

# 간단한 로그인 기능이 포함된 PHP 페이지 작성

## 과제 설명

웹의 기초인 웹서버를 구축하고 PHP 코드를 활용하는 방법에 대해 학습하였습니다.

학습한 내용을 기반으로 GET 메소드를 요청하여 로그인 기능을 처리하는 로직을 구현합니다.

## 목차

### 1. 개발 환경 설정

- XAMPP 설치
- Apache와 MySQL 실행

### 2. 데이터베이스 설계

- 데이터베이스 생성
- 테이블 생성
- 사용자 정보 삽입

### 3. 로그인 페이지 구현

- Index.html
- login.php
- 로그인 페이지 구현 결과

### 4. 외부 접속 허용 설정

- 포트 포워딩
- 포트 포워딩 설정

### 5. 참고문헌

작성날짜 : 2024.09.04 수

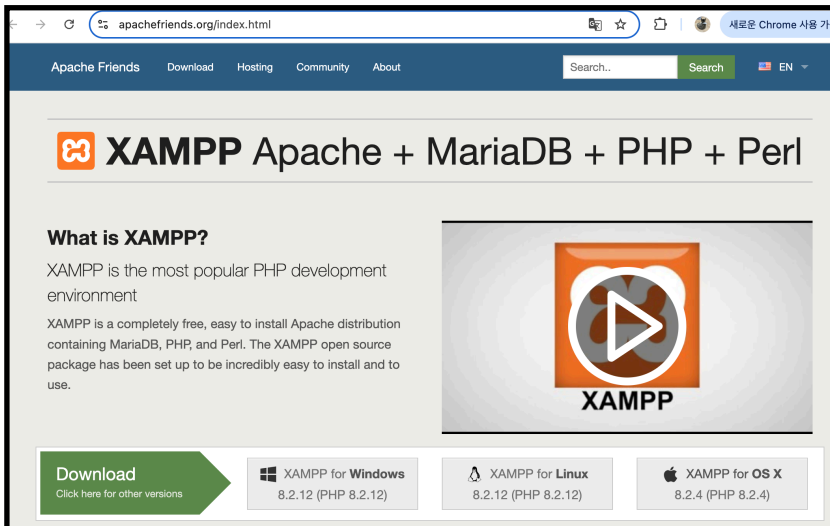
작성자 : 김정훈 국봉호 김민서 김연우

## 1. 개발 환경 설정

본 실습은 Windows 11 환경에서 진행되었습니다. 웹 서버를 구동하기 위해 Apache, PHP, MySQL을 실행하고 설정해야 합니다. XAMPP를 사용하면 이러한 환경을 쉽게 구축하고 관리할 수 있습니다.

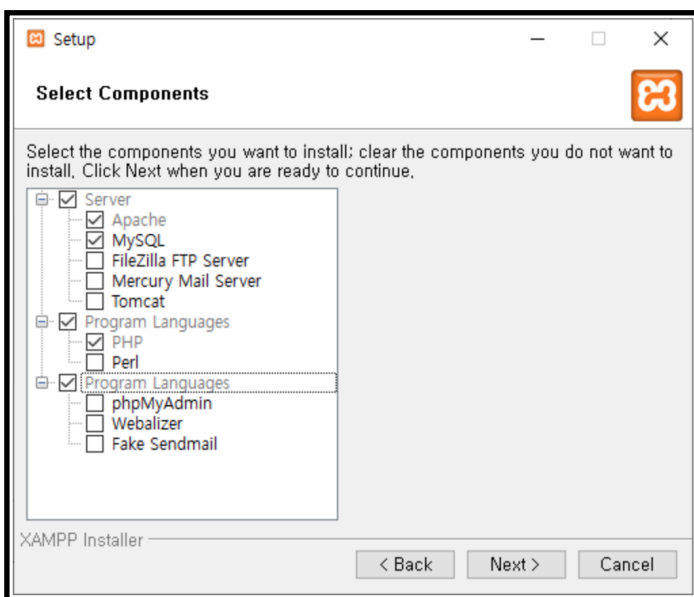
### 1-1) XAMPP 설치

XAMPP를 설치하기 위해 홈페이지([www.apachefriends.org/index.html](http://www.apachefriends.org/index.html))에 접속합니다.



그림[1] XAMPP 홈페이지

본인의 운영체제에 맞게 다운로드를 진행 후 설치 파일을 실행합니다.



그림[2] XAMPP Setup

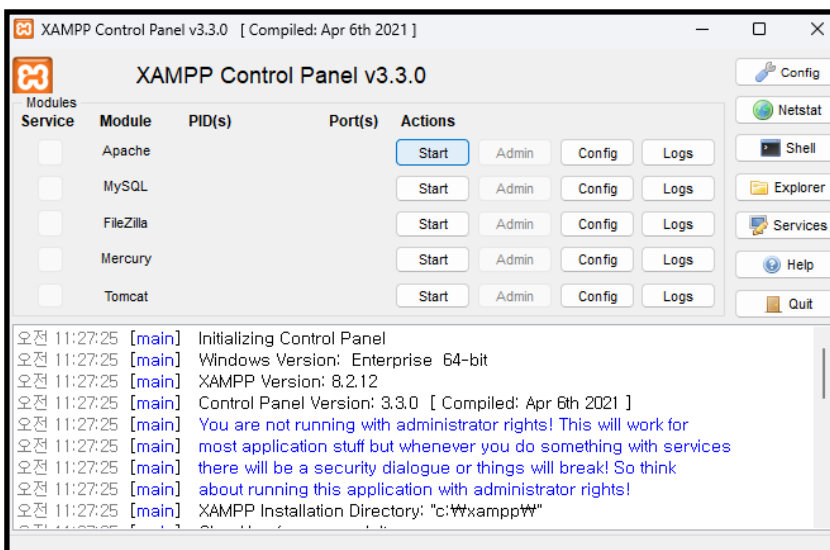
그림[2]와 같은 창이 나오면 필요한 프로그램을 선택합니다. 여기서는 기본 구성 요소인 Apache, MySQL, PHP만 선택하여 진행하였습니다.



