

---

# 데이터베이스 및 응용

## Project#1-2

### 튜토리얼 과정 및 이미지 업로드

---



Professor	이기훈 교수님
Department	컴퓨터공학과
Student ID	2012722028
Name	장 한 별
Date	2018. 9. 26.

# Introduction.

Project 1-2 에서는 Apache, PHP, MySQL를 설치 후 동작을 이해하고 이를 이용한 게시판 구축을 진행하는 것을 목표로 한다. 게시판은 쓰기, 보기, 수정, 삭제, 글 조회수 기능이 있어야하고, 추가로 게시판에 사진을 업로드하는 기능을 구현하는 것이 이번 프로젝트의 최종 목표이다.

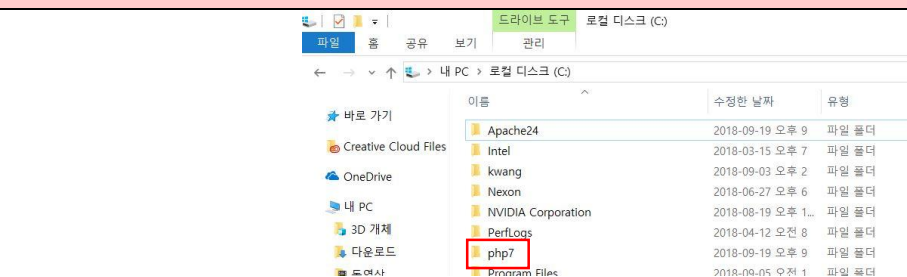
## Result.

### 1. Apache 설치

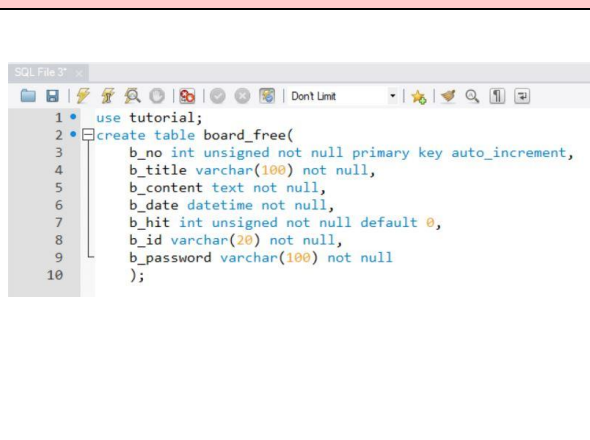


```
C:\WINDOWS\system32>httpd -k start  
C:\WINDOWS\system32>
```

### 2. PHP 설치

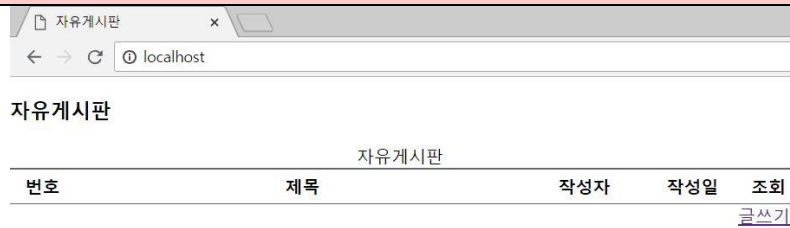


### 3. Database(MySQL) 설치



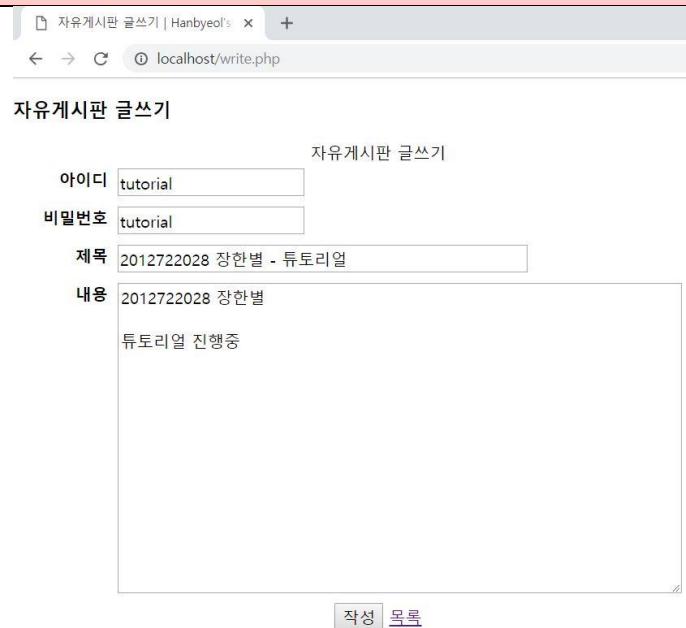
과정 1~3 그림을 통해 Apache, PHP, MySQL이 모두 정상적으로 설치되었음을 확인할 수 있다. 이를 통해 다음 내용을 이어갈 수 있는 환경이 구축되었으므로 계속 진행하도록 한다.

