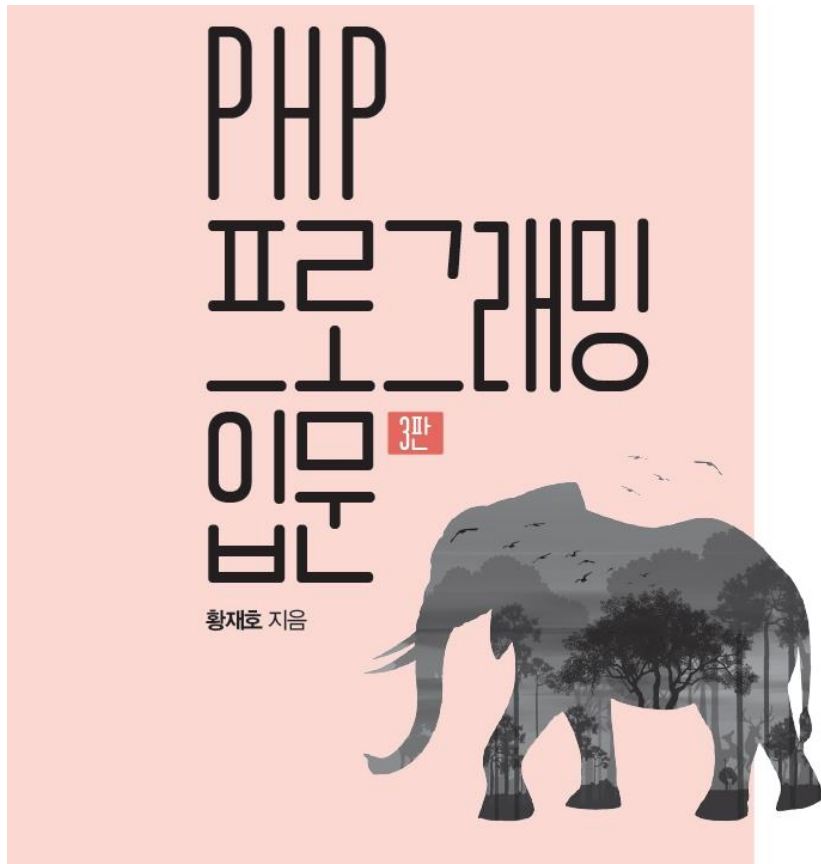


PHP 프로그래밍 입문 3판



[강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 처벌을 받을 수 있습니다.



Chapter 11 실습 사이트 소개와 회원 가입 페이지 제작

목차

01 실습 사이트 소개

02 회원 가입 페이지 제작

03 아이디 중복 확인

학습목표

- 실습 사이트의 구성을 이해할 수 있다.
- 메인 페이지의 구성을 이해할 수 있다.
- 회원 가입 페이지를 제작할 수 있다.
- 회원 정보 데이터베이스 테이블을 설계할 수 있다.
- 회원 가입 폼 양식에서 데이터를 전달할 수 있다.
- 회원 가입 정보를 데이터베이스에 저장할 수 있다.
- 아이디의 중복을 확인할 수 있다.

1. 사이트의 기능

- 실습 사이트의 메인 페이지.



그림 11-1 실습 사이트의 메인 페이지

- 회원 가입: 회원 가입 양식 페이지에 입력한 회원 정보 저장(11장)
- 로그인/로그아웃: 세션을 이용하여 로그인과 로그아웃 처리(12장)
- 회원 정보 수정: 기존의 회원 정보 수정(12장)
- 쪽지: 회원 상호 간의 메시지 송신과 수신(13장)
- 게시판: 게시판 글쓰기, 목록 보기, 내용 보기, 글 수정/삭제, 파일 첨부(14장)
- 최근 게시글: 게시판의 최근 게시글 가져오기(15장)
- 포인트: 메인 페이지에 포인트 랭킹 표시(15장)
- 관리자 모드: 회원 정보 삭제, 레벨 변경, 포인트 수정(15장)

2. 사이트의 구성

- 실습 사이트의 메인 페이지 구성. 메인 페이지는 상단 헤더, 메인 콘텐츠, 하단 푸터로 이루어져 있음.

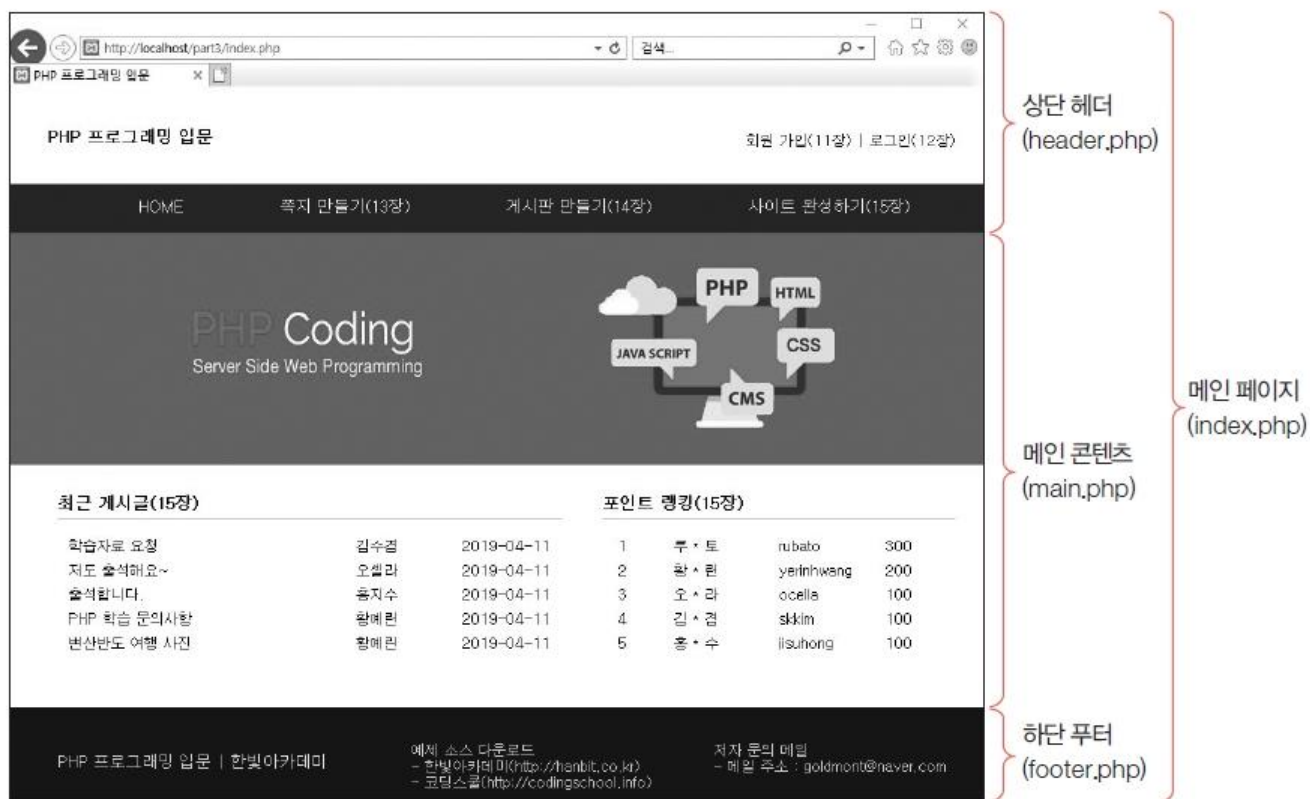


그림 11-2 메인 페이지의 구성

- **상단 헤더(header.php)** 사이트 로고, 상단 메뉴[회원 가입(11장), 로그인(12장)], 메인 메뉴[쪽지 만들기(13장), 게시판 만들기(14장), 사이트 완성하기(15장)]
- **메인 콘텐츠(main.php)** 메인 이미지, 최근 게시글, 포인트 랭킹
- **하단 푸터(footer.php)** 예제 소스 다운로드 안내, 메일 주소 안내

