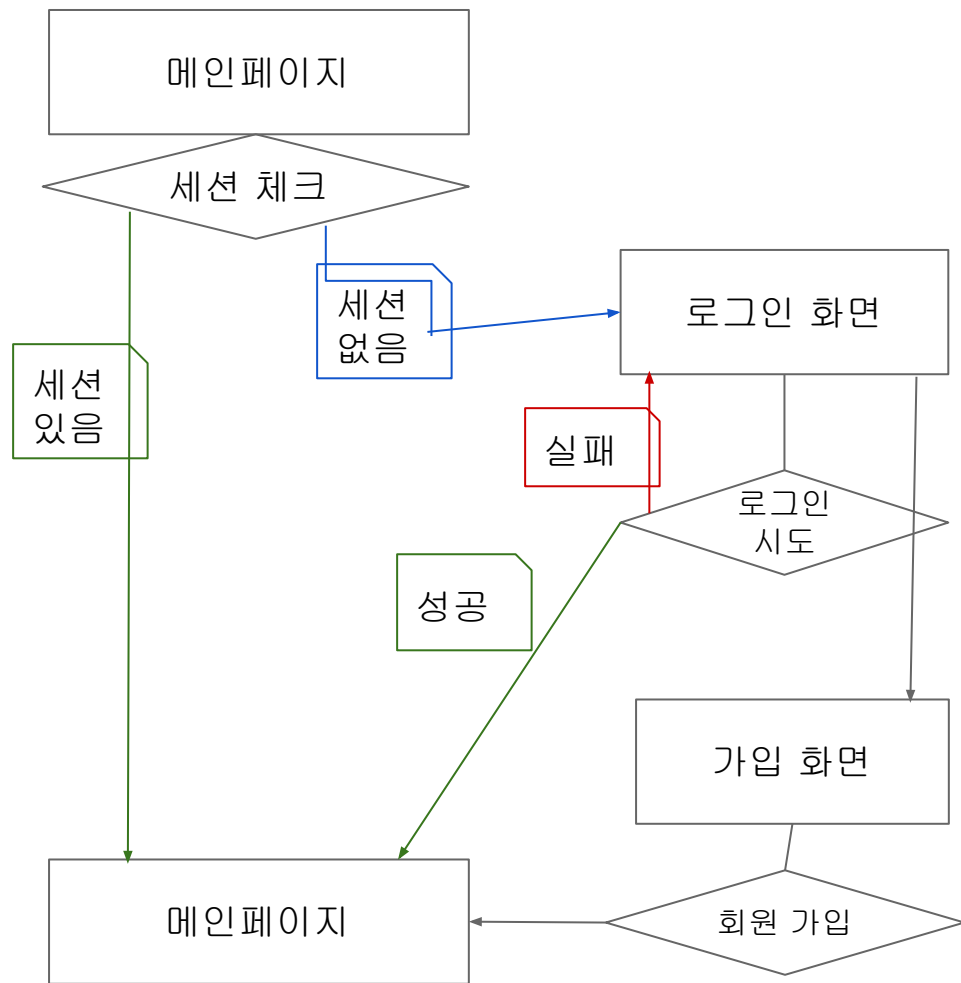
- 기계: Oracle VirtualBox 이용
- OS: Windows Server 2016 & Ubuntu 20.04
- Java: JDK 9
- Server: Apache Tomcat 10.0.5
- IDE: Eclipse 2021-03
- DB: MariaDB 10.5
- Browser: Chrome