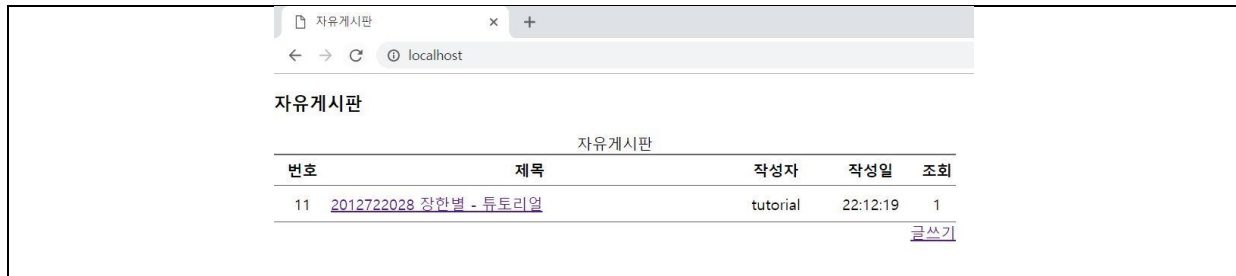
#### 4. 게시판 리스트 생성



위 그림은 게시판 리스트를 생성한 후 Web browser 화면이다. URL을 입력하는 곳에 'localhost(127.0.0.1)' 혹은 'localhost/index.php' 를 입력하면 위와 같은 결과를 확인할 수 있다. 이때 PHP와 MySQL이 연동되지 않을 시, 제대로 출력되지 않으므로, dbconfig.php 파일에서 mysqli() 함수를 잘 파악하여 올바른 순서로 값을 입력하는 것이 중요하다. 위 그림을 통해 올바르게 연동되었고, 게시판 역시 올바르게 생성이 잘 된 것을 확인할 수 있다.

#### 5. 게시글 쓰기





위 그림은 게시판에 게시글을 작성한 후 Web browser 화면이다. 아이디, 비밀번호, 제목, 내용 순으로 작성시, DataBase Attribute 인 b\_Id, b\_Password, b\_Title, b\_Content 에 실제 data인 instance (state)로써 저장되고, 위 그림과 같이 Web 으로 출력된다. Php 파일에서 SQL 문법을 정확하게 입력하는 것이 중요한데, 문자열을 표시 할 때 사용하는 큰 따옴표나 작은 따옴표의 구분이 Editor 마다 식별 차이가 발생해 튜토리얼에 주어진 코드 내용을 약간 수정했다. 이 부분은 'Conclusion.' 에서 자세히 다루도록 하겠다.