표 11-1 메인 페이지 구성에 필요한 파일과 폴더

구분	파일/폴더	설명
이미지	img	이미지 파일 저장 폴더
CSS	css/common.css	사이트에 공통으로 사용되는 CSS 파일
	css/main.css	메인 페이지의 CSS 파일
메인 페이지	index.php	메인 페이지
상단 헤더	header.php	메인 페이지의 상단 헤더
메인 콘텐츠	main.php	메인 페이지의 메인 콘텐츠
하단 푸터	footer.php	메인 페이지의 하단 푸터

- **img 폴더** 실습 사이트에서 사용되는 모든 이미지 파일을 모아놓은 폴더.
- **css 폴더** 2개의 CSS 파일이 있는데, common.css는 사이트에 공통으로 사용되는 CSS를 정의한 것이고 main.css는 메인 페이지에 사용되는 CSS를 별도로 정의한 것.

[예제 11-1] 메인 페이지 `part3/index.php`

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

①

```
<title>PHP 프로그래밍 입문</title>
```

```
<link rel="stylesheet" type="text/css" href="./css/common.css">
```

②

```
<link rel="stylesheet" type="text/css" href="./css/main.css">
```

```
</head>
```

```
<body>
```

[예제 11-1] 메인 페이지 `part3/index.php`

```
<header>
```

```
    <?php include "header.php";?>
```

③

```
</header>
```

```
<section>
```

```
    <?php include "main.php";?>
```

④

```
</section>
```

```
<footer>
```

```
    <?php include "footer.php";?>
```

⑤

```
</footer>
```

```
</body>
```

```
</html>
```

[예제 11-1] 메인 페이지 `part3/index.php`

- ① <meta> 태그를 이용하여 한글 문자셋을 UTF-8로 설정.
- ② **CSS 파일 불러오기** 사이트에 공통으로 사용되는 CSS인 common.css와 메인 페이지의 CSS인 main.css를 불러옴.
- ③ **상단 헤더 파일 불러오기** 메인 페이지의 상단을 구성하는 상단 헤더(header.php)를 불러옴. include는 파일을 현재 위치에 삽입하라는 의미.

```
include "경로/파일명";
```

- ④ **메인 콘텐츠 파일 불러오기** 메인 페이지 중앙에 있는 메인 콘텐츠(main.php)를 불러옴. 게시판의 최근 게시글을 불러오고 포인트 랭킹을 표시.
- ⑤ **하단 푸터 파일 불러오기** 메인 페이지 하단을 구성하는 하단 푸터(footer.php)를 불러옴

3. 실습 준비

3.1 실습 프로그램 복사하기

- 1장에서 내려받은 예제 소스 중 part3 폴더를 C:\xampp\htdocs에 복사.
- 이 폴더에는 3부의 실습을 진행하는 데 필요한 프로그램이 들어 있음.

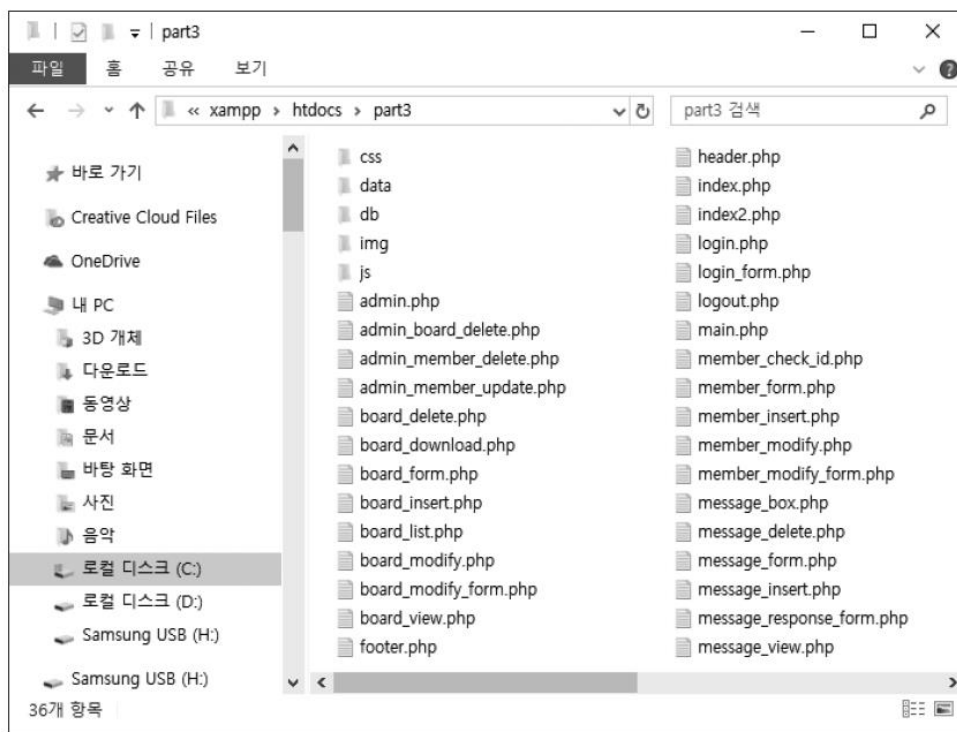


그림 11-3 C:\xampp\htdocs\part3 폴더의 구성

3.2 Apache와 MySQL 서버 프로그램 구동하기

- XAMPP 컨트롤 패널에서 Apache와 MySQL 서버 프로그램을 구동.

[TIP] XAMPP 컨트롤 패널에서 Apache와 MySQL 프로그램이 초록색 박스로 표시되면 정상 구동 중이라는 의미(9장의 [그림 9-5] 참고).

NOTE 예제 소스 내려받기

앞에서 이미 언급했듯이 실습에 필요한 예제 소스는 다음 홈페이지에서 내려받을 수 있다.

- 한빛아카데미 홈페이지: <http://www.hanbit.co.kr/src/4444>
- 코딩스쿨 홈페이지: <http://codingschool.info> → 자료실

3.3 웹 페이지 접속하기

- 실습 사이트의 메인 페이지에 접속하기 위해 웹 브라우저 주소 창에 다음 주소를 입력.
`http://localhost/part3/index.php`



그림 11-4 실습 사이트의 메인 페이지(index.php)

1. 회원 DB 테이블 설계

- 실습 사이트에서 회원 정보를 저장할 DB 테이블을 설계.
- 회원 DB 테이블의 num, id, pass, name 필드는 필수 입력 항목이므로 not null로 설정. 이때 num은 주 키와 auto_increment로 설정.