PhotoWeb의 기본 구성

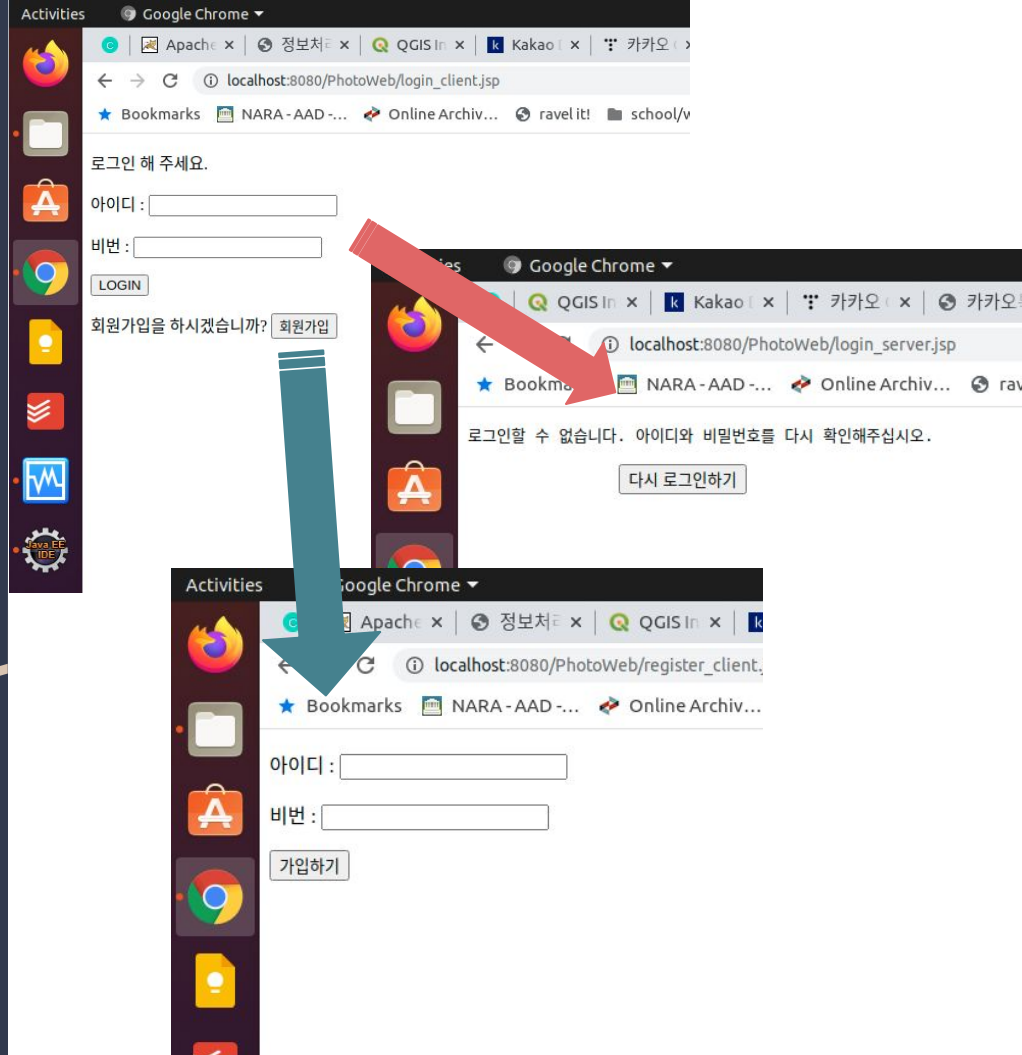




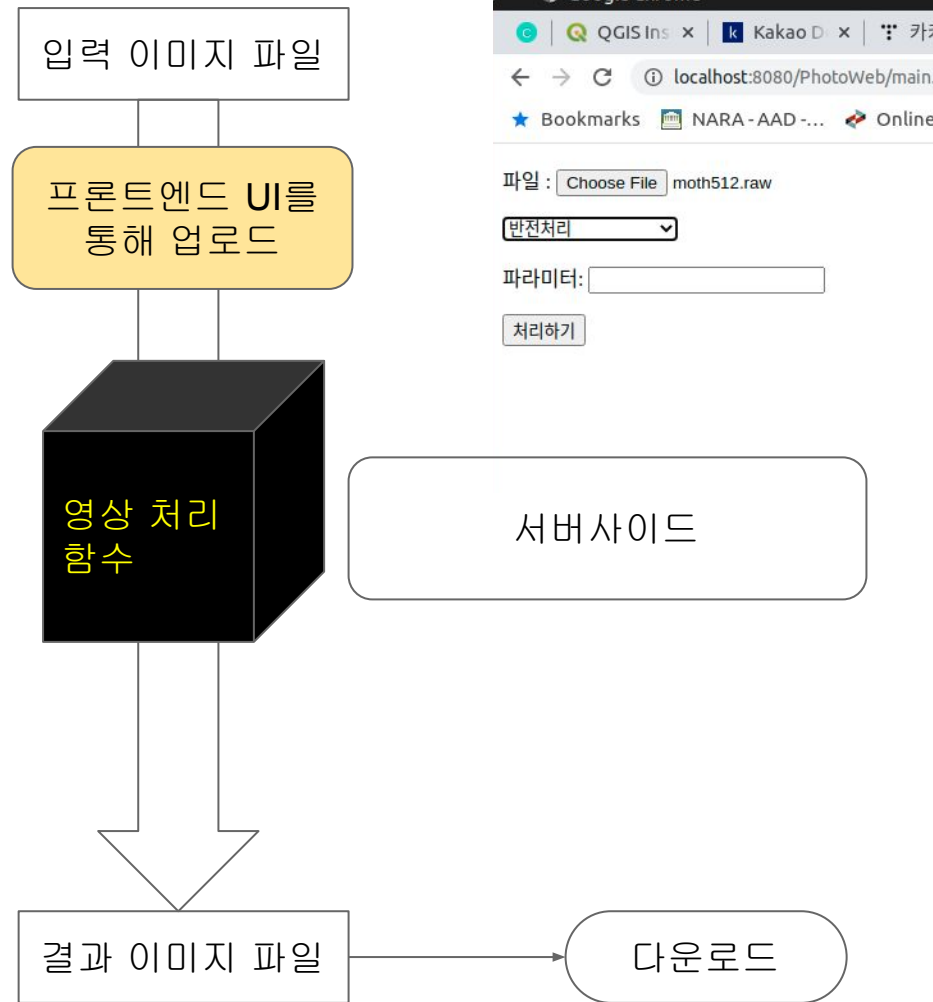
사용자 UI 흐름도 I



사용자 UI 흐름도 II



영상처리 로직 I



영상처리 로직 II

JSP

- 동적 웹페이지 개발
- HTML에 자바 코드 삽입
- 로직은 서블릿이나 자바 클래스로 독립시키기를 권장

PhotoWeb에서는

- DB에 접속해 회원가입과 로그인 작업
(dbConn.jsp, login_*.jsp, register_*.jsp, main.jsp)
- 이미지 파일의 업로드와 다운로드
(main_server.jsp)
- 이미지 파일을 읽어서 입력 배열 생성, 출력
배열에서 결과 파일 생성 (main_server.jsp)
- 영상처리 함수는 PhLogic이라는 자바 클래스에
넣고 import해서 사용 (PhLogic.java,
main_server.jsp)

PhLogic.java의 사용

```
Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
dbConn.jsp login_server.jsp login_client.jsp main.jsp main_server.jsp
1 package com.photoweb;
2
3 public class PhLogic {
4
5     public static int checkOverflow(int pix) {
6         int outPix = pix;
7         if (pix > 255)
8             outPix = 255;
9         else if (pix < 0)
10            outPix = 0;
11        return outPix;
12    }
13
14    public static int[][] toNeg(int[][] inImg) {
15        int[][] outImg = new int[inImg.length][inImg[0].length];
16
17        for (int i = 0; i < 256; i++) {
18            for (int k = 0; k < 256; k++) {
19                outImg[i][k] = 255 - inImg[i][k];
20            }
21        }
22        return outImg;
23    }
24
25    public static int[][] toBW(int[][] inImg, int param) {
26        int[][] outImg = new int[inImg.length][inImg[0].length];
27
28        for (int i = 0; i < outImg.length; i++) {
29            for (int k = 0; k < outImg[0].length; k++) {
30                outImg[i][k] = (inImg[i][k] > param) ? 255 : 0;
31            }
32        }
33        return outImg;