그림[3] Apache 서버의 방화벽 경고

도중에 Apache 서버의 방화벽 경고가 뜨는데, 액세스 허용하여 설치를 마저 진행합니다. 설치가 완료되면 Finish를 눌러 설치마법사를 종료하고 XAMPP 컨트롤 패널을 실행합니다.

## 1-2) Apache와 MySQL 실행



그림[4] XAMPP 실행

XAMPP 실행하면 그림[4]와 같은 화면이 표시됩니다. 여기서 Apache와 MySQL을 “Start” 버튼을 눌러 시작합니다.

## 2. 데이터베이스 설계

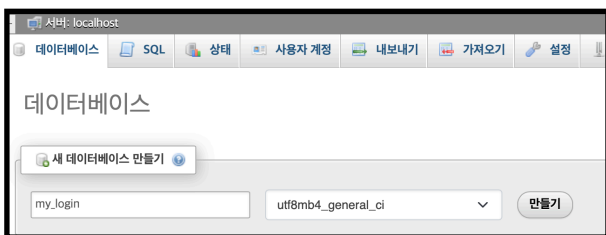
XAMPP로 Apache와 MySQL을 실행하여 웹 서버를 설정했으므로, 이제 로그인 기능을 구현하는 작업을 시작할 수 있습니다. 이를 위해 먼저 로그인 정보를 저장할 데이터베이스를 설계해야 하며, 이 과정에서 phpMyAdmin을 활용할 수 있습니다.

phpMyAdmin은 MySQL 데이터베이스를 웹 환경에서 관리하기 위해 PHP로 작성된 오픈 소스 도구입니다. 이 도구를 통해 데이터베이스, 테이블, 필드, 열을 생성, 수정, 삭제할 수 있으며, SQL 쿼리 실행이나 사용자 및 권한 관리를 포함한 다양한 작업을 수행할 수 있습니다.

phpMyAdmin은 특히 웹 호스팅 서비스를 위한 가장 널리 사용되는 MySQL 관리 도구 중 하나로 잘 알려져 있습니다.

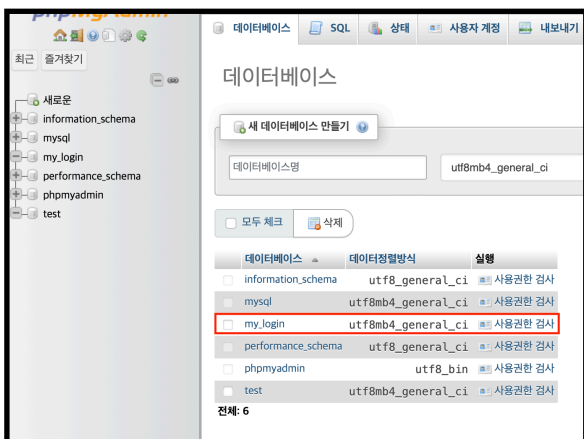
### 2-1) 데이터베이스 생성

주소창에 `http://localhost/phpmyadmin/index.php`를 입력하여 phpMyAdmin에 접속합니다. 데이터베이스를 생성하기 위해 먼저 데이터베이스 메뉴로 들어가서 새 데이터베이스 만들기에서 데이터베이스 이름을 입력하고 만들기를 눌러서 데이터베이스를 생성합니다.



그림[5] 데이터베이스 생성

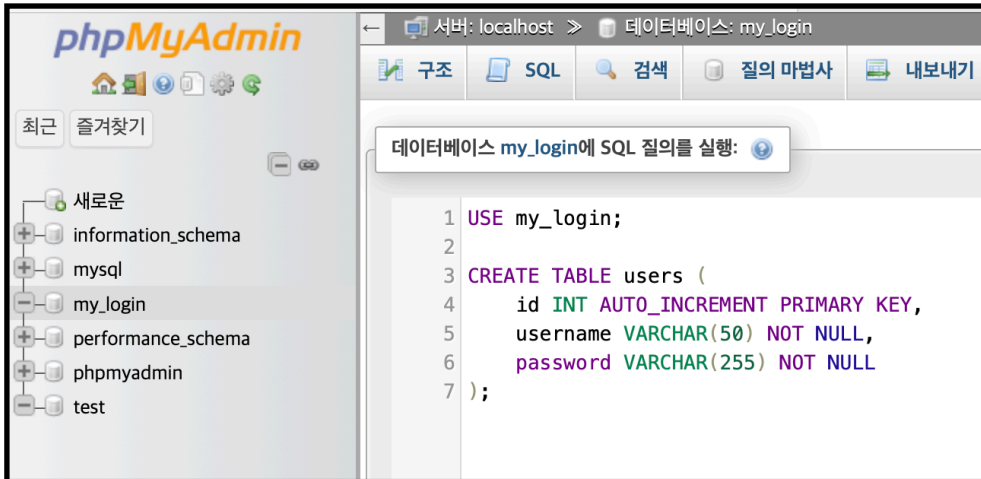
my\_login 명의 데이터베이스를 생성하였으며 그림[6]과 같이 성공적으로 생성된 걸 확인할 수 있습니다.