표 11-2 회원 DB 테이블(테이블명: members)

필드명	데이터형	null	추가 사항	설명
num	int	not null	primary key, auto_increment	일련번호
id	char(15)	not null		아이디
pass	char(15)	not null		비밀번호
name	char(10)	not null		이름
email	char(80)			이메일 주소
regist_day	char(20)			가입일
level	int			회원의 레벨
point	int			회원의 포인트

2. 회원 DB 테이블 생성

- 10장에서 설명한 SQL 명령 일괄 실행 방법으로 회원 DB 테이블을 생성.
- 회원 DB 테이블인 members 테이블을 생성하는 명령이 저장된 파일은 다음과 같음.

회원 DB 테이블(members) 생성 파일(part3/db/members.sql)

```
create table members (  
    num int not null auto_increment,  
    id char(15) not null,  
    pass char(15) not null,  
    name char(10) not null,  
    email char(80),  
    regist_day char(20),  
    level int,  
    point int,  
    primary key(num)  
);
```

- 데이터베이스 작업을 위해 웹 브라우저 주소 창에 다음을 입력하여 XAMPP 메인 화면을 확인

http://localhost

- phpMyAdmin에 접속하기 위해 XAMPP 메인 화면 오른쪽 상단의 [phpMyAdmin] 메뉴를 클릭.

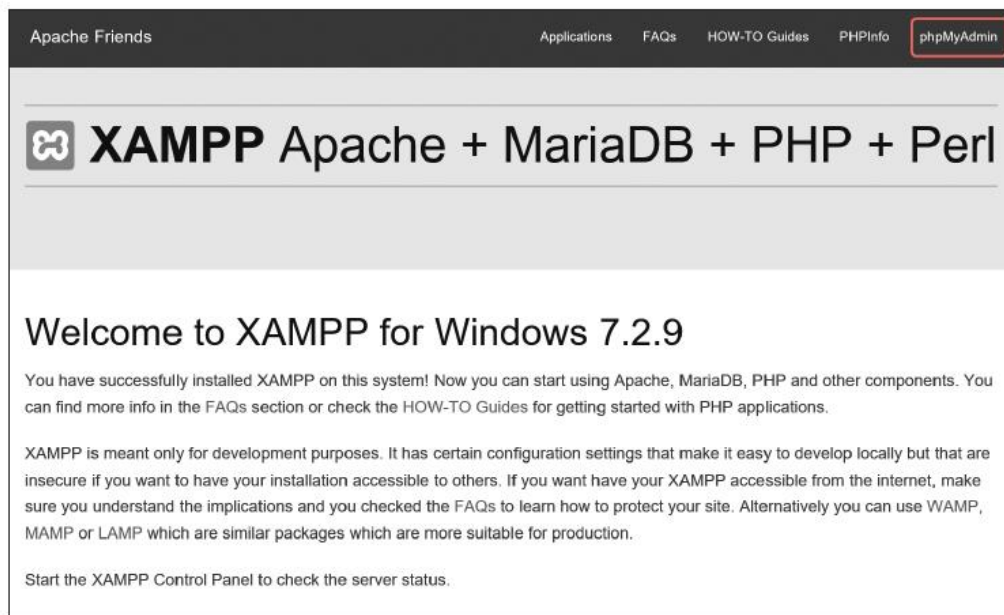


그림 11-6 XAMPP 메인 화면

- phpMyAdmin 로그인 화면이 나타남.
- 앞에서 생성한 user1 계정으로 접속하기 위해 phpMyAdmin 로그인 화면에 다음 정보를 입력.
 - ✓ 사용자명: user1
 - ✓ 암호: 12345



phpMyAdmin

phpMyAdmin에 오셨습니다

언어 - *Language*

한국어 - Korean

로그인

사용자명:

암호:

실행

그림 11-7 phpMyAdmin 로그인 화면

- 다음과 같은 화면이 나타나면 왼쪽 목록에서 sample을 클릭. 실습에서 사용할 데이터베이스로 sample 데이터베이스를 선택한 것.

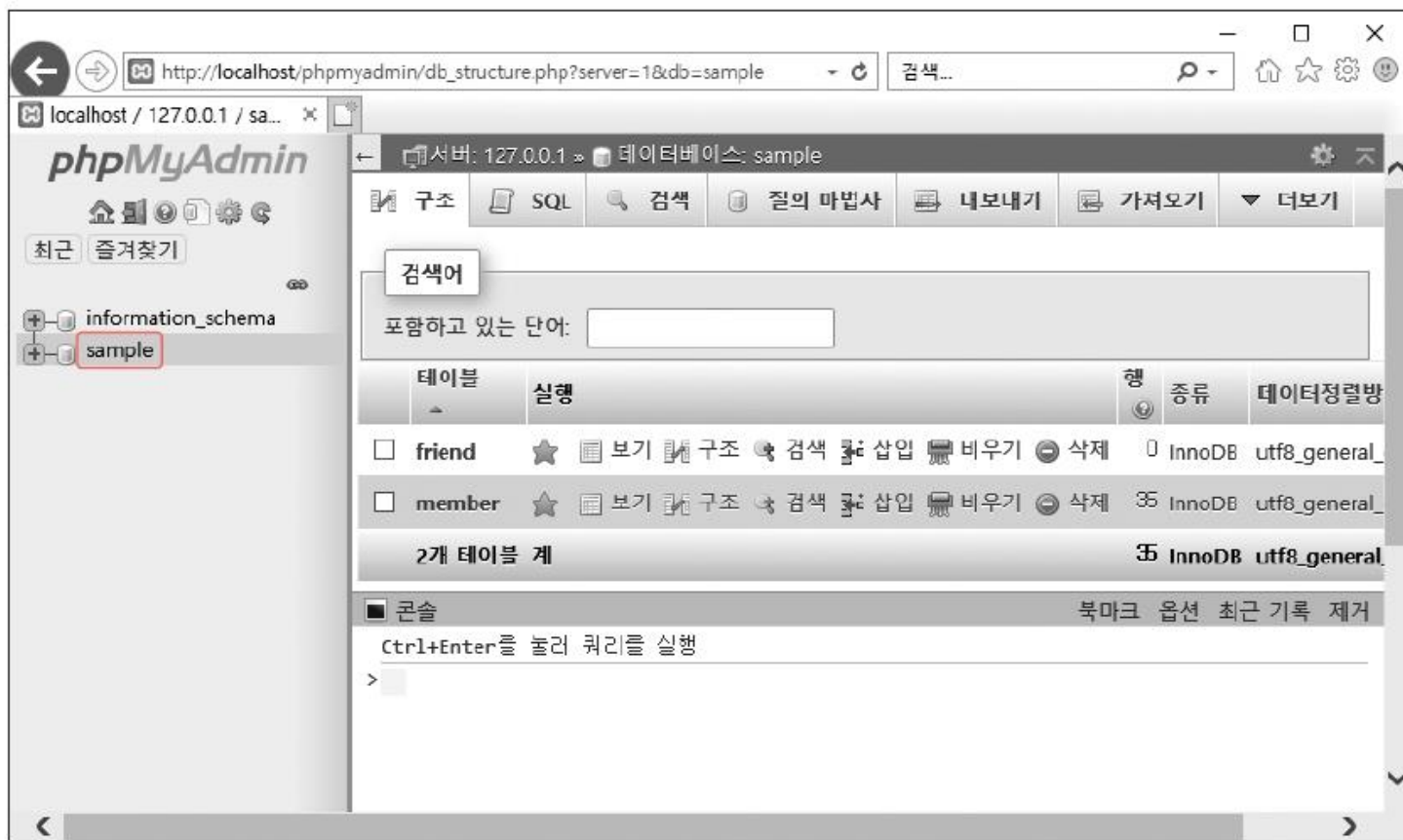


그림 11-8 sample 데이터베이스 선택

- 회원 DB 테이블을 생성하기 위해 만든 members.sql 파일을 DB로 가져와야 함.
- 화면 오른쪽 상단의 [가져오기] 메뉴를 클릭한 뒤 중앙의 <찾아보기>를 클릭하여 C:\xampp\htdocs\part3\db\members.sql을 선택. 하단의 <실행>을 클릭.