34    }
35}
```

```
Eclipse
5월 3일 05 : 04
eclipse-workspace - PhotoWeb/WebContent/main_server.jsp
File Edit Source Refactor Navigate Search Project Run Window Help
dbConn.jsp login_server.jsp login_client.jsp main.jsp main_server.jsp register_server.jsp register_client.jsp
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3 <%@ page import="java.io.*"%>
4 <%@ page import="com.oreilly.servlet.*"%>
5 <%@ page import="com.oreilly.servlet.multipart.*"%>
6 <%@ page import="java.util.*"%>
7 <%@ page import="java.io.*"%>
8 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
12 <title>Insert title here</title>
13 </head>
14 <body>
15     <%@ page import="com.photoweb.PhLogic"%>
16
17     // 클라이언트에서 전송한 정보 읽기
18     MultipartRequest multi = new MultipartRequest(request, "/home/zenbuntu20/photoweb/uploads", 5 * 1024 * 1024, "utf-8",
19         new DefaultFileRenamePolicy());
20
21     String filename, algo, param; // global params
22     String tmp;
23
24     outW = inW;
25     int[][] outImage = new int[outH][outW];
26
27     // switchboard
28     switch (algo) {
29     case "1": // 반전
30         outImage = PhLogic.toNeg(inImage);
31         break;
32     case "2": // 밝기 조절
33         outImage = PhLogic.adjBright(inImage, adjVal);
34         break;
35     case "3": // 흑백
36         outImage = PhLogic.toBW(inImage, adjVal);
37         break;
38     case "4": // 컨트라스트 조절
39         outImage = PhLogic.adjContr(inImage, adjVal);
40         break;
41     }
42 }
```

Ver.1.5 목표

- Javascript 연동으로 더 풍부한 UI 개발
 - 처리 전과 후의 이미지 화면에 출력
- 영상처리 함수 추가
- 결과 이미지 파일 다운로드 시 저장 위치와 파일 이름 지정