그림[6] 데이터베이스 생성 결과

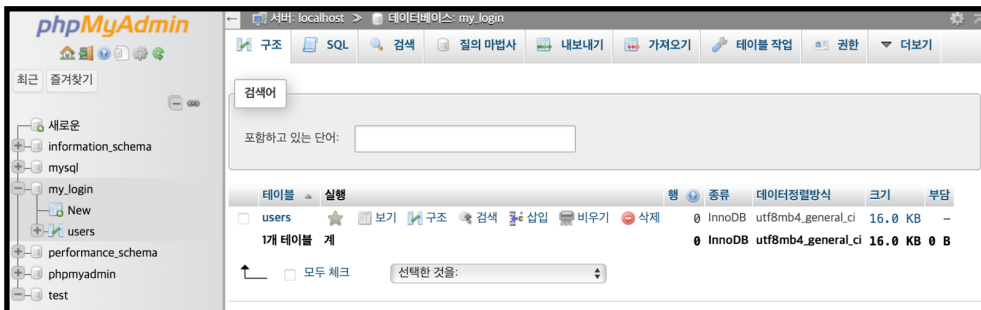
## 2-2) 테이블 생성

SQL 쿼리문의 CREATE TABLE users 구문을 활용하여 my\_login 데이터베이스에서 users 테이블을 생성합니다. AUTO\_INCREMENT를 설정하여 새로운 사용자가 추가될 때마다 id 값이 자동으로 1씩 증가하도록 하고 username과 password는 각각 50자, 255자를 한계로 두었습니다. 또한, NOT NULL 구문을 사용하여 값이 반드시 존재하도록 설정합니다.



그림[7] 테이블 생성 쿼리문 작성

그림[8]은 성공적으로 테이블을 설정한 화면입니다. my\_login 데이터베이스 내에 users 테이블이 생성된 것을 확인할 수 있습니다.

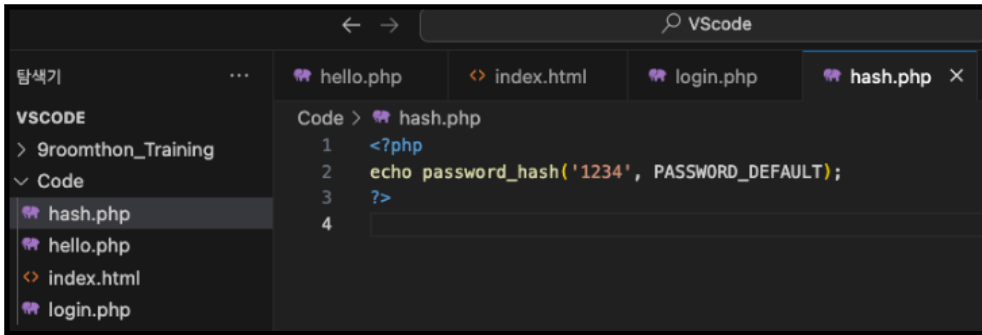


그림[8] 테이블 생성 성공

## 2-3) 사용자 정보 삽입

로그인 시스템에서는 보안을 위해 비밀번호를 암호화된 상태로 저장해야 합니다. 이를 위해 password\_hash() 함수를 사용하여 비밀번호를 안전하게 암호화합니다. 이후 사용자가 로그인할 때 입력한 비밀번호는 다시 암호화된 후, 데이터베이스에 저장된 암호화된 비밀번호와 password\_verify() 함수를 사용해 비교됩니다. 두 값이 같으면 로그인에 성공하고, 다르면 로그인에 실패하게 됩니다.

XAMPP의 Apache 서버는 XAMPP가 설치되어 있는 폴더에서 'htdocs' 폴더를 기준으로 실행됩니다. 즉, 웹페이지를 구성하는 html이나 php파일을 htdocs 폴더 내에 넣어야 한다는 의미입니다. 따라서 그림[9]과 같이 XAMPP의 htdocs 폴더에 hash.php 파일을 생성합니다.



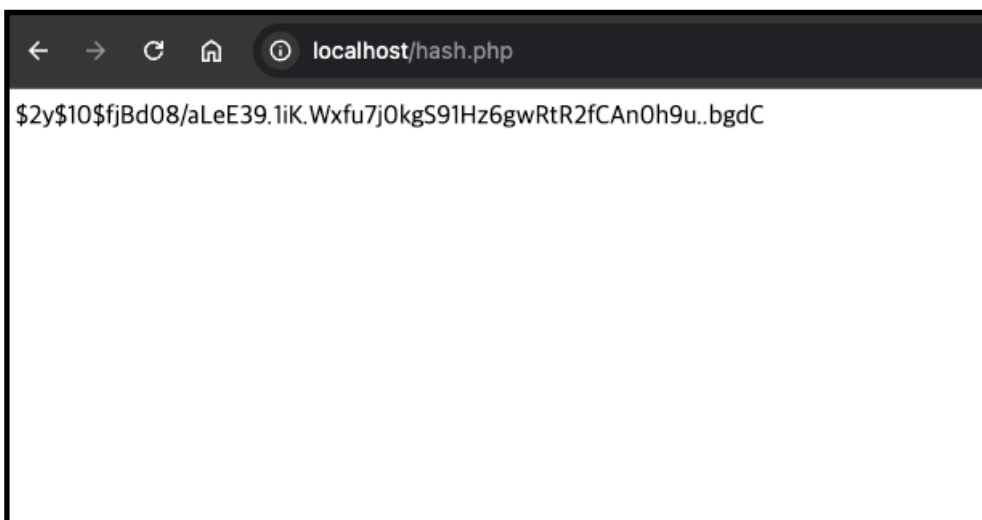
그림[9] htdocs 폴더에 hash.php 파일 생성

그런 다음 그림[10]과 같이 password\_hash() 함수에 사용자 비밀번호를 입력값으로 넣고 PHP 에서 제공하는 기본 암호화 알고리즘을 사용하여 비밀번호를 암호화하는 코드를 작성합니다.