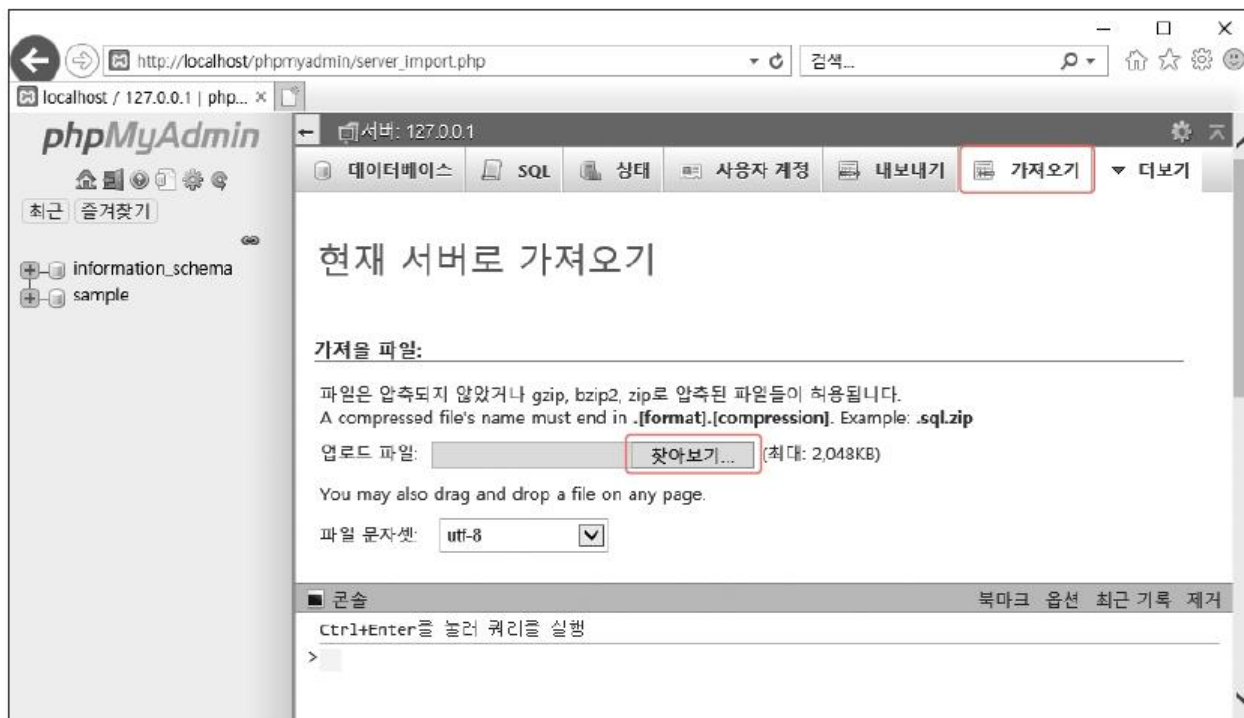


그림 11-9 members.sql 파일 가져오기

- 화면 왼쪽 목록에서 members 테이블을 선택한 뒤 상단의 [구조] 메뉴를 클릭하여 테이블의 구조를 살펴보면 앞에서 설계 및 생성한 것과 내용이 일치함을 확인할 수 있음.

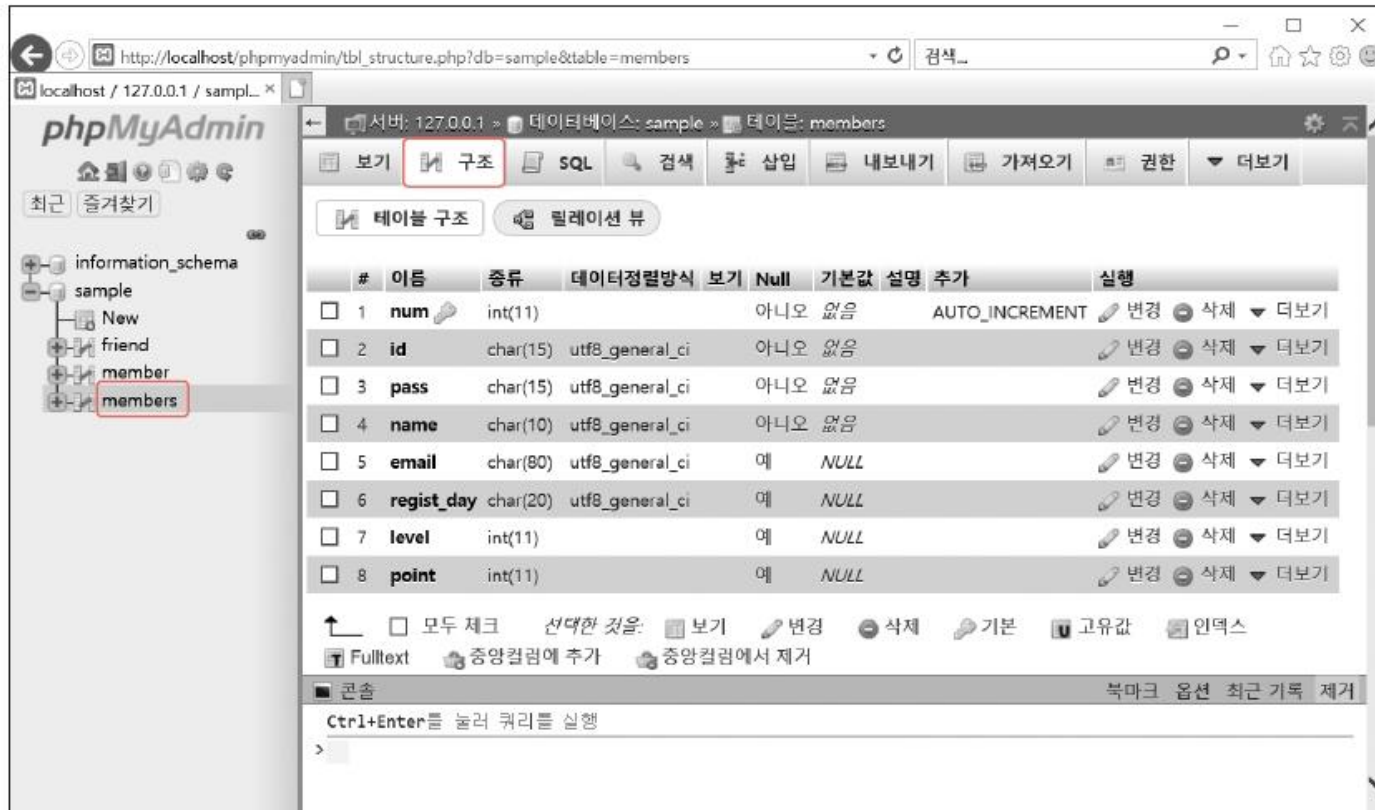


그림 11-10 members 테이블의 구조

3. 회원 가입 페이지 작성

- 웹 브라우저 주소 창에 다음 주소를 입력하면 회원 가입 페이지를 볼 수 있음.