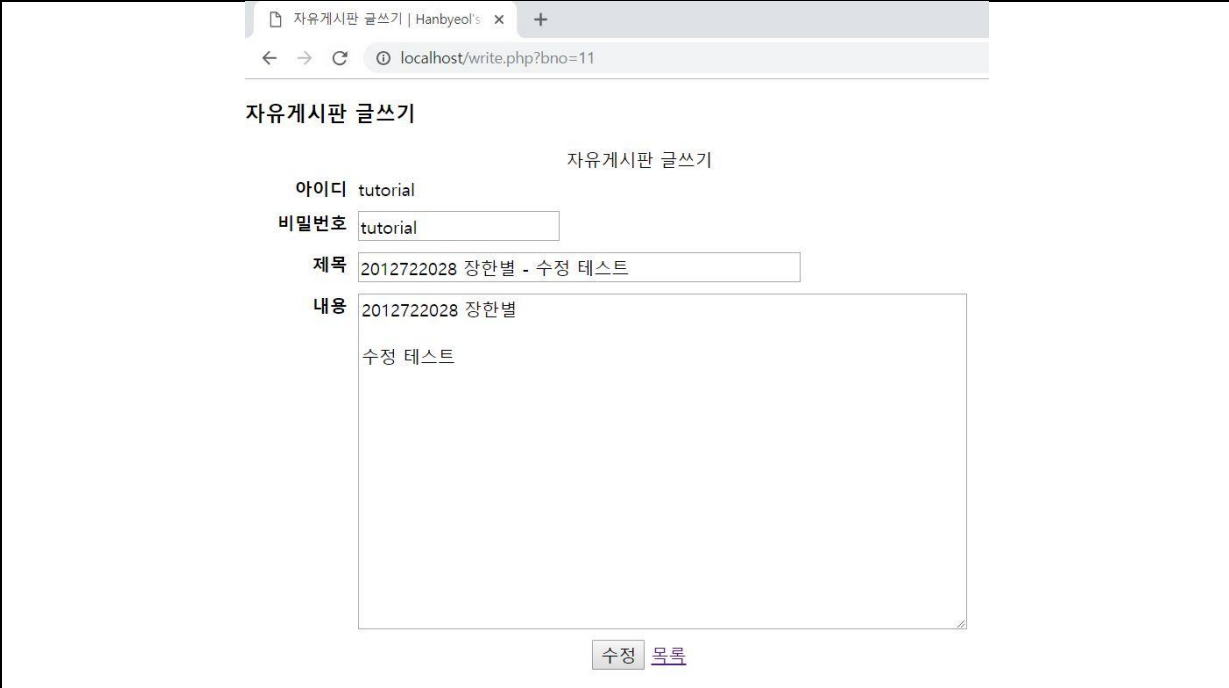
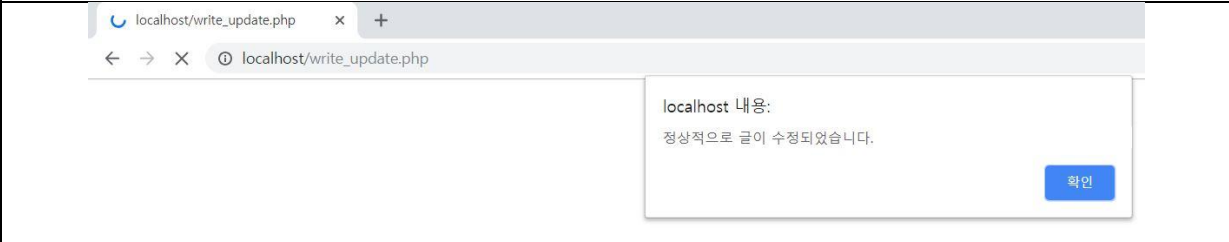

어찌됐든 위 화면과 같이 정상적으로 글이 등록되었음을 확인할 수 있다. 하지만 아직 view.php 파일을 작성하지 않아 작성한 게시글 확인은 불가하다.