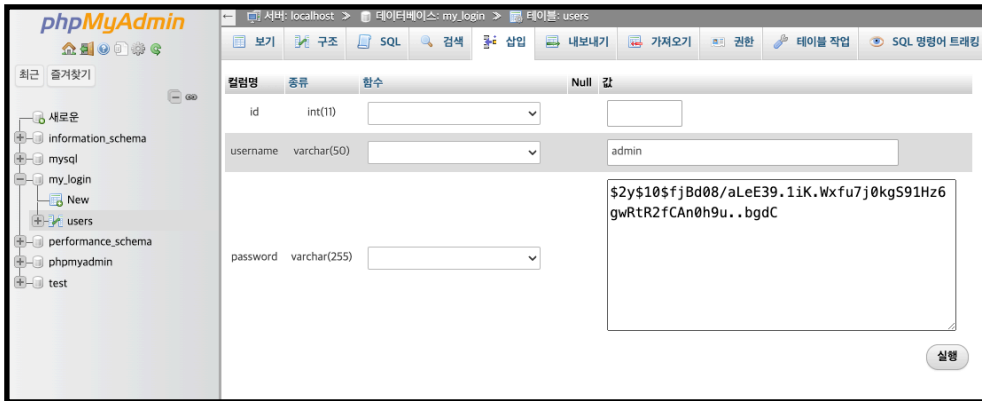
그림[10] 비밀번호 암호화 코드

웹 브라우저에서 http://localhost/hash.php 주소를 통해 해당 파일을 실행한 결과를 확인하여 데이터베이스에 저장할 수 있는 사용자 비밀번호의 암호화된 값을 획득합니다.



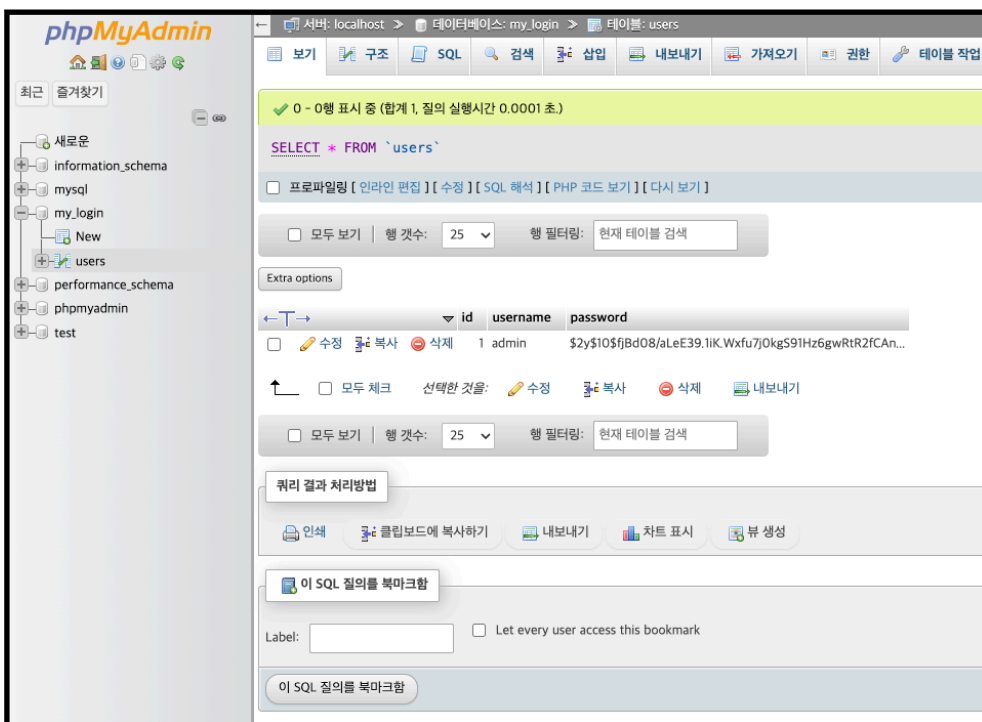
그림[11] 비밀번호 암호화 확인

my\_login 데이터베이스의 users 테이블에 사용자 이름과 암호화된 비밀번호를 삽입합니다.



그림[12] 사용자 정보 삽입

users 테이블에 첫 번째 사용자 정보가 성공적으로 삽입된 것을 확인할 수 있습니다. 첫 번째 레코드가 정상적으로 삽입된 것을 통해, 로그인 시스템에서 사용자를 조회할 준비가 완료되었음을 알 수 있습니다.



그림[13] 테이블에 사용자 정보 삽입 완료

### 3. 로그인 페이지 구현

XAMPP의 Apache 서버는 XAMPP가 설치되어 있는 폴더에서 'htdocs'폴더를 기준으로 실행됩니다. 즉, 웹페이지를 구성하는 html이나 php파일을 htdocs 폴더 내에 넣어야 한다는 의미입니다.

#### 3-1) index.html

사용자에게 로그인 폼을 제공하고 ID와 비밀번호를 입력받습니다. 정보는 POST 메소드를 통해 login.php로 전송합니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login Page</title>
</head>
<body>
  <h2>Login Form</h2>
  <form action="login.php" method="post">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required><br><br>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required><br><br>
    <input type="submit" value="Login">
  </form>
</body>
</html>
```

그림[14] index.html 코드

각 코드 라인별로 다음과 같습니다.

line1: HTML5 문서임을 명시

line2: lang 속성으로 문서의 기본 언어를 영어로 설정

line4: meta 태그로 문자 인코딩을 UTF-8로 설정

line5: title 태그로 제목 설정

line9: 데이터를 전송할 대상을 login.php로 설정, POST방식으로 데이터 전송

line11~14: input 필드로 username과 비밀번호를 입력받습니다. required 속성은 필수 입력임을 나타냅니다.



### 3-2) index.html

로그인 정보를 받아 데이터 베이스와 비교하여 인증을 수행합니다. 비밀번호는 해시된 상태로 저장되어 있고, 검증 함수를 통해 검증합니다. 인증 결과에 따라 성공 또는 실패 메시지를 출력합니다.