`http://localhost/part3/member_form.php`

- 또는 메인 페이지(index.php) 오른쪽 상단의 [회원 가입(11장)] 메뉴를 클릭해도 됨.



The screenshot shows a web browser displaying the 'PHP Coding' website. The page has a dark header with navigation links: 'HOME', '쪽지 만들기(13장)', '게시판 만들기(14장)', and '사이트 완성하기(15장)'. In the top right corner, there are two links: '회원 가입(11장)' (highlighted with a red box) and '로그인(12장)'. The main content area features a large banner with the text 'PHP Coding Server Side Web Programming' and a diagram showing the relationship between PHP, HTML, CSS, JavaScript, and CMS. Below the banner is a '회원 가입' (Member Registration) form. The form includes input fields for '아이디' (ID), '비밀번호' (Password), '비밀번호 확인' (Confirm Password), '이름' (Name), and '이메일' (Email). There is a '중복확인' (Check Duplicate) button next to the ID field. At the bottom of the form, there are two buttons: '저장하기' (Save) and '취소하기' (Cancel).

그림 11-11 회원 가입 페이지(member_form.php)

표 11-3 회원 가입 페이지 구성에 필요한 파일

구분	파일	설명
DB	db/members.sql	회원 DB 테이블 생성
CSS	css/common.css	사이트에 공통으로 사용되는 CSS 파일
	css/member.css	회원 가입 CSS 파일
자바스크립트	js/member.js	회원 가입 페이지의 자바스크립트 파일
회원 가입 페이지	member_form.php	회원 가입 페이지
기능 페이지	member_insert.php	가입 정보를 DB에 저장
	member_check_id.php	아이디 중복 확인

- member.css는 회원 가입 페이지에서 사용되는 CSS 파일.
- member.js는 회원 가입 시 입력 창에 데이터가 제대로 입력되었는지 확인.
- member_insert.php는 사용자가 입력한 회원 가입 정보를 DB에 저장.
- member_check_id.php는 사용하려는 아이디가 기존 아이디와 중복되는지 검사.

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

```
<!-- 생략 -->
<link rel="stylesheet" type="text/css" href="/css/common.css">
<link rel="stylesheet" type="text/css" href="/css/member.css">
<script>
    function check_input()
    {
        if (!document.member_form.id.value) {
            alert("아이디를 입력하세요!");
            document.member_form.id.focus();
            return;
        }
        if (!document.member_form.pass.value) {
            alert("비밀번호를 입력하세요!");
            document.member_form.pass.focus();
            return;
        }
    }
}
```

①

②

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

```
if (!document.member_form.pass_confirm.value) {  
    alert("비밀번호 확인을 입력하세요!");  
    document.member_form.pass_confirm.focus();  
    return;  
}  
  
if (!document.member_form.name.value) {  
    alert("이름을 입력하세요!");  
    document.member_form.name.focus();  
    return;  
}  
  
if (!document.member_form.email1.value) {  
    alert("이메일 주소를 입력하세요!");  
    document.member_form.email1.focus();  
    return;  
}
```

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

```
if (!document.member_form.email2.value) {  
    alert("이메일 주소를 입력하세요!");  
    document.member_form.email2.focus();  
    return;  
}
```

```
if (document.member_form.pass.value !=  
    document.member_form.pass_confirm.value) {  
    alert("비밀번호가 일치하지 않습니다.₩n다시 입력해 주세요!");  
    document.member_form.pass.focus();  
    document.member_form.pass.select();  
    return;  
}
```

③

```
document.member_form.submit();  
}
```

④

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

```
function reset_form() {  
    document.member_form.id.value = "";  
    document.member_form.pass.value = "";  
    document.member_form.pass_confirm.value = "";  
    document.member_form.name.value = "";  
    document.member_form.email1.value = "";  
    document.member_form.email2.value = "";  
    document.member_form.id.focus();  
    return;  
}
```

⑤

```
function check_id() {  
    window.open("member_check_id.php?id=" + document.member_form.  
        id.value, "IDcheck",  
        "left=700,top=300,width=350,height=200,scrollbars=no,  
        resizable=yes");  
}
```

⑥

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

```
</script>
<!-- 생략 -->
<div id="join_box">
  <form name="member_form" method="post" action="member_insert.php">
    <h2>회원 가입</h2>
    <div class="form id">
      <div class="col1">아이디</div>
      <div class="col2">
        <input type="text" name="id">
      </div>
      <div class="col3">
        <a href="#"></a>
      </div>
    </div>
    <div class="clear"></div>
```

⑦

⑧

⑨

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

```
<div class="form">  
  <div class="col1">비밀번호</div>  
  <div class="col2">  
    <input type="password" name="pass">  
  </div>
```

⑩

```
</div>
```

```
<div class="clear"></div>
```

```
<div class="form">
```

```
  <div class="col1">비밀번호 확인</div>
```

```
  <div class="col2">
```

```
    <input type="password" name="pass_confirm">
```

⑪

```
  </div>
```

```
</div>
```

```
<div class="clear"></div>
```

```
<div class="form">
```

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

```
<div class="col1">이름</div>
<div class="col2">
    <input type="text" name="name">
</div>
</div>
<div class="clear"></div>
<div class="form email">
    <div class="col1">이메일</div>
    <div class="col2">
        <input type="text" name="email1">@<input type="text"
        name="email2">
    </div>
</div>
</div>
```

⑫

⑬

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

```
<div class="clear"> </div>
<div class="bottom_line"> </div>
<div class="buttons">
    <img style="cursor:pointer" onclick="check_input()" src=           ⑭
        "./img/button_save.gif"> &nbsp;
    <img id="reset_button" style="cursor:pointer" src=               ⑮
        "./img/button_reset.gif" onclick="reset_form()">
</div>
</form>
</div> <!-- join_box -->
<!-- 생략 -->
```


[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

① `check_input()` 함수 정의

- 자바스크립트 함수인 `check_input()`을 정의.
- `check_input()` 함수는 ⑭의 <저장하기> 버튼(`button_save.gif`)을 클릭하면 동작하는 이벤트 핸들러인 `onclick="check_input()"`에 의해 호출.