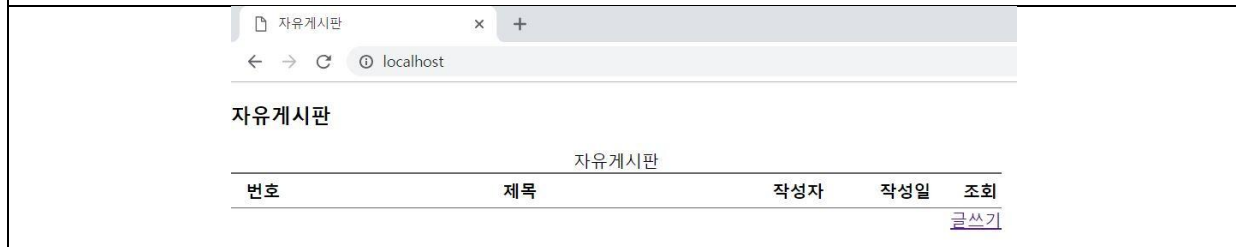
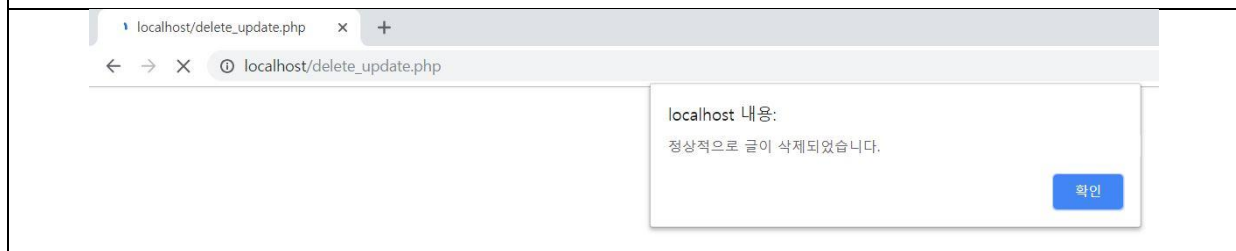
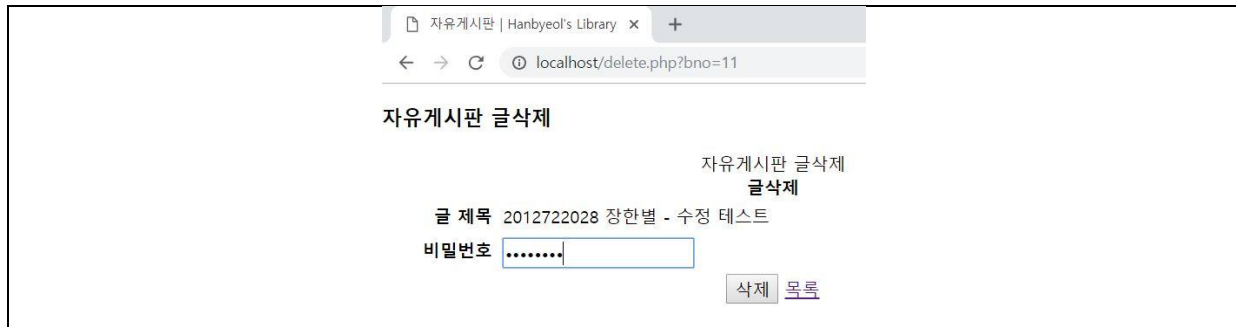
## 6. 게시글 보기



위 그림은 게시판에 접속 후 게시글을 확인한 상태의 Web browser 화면이다. view.php 를 작성 후 작성한 게시글을 확인할 수 있는 기능이 추가되었다. 위 그림과 같이 정상적으로 작동하는 것을 확인할 수 있다.

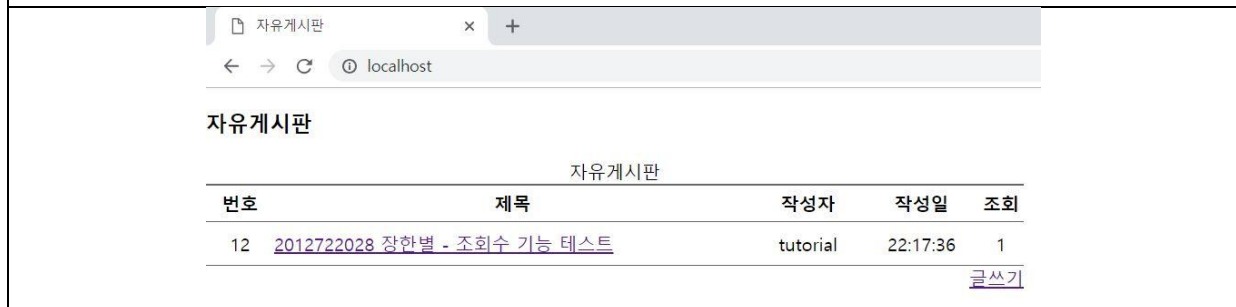
## 7. 게시글 수정




<p>위 그림은 게시글을 수정한 후의 Web browser 화면이다. write.php 와 write_update.php 파일의 내용을 수정 및 추가하면 위와 같은 결과를 얻을 수 있다. 수정을 원하는 게시글을 클릭해 들어간 후 제목이나 내용을 변경한 후 수정 버튼을 누르면 된다. 정상적으로 글이 수정되었음을 확인할 수 있다.</p>
<p><b>8. 게시글 삭제</b></p>



위 그림은 게시글을 삭제한 후의 Web browser 화면이다. view.php 파일의 내용을 수정하고, delete.php와 delete\_update.php 파일을 생성 하면 위와 같은 결과를 얻을 수 있다. 원하는 게시글을 선택하여 비밀번호를 입력 후 삭제 버튼을 클릭하면 삭제가 가능하다. 이때 당연하게도 비밀번호를 정확하게 입력해야 삭제할 수 있다. 위 그림을 통해 정상적으로 글이 삭제되었음을 확인할 수 있다.