```
<?php
$servername = "localhost";
$dbusername = "root";
$dbpassword = "";
$dbname = "mydb";

// MySQL 데이터베이스 연결
$conn = new mysqli('localhost', 'root', 'password', 'my_login');

// 연결 확인
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// POST 메소드로 사용자 입력 가져오기
$username = $_POST['username'];
$password = $_POST['password'];

// 데이터베이스에서 사용자 정보 조회
$sql = "SELECT * FROM users WHERE username = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("s", $username);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    // 사용자가 존재할 경우
    $row = $result->fetch_assoc();
    if (password_verify($password, $row['password'])) {
        echo "<h1>Login Successful</h1>";
        echo "<p>Welcome, $username!</p>";
    } else {
        echo "<h1>Login Failed</h1>";
        echo "<p>Incorrect password</p>";
    }
} else {
    echo "<h1>Login Failed</h1>";
    echo "<p>User not found</p>";
}

$stmt->close();
$conn->close();
?>
```

그림[15] index.html 코드

각 코드 라인별로 다음과 같습니다.

line8: MySQL 데이터베이스에 연결

line11~13: 데이터베이스 연결이 성공적으로 이루어졌는지 확인

line16~17: HTML 폼에서 사용자명과 비밀번호를 전달 받습니다.

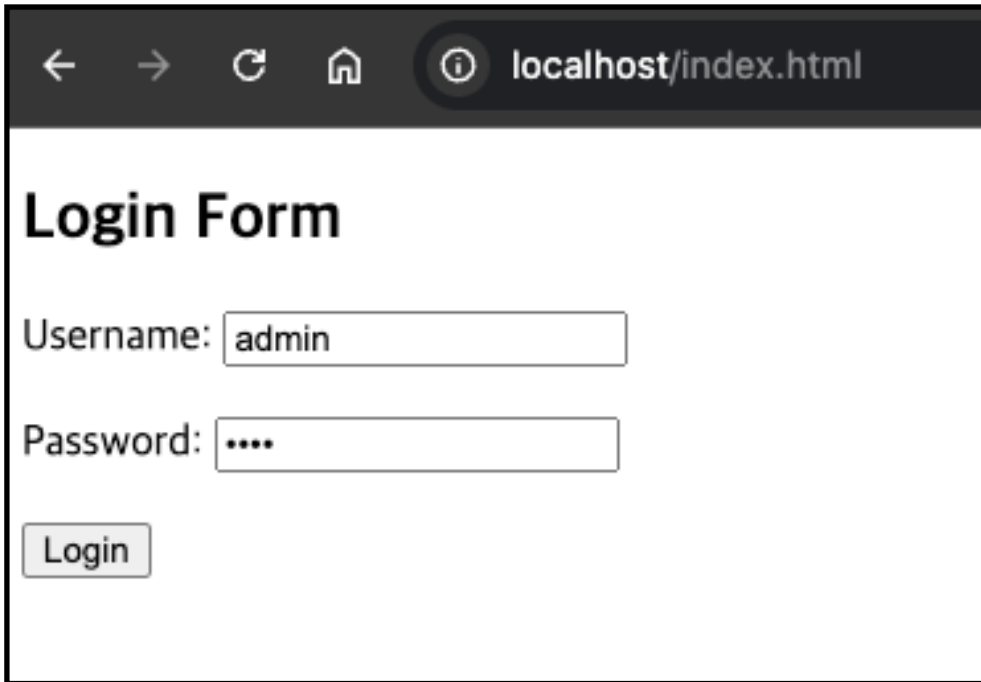
line20~24: 데이터에서 사용자를 조회합니다.

line26~39: 입력된 비밀번호가 데이터베이스에 있는 해시된 비밀번호와 일치하는지 확인합니다.

### 3-3) 로그인 페이지 구현 결과

다음은 로그인 기능 구현 후 성공 및 실패 결과 과정입니다. 사용자가 올바른 로그인을 시도할 경우, 그림[18]와 같이 로그인을 성공하였음을 확인할 수 있습니다.

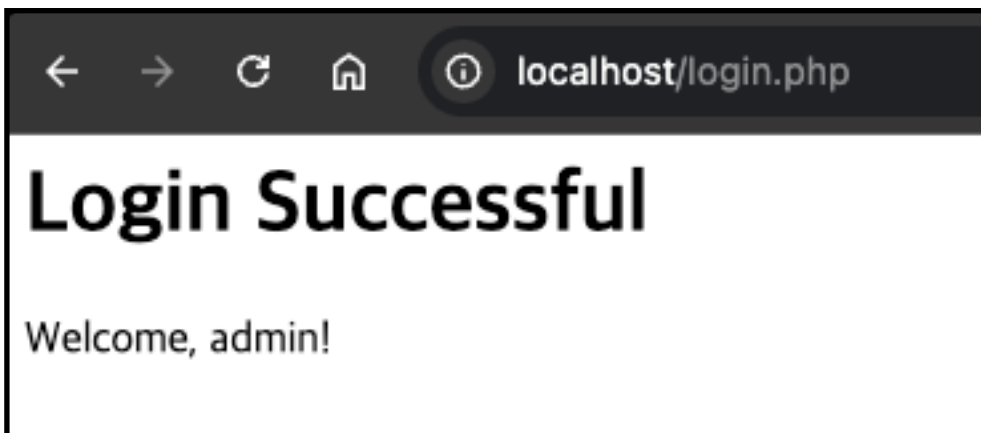
(Username : admin / Password : 1234)



The screenshot shows a web browser window with the address bar displaying 'localhost/index.html'. The main content area features a 'Login Form' with the following elements:

- A 'Username:' label followed by a text input field containing the value 'admin'.
- A 'Password:' label followed by a password input field containing four dots '....'.
- A 'Login' button located below the password field.

그림[16] 올바른 로그인 시도



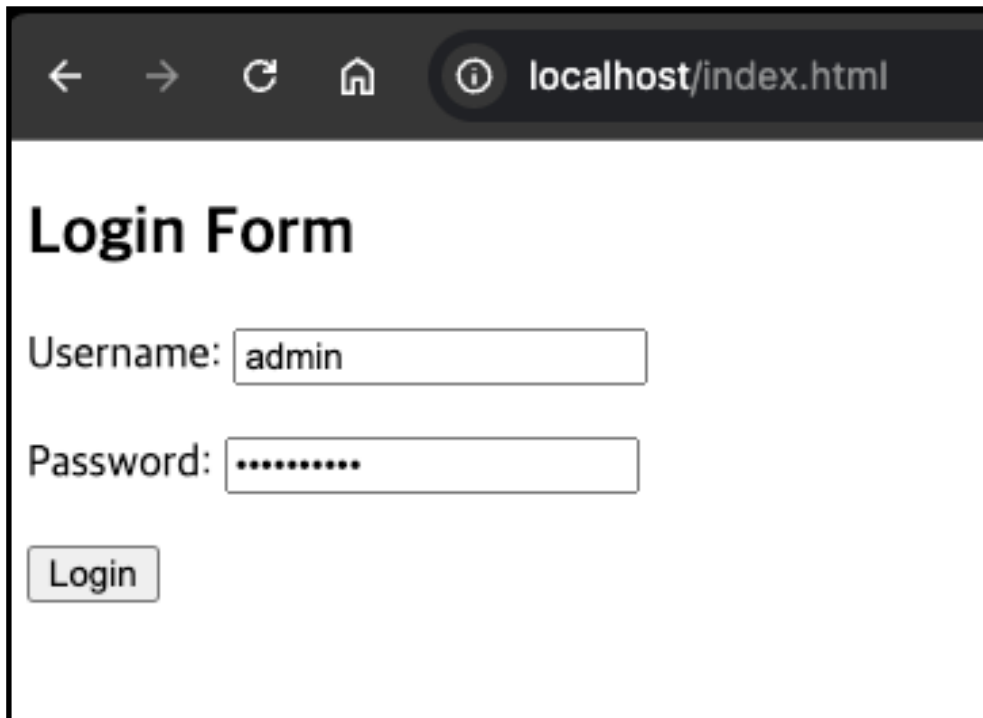
The screenshot shows a web browser window with the address bar displaying 'localhost/login.php'. The main content area displays a successful login message:

- A large heading 'Login Successful'.