② 폼 양식의 데이터 검사

- 회원 가입 페이지의 입력 창(아이디, 비밀번호, 비밀번호 확인, 이름, 이메일)에 데이터가 있는지 검사.
- `document.member_form.id.value`는 회원 가입 페이지에 사용자가 입력한 데이터.
- if문 조건식인 `!document.member_form.id.value`는 아이디 입력 창에 데이터를 입력하지 않으면 참이 되어 `alert("아이디를 입력하세요!");`에 의해 화면에 경고 창 나타남.
- 아이디 입력 창의 데이터를 검사하는 `document.member_form.id.value` 각 요소의 의미.
 - ✓ `document`: 현재 웹 페이지의 문서 객체인 `join_form.php` 파일 자체
 - ✓ `member_form`: ⑦의 <form> 태그 `name` 속성 값인 `member_form`
 - ✓ `id`: ⑧에서 명시한 아이디 입력 창의 `name` 속성 값인 `id`

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

③ 입력 창의 데이터 일치 검사

- 비밀번호와 비밀번호 확인 입력 창에 입력된 데이터가 같은지 비교.
- 데이터가 서로 다르면 '비밀번호가 일치하지 않습니다.₩n다시 입력해 주세요!' 라는 메시지의 경고 창을 띄움.
- ₩n은 줄 바꿈을 나타내는 코드.

④ 폼 양식의 데이터 전달

- 사용자가 입력한 데이터를 POST 방식으로 전달.
- ⑦의 action 속성에 설정된 `member_insert.php`에 전달.

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

⑤ 입력 데이터 초기화 함수

- 회원 가입 페이지 하단의 <취소하기> 버튼을 클릭하면 입력 창에 입력되어 있는 데이터를 초기화(reset)하여 사용자가 다시 데이터를 입력할 수 있게 하는 `reset_form()` 함수 정의.
- `reset_form()` 함수는 ⑮의 <취소하기> 버튼(`button_reset.gif`)을 클릭하면 이벤트 핸들러인 `onclick="reset_form()"`에 의해 호출됨.

⑥ `check_id()` 함수 정의

- 회원 가입 페이지에서 아이디 입력 창 옆의 <중복확인> 버튼을 클릭하면 입력된 아이디가 중복되는지 검사.

⑦ <form> 태그와 action 속성 설정

- 사용자가 회원 가입 데이터를 입력한 뒤 하단에 있는 <저장하기> 버튼(`button_save.gif`)을 클릭하면 자바스크립트의 `onclick` 이벤트 핸들러에 설정된 ①의 `check_input()` 함수가 호출됨.
- `check_input()` 함수는 입력 창에 데이터를 모두 입력했는지 검사하고, 이상이 없으면 ④의 `document.member_form.submit()`에 의해 ⑦의 action 속성에 설정된 `member_insert.php`로 페이지가 이동.

[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

⑧ 아이디 입력 창의 name 속성 설정

- 사용자가 입력한 아이디는 name 속성으로 설정된 id에 의해 POST 방식인 `$_POST["id"]`로 ⑦의 `member_insert.php`에 전달됨.

⑨ 중복확인 버튼

- 아이디 입력 창 옆의 <중복확인> 버튼을 클릭하면 onclick 이벤트 핸들러에 설정된 `check_id()` 함수가 호출됨.

⑩ ~ ⑬ 비밀번호, 비밀번호 확인, 이름, 이메일 입력 창의 name 속성 설정

- ⑧과 같은 방식으로 비밀번호, 이름, 이메일의 name 속성에 각각 설정된 `pass`, `name`, `email1`, `email2`가 `$_POST["pass"]`, `$_POST["name"]`, `$_POST["email1"]`, `$_POST["email2"]`의 변수가 되어 `member_insert.php`로 전달됨.
- ⑪에서 비밀번호 확인 입력 창에 설정된 name 속성 값인 `pass_confirm`은 `member_insert.php`로 전달되지 않고 ①의 `check_input()` 함수에서 사용자 비밀번호를 재확인할 때 사용.

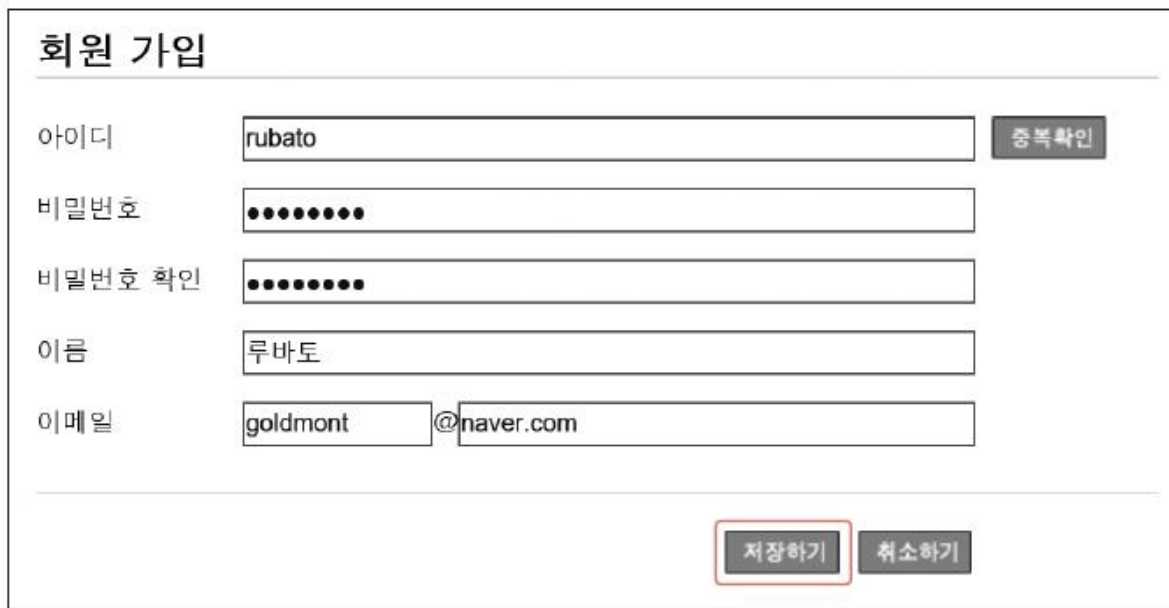
[예제 11-2] 회원 가입 페이지 `part3/member_form.php`

⑭ <저장하기> 버튼

- 사용자가 회원 가입 정보를 입력하고 하단의 <저장하기> 버튼(button_save.gif)을 클릭하면 자바스크립트의 onclick 이벤트 핸들러에 설정된 check_input() 함수가 실행.
- 사용자가 입력한 데이터는 ④의 document.member_form.submit()에 의해 ⑦의 action 속성에 설정된 member_insert.php로 전달됨과 동시에 페이지 이동.
- member_insert.php에서는 사용자가 입력한 데이터를 DB에 저장.

4. 회원 가입 정보 저장

- 회원 가입 페이지에 다음과 같이 입력한 다음 <저장하기>를 클릭.



회원 가입

아이디 rubato 중복확인

비밀번호

비밀번호 확인

이름 루바토

이메일 goldmont @naver.com

저장하기 취소하기

그림 11-12 회원 가입 페이지(member_form.php)에 데이터 입력

- 회원 가입 페이지에서 <저장하기>를 클릭하면 [예제 11-2]의 ⑦에서 <form> 태그의 action 속성에 설정한 member_insert.php로 페이지가 이동하고, 사용자가 입력한 데이터는 members 테이블에 저장.