## 9. 글 조회수 기능



위 그림은 게시글을 생성 후 조회 수를 확인할 수 있는 Web browser 화면이다. 게시판에 게시글을 생성 후 1번 조회 후, 조회수를 확인하면 '1'인 것을 확인할 수 있다. 사용자 브라우저의 쿠키를

사용하여, 만료되지 않은 쿠키가 존재하면 조회수를 올리지 않도록 설정되어 있어 그 이상 올라가지 않는다. 위 그림을 통해 올바르게 조회수를 볼 수 있음을 확인 할 수 있다.

## 10. 이미지 업로드

```
SQL File 3* x
1 • use tutorial;
2 • create table board_free(
3     b_no int unsigned not null primary key auto_increment,
4     b_title varchar(100) not null,
5     b_content text not null,
6     b_date datetime not null,
7     b_hit int unsigned not null default 0,
8     b_id varchar(20) not null,
9     b_password varchar(100) not null,
10    b_image BLOB not null
11 );
```

먼저, MySQL에서 기존 table인 board\_free 에 새로운 column 인 'b\_image' 를 추가한다. 'b\_image' 는 업로드할 이미지의 파일(경로)을 담는 column 으로, 파일의 크기가 모두 다르고 확정되지 않았기에 blob 형태로 생성하도록 한다.

```
mysql> show full columns from board_free;
```

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
b_no	int(10) unsigned	NULL	NO	PRI	NULL	auto_increment	select,insert,update,references	
b_title	varchar(100)	utf8mb4_0900_ai_ci	NO		NULL		select,insert,update,references	
b_content	text	utf8mb4_0900_ai_ci	NO		NULL		select,insert,update,references	
b_date	datetime	NULL	NO		NULL		select,insert,update,references	
b_hit	int(10) unsigned	NULL	NO		0		select,insert,update,references	
b_id	varchar(20)	utf8mb4_0900_ai_ci	NO		NULL		select,insert,update,references	
b_password	varchar(100)	utf8mb4_0900_ai_ci	NO		NULL		select,insert,update,references	
b_image	blob	NULL	NO		NULL		select,insert,update,references	

```
8 rows in set (0.00 sec)
mysql>
```

위 그림은 MySQL Command line 에서 board\_free table 에 존재하는 모든 column 들을 출력하는 명령어를 입력해 본 결과 화면이다. Command line에서도 'b\_image' 가 제대로 생성되었음을 확인할 수 있다. 이곳에 업로드할 이미지 파일을 담을 예정이다.

**<h3>자유게시판 글쓰기</h3>**

```
<div id="boardWrite">
    <form action="./write_update.php" method="post" enctype="multipart/form-data">
```

위 그림은 write.php 파일에서 이미지 전송을 위해 form tag를 수정한 화면이다.

Enctype = "multipart/form-data" 를 추가하여 이미지 파일이 server 에 전송될 수 있도록 수정했다.

```
<tr>
    <th scope="row"><label for="bImage">이미지</label></th>
    <td class="image">
        <input type="file" name="bImage" id="bImage"
            value="<?php echo isset($row['b_image'])? $row['b_image']:null?>"
        </td>
    </tr>
```

위 그림은 write.php 파일에서 input tage를 수정한 화면이다. Type 을 file로 설정했고, name 과 id

모두 bImage로 등록했다. value 값으로는 table 의 column 중 b\_image를 참조하여 넣어주도록 했다.

```
//글 등록
} else {
    $upload_image = basename($_FILES['bImage']['name']); // 파일명
    move_uploaded_file($_FILES['bImage']['tmp_name'], $upload_image);

    $sql = "insert into board_free (b_no, b_title, b_content, b_date, b_hit, b_id, b_password, b_image)
            values(null, '$bTitle', '$bContent', '$date', 0, '$bID', '$bPassword', '$upload_image)";
    $msgState = '등록';
}
```

위 그림은 write\_update.php 파일에서 수정한 화면이다. \$upload\_image 라는 변수에 업로드할 파일명을 저장 하도록 하고 'move\_uploaded\_file()' 함수를 사용해 이미지를 업로드 시켰다. SQL 명령어를 저장하는 \$sql 에 'b\_image'를 추가하고 values 값으로 방금 업로드한 파일명이 저장된 \$upload\_image 를 넣어 MySQL 과의 연동이 잘 이루어지도록 했다. 사소한 부분처럼 보이지만, 이 부분을 빼놓으면 web browser 에서도 출력이 되지 않는 오류가 발생해 매우 중요한 부분이라고 생각한다.

```
$sql = "select b_title, b_content, b_date, b_hit, b_id, b_image from board_free where b_no = '$bNo'";
$result = $db->query($sql);
$row = $result->fetch_assoc();
```