- A line of text 'Welcome, admin!' below the heading.

그림[17] 로그인 성공

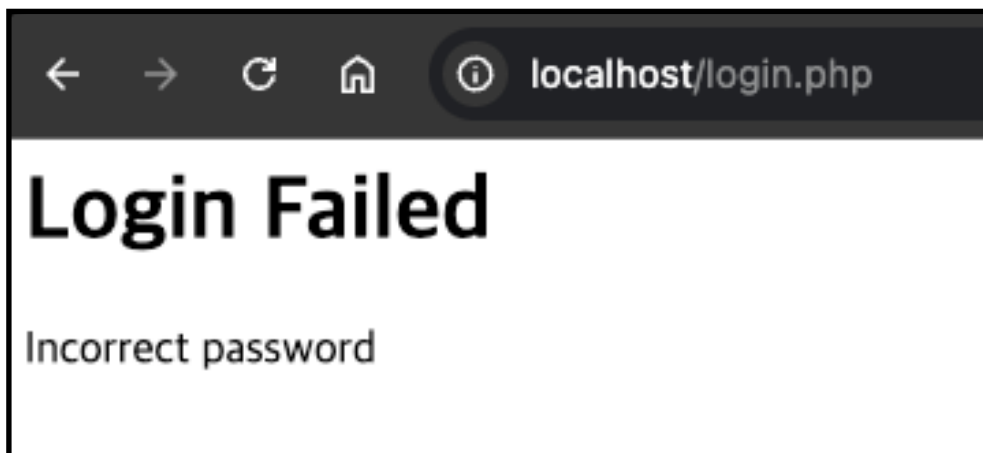
하지만 잘못된 로그인을 시도할 경우, 그림[20]와 같이 로그인을 실패하였음을 확인할 수 있습니다.

(Username : admin / Password : 1234를 제외한 다른 값)



The screenshot shows a web browser window with the address bar displaying 'localhost/index.html'. The page content is titled 'Login Form'. Below the title, there are two input fields: 'Username:' followed by a text box containing 'admin', and 'Password:' followed by a text box filled with dots. Below these fields is a button labeled 'Login'.

그림[18] 잘못된 로그인 시도



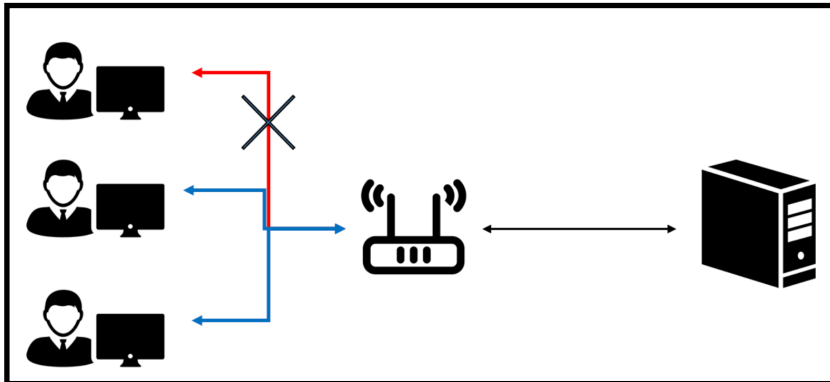
The screenshot shows a web browser window with the address bar displaying 'localhost/login.php'. The page content is titled 'Login Failed'. Below the title, the text 'Incorrect password' is displayed.

그림[19] 로그인 실패

## 4. 외부 접속 허용 설정

### 4-1) 포트 포워딩

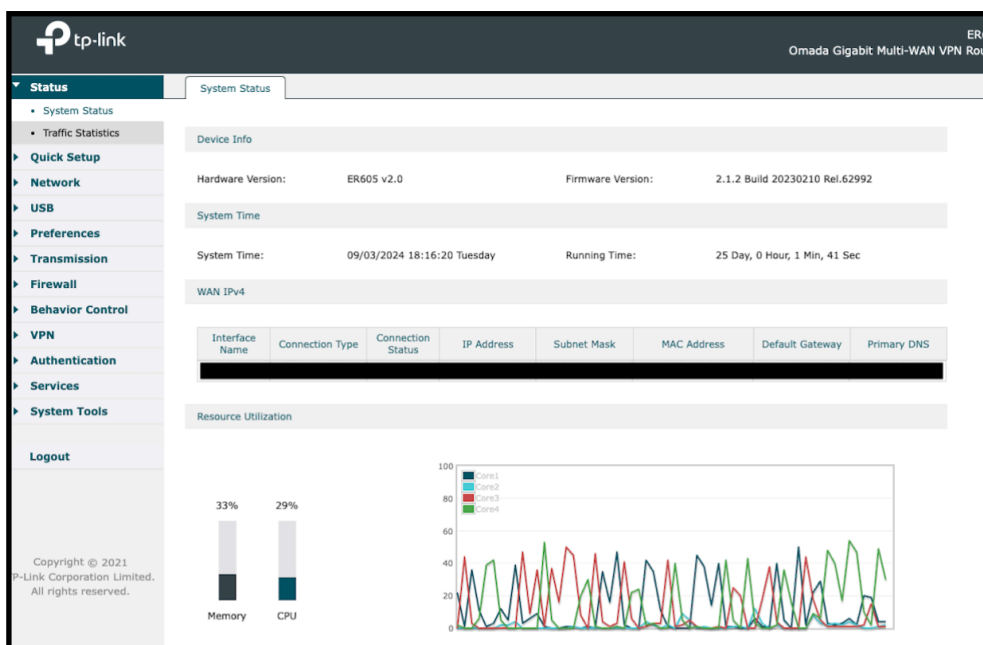