[예제 11-3] 회원 가입 정보 저장 페이지

part3/member_insert.php

```
<?php
```

```
$id = $_POST["id"]; ①
```

```
$pass = $_POST["pass"];
```

```
$name = $_POST["name"];
```

```
$email1 = $_POST["email1"];
```

```
$email2 = $_POST["email2"];
```

```
$email = $email1."@".$email2; ②
```

```
$regist_day = date("Y-m-d (H:i)"); // 현재의 '연-월-일-시-분' 저장 ③
```

```
$con = mysqli_connect("localhost", "user1", "12345", "sample"); ④
```

```
$sql = "insert into members(id, pass, name, email, regist_day,  
level, point) "; ⑤
```

[예제 11-3] 회원 가입 정보 저장 페이지 `part3/member_insert.php`

```
$sql .= "values('$id', '$pass', '$name', '$email', '$regist_day', 9, 0)";
```

```
mysqli_query($con, $sql);    // $sql에 저장된 명령 실행 ⑥
```

```
mysqli_close($con); ⑦
```

```
echo "
```

```
    <script>
```

```
        location.href = 'index.php'; ⑧
```

```
    </script>
```

```
    ";
```

```
?>
```


[예제 11-3] 회원 가입 정보 저장 페이지 `part3/member_insert.php`

① 전역 변수 \$_POST

- \$_POST["id"]는 전역 변수 \$_POST의 id가 지시하는 요소의 값, 즉 [예제 11-2]의 ⑧에서 <input> 태그의 name 속성에 설정된 id를 의미.
- 이 값은 회원 가입 페이지에서 사용자가 아이디 입력 창에 입력한 데이터.
- 사용자가 입력한 비밀번호, 이름, 이메일은 \$_POST["pass"], \$_POST["name"], \$_POST["email1"], \$_POST["email2"]로 전달받음.

② 이메일 주소 하나로 합치기

- 사용자가 나누어 입력한 이메일 주소를 하나로 합쳐서 \$email에 저장.

③ 현재 날짜와 시간

- date() 함수를 이용하여 구한 현재 날짜와 시간을 \$regist_day에 저장.
- date() 함수는 괄호 안에 있는 인자의 형식 문자에 따라 날짜와 시간의 출력 형식이 결정.

[예제 11-3] 회원 가입 정보 저장 페이지

part3/member_insert.php

표 11-4 date() 함수의 형식 문자

형식 문자	의미	반환 값
a	소문자로 오전과 오후	am, pm
A	대문자로 오전과 오후	AM, PM
d	앞에 0이 붙는 두 자리 형식의 일	01~31
D	세 글자 문자 형식의 요일	Mon~Sun
g	0이 붙지 않는 12시간 형식의 시	1~12
G	0이 붙지 않는 24시간 형식의 시	0~23
h	0이 붙는 12시간 형식의 시	01~12
H	0이 붙는 24시간 형식의 시	00~23
i	0이 붙는 형식의 분	00~59
m	0이 붙는 숫자 형식의 월	01~12
M	세 글자 문자 형식의 월	Jan~Dec
s	0이 붙는 형식의 초	00~59
Y	네 자릿수 연도	예: 1999, 2005
y	두 자릿수 연도	예: 99, 05

③의 date("Y-m-d (H:i)")는 현재 날짜와 시간을 2015-09-09 (13:53) 형태로 표시.

[예제 11-3] 회원 가입 정보 저장 페이지 `part3/member_insert.php`

④ MySQL 서버 접속

- 사용 중인 컴퓨터에 설치된 MySQL의 계정(user1), 비밀번호(12345), 데이터베이스(sample)에 연결.
- 연결 성공하면 \$con에 연결 식별자 번호 저장, 실패하면 \$con에 false 값 저장.

mysqli_connect() 함수	
형식	mysqli_connect(host, username, password, dbname);
기능	MySQL 서버와의 연결을 시도한다.
인자	host: 서버의 호스트명 또는 '호스트명 : 포트 번호' username: 사용자 계정 password: 사용자 계정의 비밀번호 dbname: 사용할 데이터베이스
반환 값	성공: 식별자 번호, 실패: false

[예제 11-3] 회원 가입 정보 저장 페이지 `part3/member_insert.php`

⑤ MySQL 명령 설정

- members 테이블에 데이터를 저장하는 MySQL 명령을 `$sql`에 저장.
- 회원 레벨을 나타내는 level 필드에는 기본 레벨인 9 저장.
- 회원 포인트를 의미하는 point 필드에는 초깃값으로 0 입력.

⑥ MySQL 명령 실행

- `mysqli_query()` 함수를 이용하여 MySQL 명령을 실행.

mysqli_query() 함수	
형식	<code>mysqli_query(connection, query);</code>
기능	설정된 MySQL 연결에서 해당 MySQL 명령을 실행한다.
인자	connection: <code>mysqli_connect()</code> 에 의해 설정된 연결 식별자 query: 실행할 MySQL 쿼리 명령
반환 값	성공: SELECT, SHOW, DESCRIBE, EXPLAIN 명령 성공 시 <code>mysqli_result</code> 객체 반환, 그 외 명령 성공 시 <code>true</code> 반환 실패: <code>false</code>

[예제 11-3] 회원 가입 정보 저장 페이지 `part3/member_insert.php`

⑦ MySQL 명령 실행

`mysqli_close()` 함수를 이용하여 MySQL 서버와의 연결을 끊음.

mysqli_close() 함수	
형식	<code>mysqli_close(connection);</code>
기능	MySQL 서버와의 연결을 해제한다.
인자	connection: 연결 식별자
반환 값	성공: true, 실패: false

⑧ 메인 페이지로 이동

`location.href = 'index.php';`는 메인 페이지인 `index.php`로 이동하라는 의미. 이는 자바스크립트 코드로 사용 형식은 다음과 같음.

```
location.href = "경로 및 파일명";
```

[예제 11-3] 회원 가입 정보 저장 페이지 part3/member_insert.php

- phpMyAdmin에 접속하여 실제로 members 테이블에 새로운 레코드가 생성되는지 확인 가능.
- [예제 11-3]의 member_insert.php가 실행되면 회원 가입 폼 양식에서 사용자가 입력한 데이터가 다음과 같이 members 테이블에 저장.