위 그림은 view.php 파일에서 수정한 화면이다. Write\_update.php 파일에서 \$sql 을 수정한 것과 마찬가지로, 여기서도 \$sql 에 b\_image 를 추가했다.

```
<div id="boardContent"><?php echo $row['b_content']?></div>
<div id="boardImage">

```

그 후 출력하는 부분에서 img tag 를 추가하여 업로드한 이미지가 제대로 출력될 수 있도록 조정했다. 이미지 파일이 너무 커서 web browser 의 50% 크기에 맞춰 출력하는 option을 넣었다. (width = "50%" )

자유게시판 글쓰기 | Hanbyeol's

← → ↻ localhost/write.php

자유게시판 글쓰기

아이디

비밀번호

제목

내용

이미지  DSC01487.JPG

[목록](#)



자유게시판 | Hanbyeol's Library

localhost/view.php?bno=48

자유게시판 글쓰기

2012722028 장한별

작성자: tutorial 작성일: 2018-09-26 14:58:31 조회: 1

2012722028 장한별



[수정](#) [삭제](#) [목록](#)

위 그림은 write.php, write\_update.php, view.php 파일들을 모두 수정후 Web browser 화면이다.htdocs directory 내에 있는 이미지를 선택해 게시글을 작성했다. 위 화면을 통해 정상적으로 이미지가 업로드 되었음을 확인할 수 있다.

## Conclusion.

이번 프로젝트의 내용은 Apache, PHP, MySQL 을 모두 설치 후, 연동시켜 이를 활용해 게시판을 구축하는 것이었다. 게시판 기능 중 게시글 쓰기, 보기, 수정, 삭제, 글 조회 수 기능은 기본적으로 주어져 있어서 간단하게 해결할 수 있을 줄 알았지만 아니었다. Php 문에서 \$sql 에 sql 명령문을 삽입할 시 각 column 들을 큰따옴표나 작은따옴표로 구분해줘야 하는데 문제는 sql 명령문 전체도 큰따옴표나 작은 따옴표로 표시하는 부분에 있어 전자의 따옴표인지 후자의 따옴표인지 Editor 가 식별하지 못하는 문제가 있었다.

필자는 HTML, PHP 이나 MySQL 문법에 대한 배경 지식이 없었기 때문에 무엇이 문제인지조차 알지 못해 이 문제를 해결하는 것이 이 프로젝트에서 가장 힘들었던 일이었던 것 같다. 그래서 처음부터 기본적인 web 과 MySQL 기본 문법을 쌓는 것이 우선이라고 생각해서 인터넷 강의를 듣고 관련 서적을 읽으며 파악했다. 그 후 다시 처음부터 코드를 작성해보니 문제가 있는 곳을 발견할 수 있었다. 필자는 \$sql 에 전체 명령어를 삽입하는 부분을 큰따옴표로, 그 안에 column 이나 value 들을 삽입할 때는 작은따옴표로 구분해(강의 자료를 약간 수정) 이 문제를 해결할 수 있었다. 그 이후 MySQL command line 도 같이 사용하며 실제 생성한 table 과 column 들을 확인하며 진행했다.

이렇게 배경 지식을 먼저 쌓고 시작했기에 이미지를 업로드 하는 부분은 비교적 쉽게 수행할 수 있었다. \$sql 변수에 'b\_image' column 을 삽입하지 않았던 문제는 천천히 하나하나 읽어보면서 찾을 수 있었고, 이미지 파일의 경로를 제대로 설정하지 못하는 문제는 httdocs directory 에 이미지를 옮겨 해결할 수 있었다. Php는 비교적 오래된 기술이라는 평가가 있지만, 이번 프로젝트를 통해 Web, DataBase 두 분야에 기초적인 초석을 다질 수 있었던 좋은 기회였던 것 같다.

## Reference.

데이터베이스설계 / 김은경 / 한국기술교육대학교 출판부

<http://mcpaint.tistory.com/194>

[http://www.joshi.co.kr/index.php?mid=board\\_EudV58&document\\_srl=293259](http://www.joshi.co.kr/index.php?mid=board_EudV58&document_srl=293259)

Inflearn WEB3 – PHP & MySQL : <https://www.inflearn.com/course/web3-php-mysql/>