외부 기기가 내부에 있는 컴퓨터나 서버에 접속하려면, 먼저 공유기를 통과해야 합니다. 공유기는 방화벽 역할을 하기 때문에, 외부에서 내부 컴퓨터에 접근을 허용하려면 특정 포트를 열어줘야 합니다. 이러한 역할을 하는 것이 포트 포워딩(Port Forwarding)입니다. 포트 포워딩은 공유기의 포트를 통해 외부 네트워크에서 내부 네트워크의 특정 기기와 통신할 수 있게 해주는 기능입니다. 이를 통해 허가된 포트를 통해 접근한 외부 사용자만이 서버 PC에 접속할 수 있습니다.



그림[20] 외부접속->라우터->서버

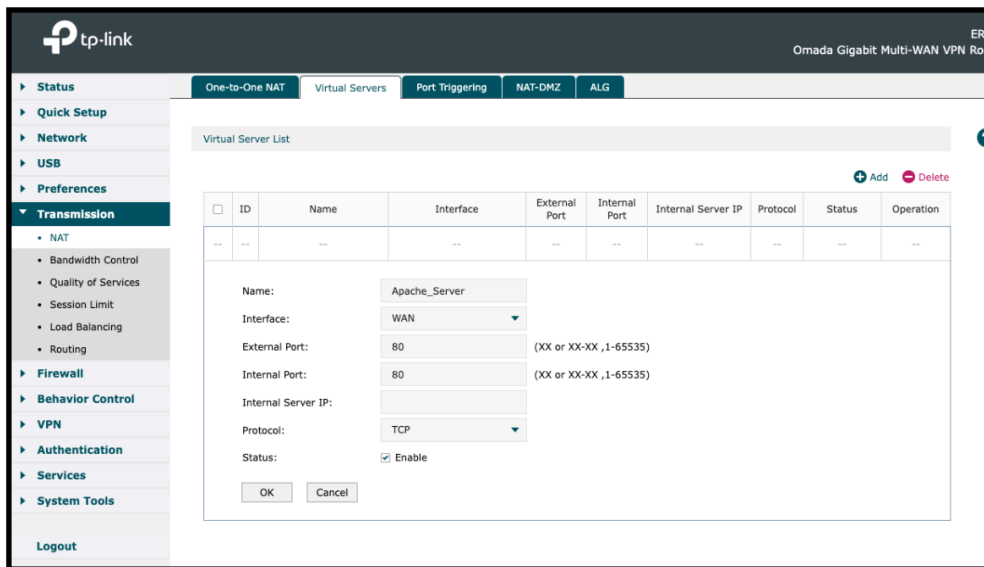
### 4-2) 포트 포워딩 설정

외부에서 접속을 허용하기 위해 공유기 관리 페이지에 접속합니다.

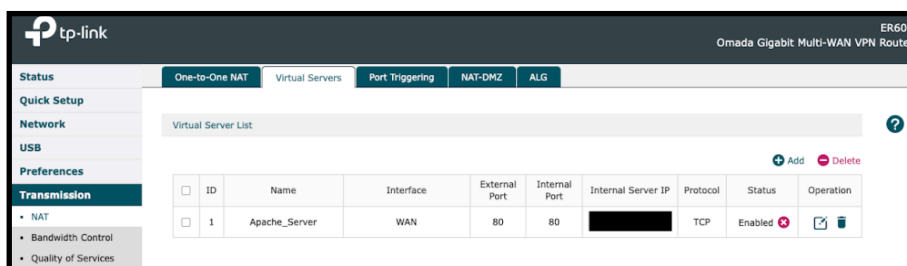


그림[21] 공유기 설정 페이지

Virtual Servers 설정 페이지에서 Apache Server 포트 설정합니다. 포트 설정이 완료되면 그림 [23]와 같은 화면이 나타납니다.

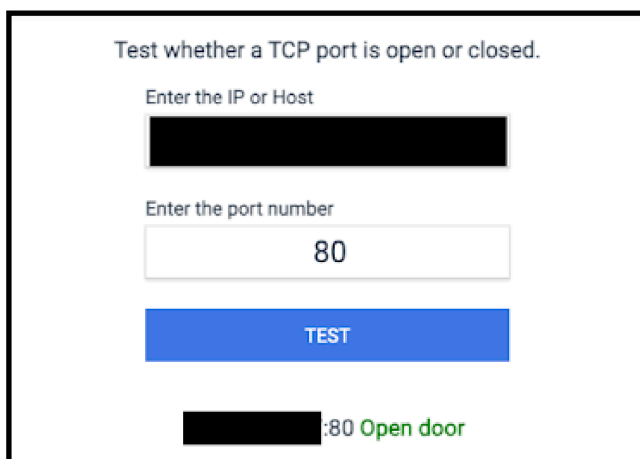


그림[22] Virtual Servers 설정 페이지를 통한 Apache Server 포트 설정



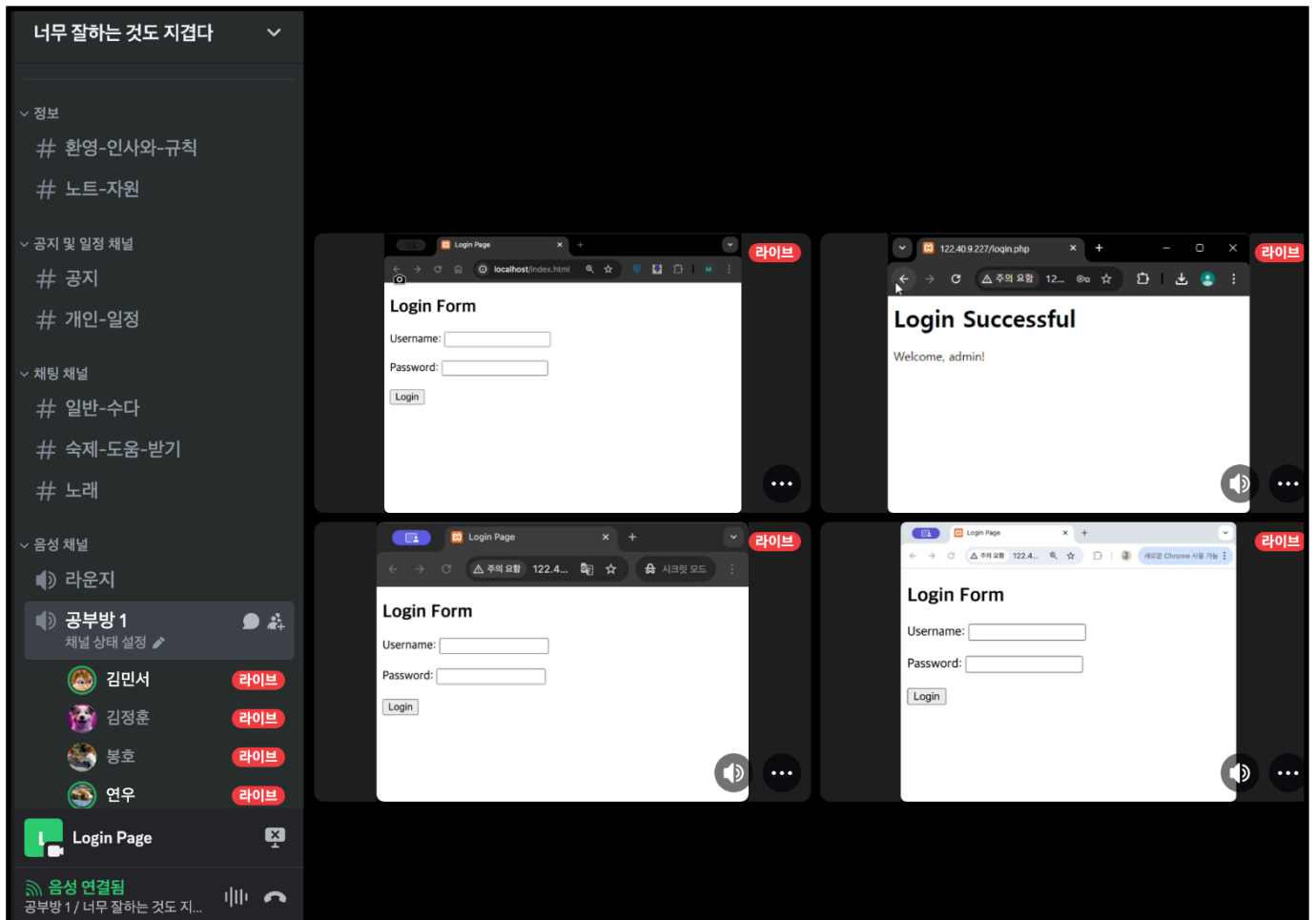
그림[23] Apache Server 포트 설정 완료

포트 포워딩이 되었는지 확인하기 위해 해당 링크(<https://yip.org/en/port-check>)에 접속해서 확인합니다. 차례대로 외부에서 접속을 위한 외부 IP, 포트 번호, 확인 결과를 알 수 있습니다.



그림[24] 포트 포워딩 확인 결과

이후 스터디원들과 함께 외부 접속이 가능한지 확인해보았습니다. 한 명은 본인의 localhost로 접속을 시도하고 다른 사람들은 해당 IP(122.XX.X.XXX)로 접속하여 외부 환경에서 내부로 접속되는 것을 확인할 수 있었습니다.



그림[25] 외부 환경에서 접속 결과

## 5. 참고 문헌

XAMPP 설치 및 사용 : <https://teserre.tistory.com/12>

로그인 페이지 작성 : <https://velog.io/@910/PHP-MySQL-로그인-페이지-만들기>

포트포워딩 : <https://panython.tistory.com/28>