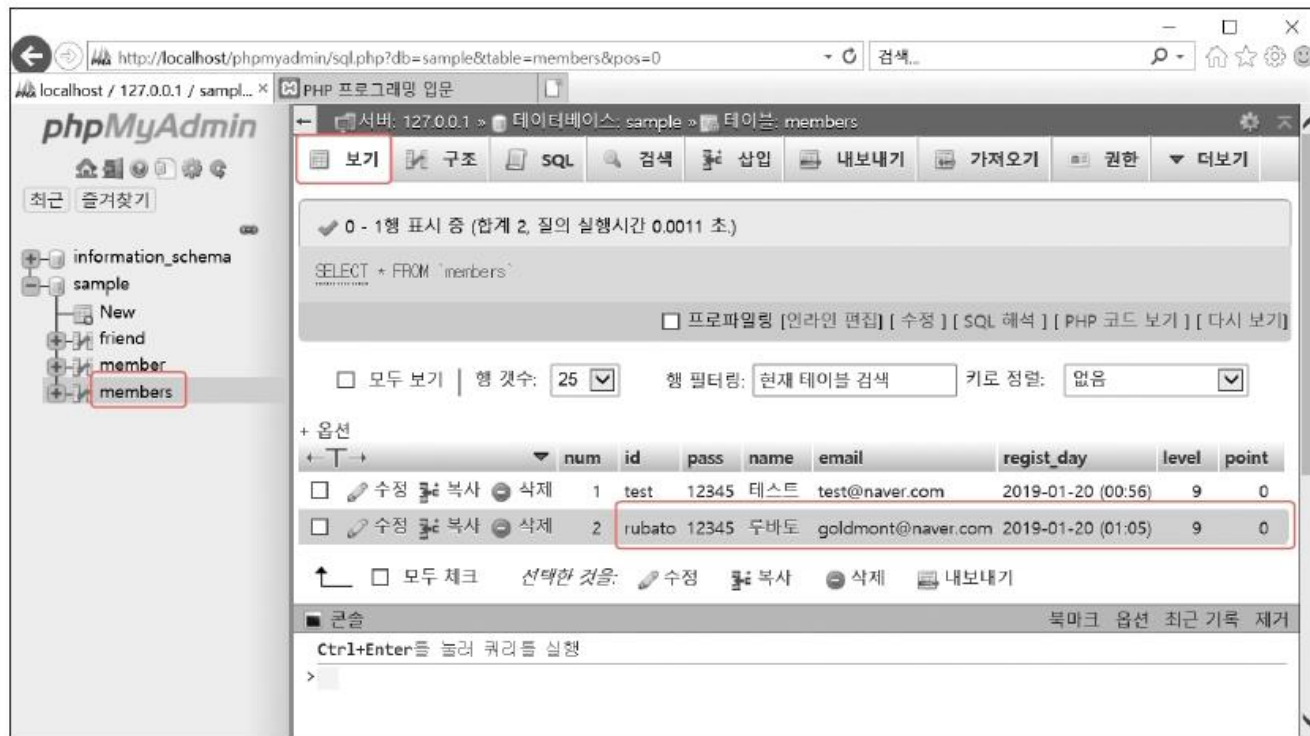


그림 11-13 members 테이블에 저장된 회원 가입 정보

- <중복확인> 버튼(check_id.gif)을 클릭하면 onclick 이벤트 핸들러에 설정된 check_id() 함수가 실행. 이 함수는 [예제 11-2]의 ⑥에 정의되어 있음.
- check_id() 함수가 실행되면 window.open() 함수에 의해 member_check_id.php가 새 창으로 열림.

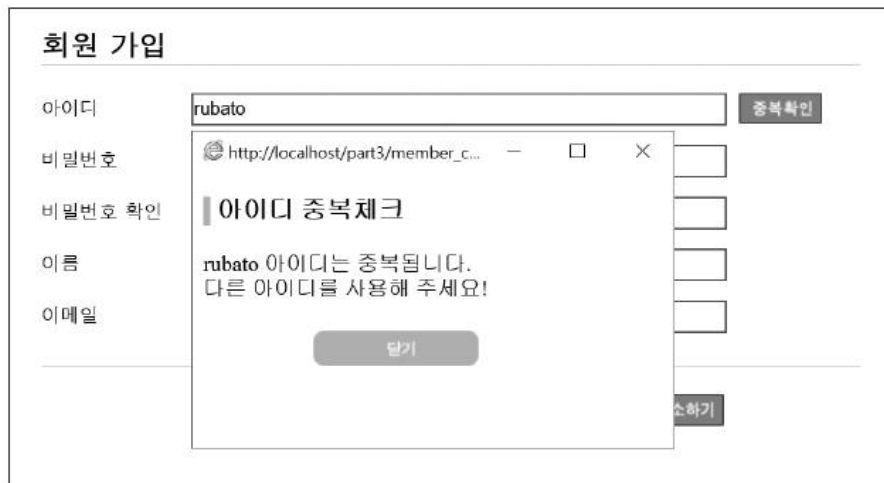


그림 11-14 아이디 중복 확인 페이지

[예제 11-4] 아이디 중복 확인 페이지

part3/member_check_id.php

```
<!-- 생략 -->
<h3>아이디 중복체크</h3>
<p>
<?php
    $id = $_GET["id"];                                ①
    if (!$id)                                          ②
    {
        echo("<li>아이디를 입력해 주세요!</li>");
    }
    else
    {
        $con = mysqli_connect("localhost", "user1", "12345", "sample"); ③
        $sql = "select * from members where id='$id'";
        $result = mysqli_query($con, $sql);
        $num_record = mysqli_num_rows($result);
```


[예제 11-4] 아이디 중복 확인 페이지 part3/member_check_id.php

```
if ($num_record) ④
{
    echo "<li>".$id." 아이디는 중복됩니다.</li>";
    echo "<li>다른 아이디를 사용해 주세요!</li>";
}
else
{
    echo "<li>".$id." 아이디는 사용 가능합니다.</li>";
}
mysqli_close($con);
}
?>
</p>
<div id="close">
    
</div>
<!-- 생략 -->
```

[예제 11-4] 아이디 중복 확인 페이지 `part3/member_check_id.php`

① 아이디 전달받기

- [예제 11-2]의 ⑥에서 `window.open()` 함수의 첫 번째 인자는 다음과 같음.

```
member_check_id.php?id=" + document.member_form.id.value
```

- `check_id.php`에서는 중복 확인을 위해 사용자가 입력한 아이디를 GET 방식으로 전달받음.
- 사용자가 아이디 입력 창에 입력한 `document.member_form.id.value` 값을 전역 변수 `$_GET["id"]`에 의해 전달받음.

② 아이디 입력 검사

- 아이디 입력 창에 데이터 입력하지 않고 <중복확인> 버튼 클릭하면 `$id`는 null 값이 됨.
- if문 조건식인 `!$id`가 참이 되어 '아이디를 입력해 주세요!'라는 메시지 경고 창을 띄움.

[예제 11-4] 아이디 중복 확인 페이지 [part3/member_check_id.php](#)

③ members 테이블 검색

- 사용자가 아이디를 입력하면 members 테이블에 동일한 아이디가 있는지 검사. `mysqli_connect()` 함수로 다음 MySQL 명령을 실행하여 동일한 아이디가 있는지 검색.

```
select * from members where id='$id'
```

- 명령을 실행한 결과를 저장한 `$result`의 레코드 개수를 `mysqli_num_rows()` 함수로 세고 그 결과를 `$num_record`에 저장.

mysqli_num_rows() 함수	
형식	<code>mysqli_num_rows(result);</code>
기능	result의 행 수, 즉 레코드 개수를 반환한다.
인자	result: <code>mysqli_query()</code> 함수를 실행한 반환 값
반환 값	result의 행 수, 즉 레코드 개수

[예제 11-4] 아이디 중복 확인 페이지 `part3/member_check_id.php`

④ 결과 메시지 출력

- \$num_record가 값을 가지면 DB에 동일한 아이디가 존재한다는 것을 의미.
- if문의 조건식이 참이 되어 \$id의 값과 함께 '아이디는 중복됩니다. 다른 아이디를 사용해 주세요!'라는 메시지를 출력.
- \$num_record가 null 값이면 DB에 동일한 아이디가 존재하지 않는다는 의미이므로 '아이디는 사용 가능합니다.'라는 메시지를 출력.

PHP 프로그래밍 입문 3판



감사합니다.