



영진전문대학교

글로벌시스템융합과 - GSC

PHP – Board (게시판)

기본형

정영철 교수

글로벌시스템융합과



A.I WITH BETTER LIFE

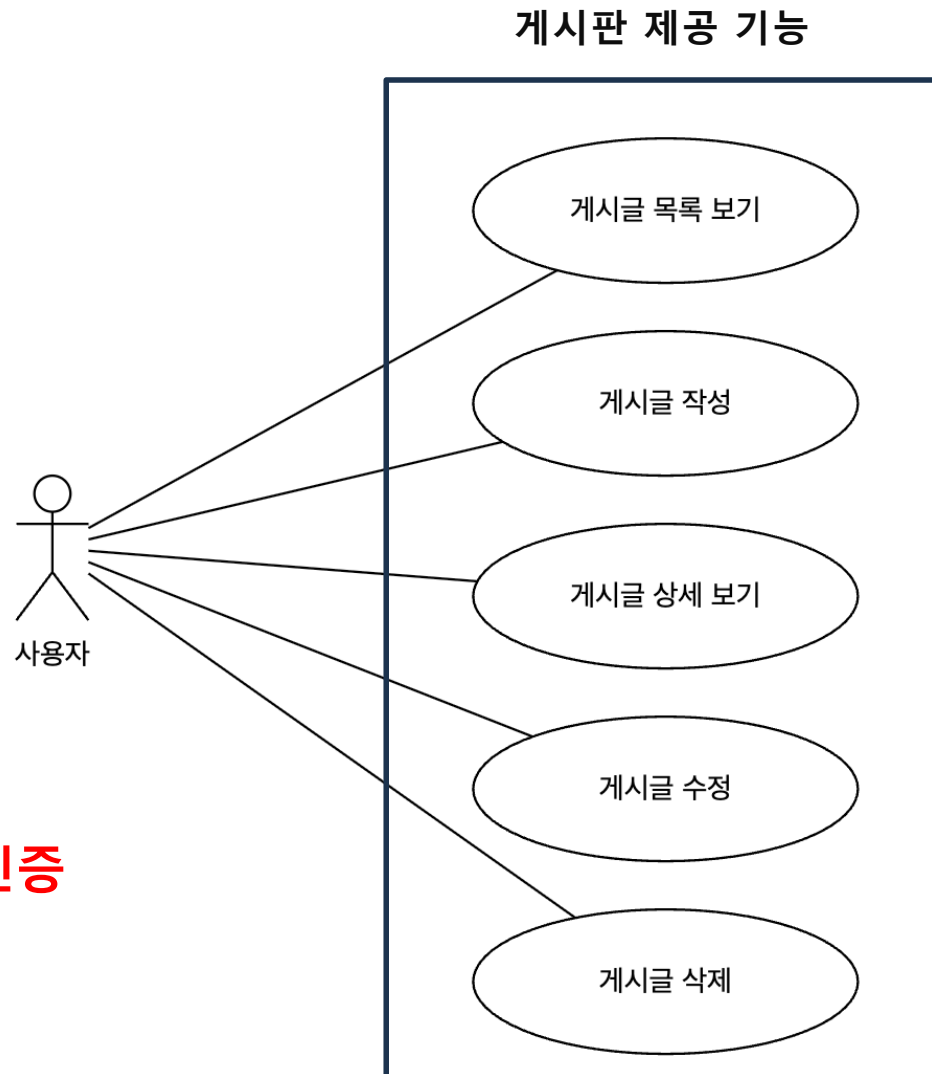
Board(게시판) 기능 요구사항

기능	요구사항
글 작성	<ul style="list-style-type: none">이름, 제목, 내용, 비밀번호를 모두 입력해야 함입력된 내용은 DB에 저장됨글 작성 완료 후 글 목록 보기 이동
글 목록 보기	<ul style="list-style-type: none">최신 글이 위로 정렬됨한 페이지당 5개 글 표시페이지 이동 기능 포함 (페이징)
글 상세 보기	<ul style="list-style-type: none">제목, 작성자, 작성일, 수정일(있으면), 본문을 표시함
글 수정	<ul style="list-style-type: none">글 상세보기 페이지에서 수정 요청 가능수정 전 비밀번호 확인 필요- 수정 완료 시 updated_at 자동 갱신
글 삭제	<ul style="list-style-type: none">글 상세보기 페이지에서 삭제 요청 가능삭제 전 비밀번호 확인 필요- 삭제되면 목록으로 이동
기타 조건	<ul style="list-style-type: none">로그인 없이 누구나 사용 가능보안 위해 비밀번호는 해쉬값으로 DB에 저장

Use case diagram : 기본형 게시판

- 사용자(User)는 로그인 없이 다음 기능을 수행

- ① 게시글 목록 보기
- ② 게시글 작성
- ③ 게시글 상세 보기
- ④ 게시글 수정
- ⑤ 게시글 삭제



➤ 작성 시 입력한 비밀번호로 인증

DB schema

```
-- 데이터베이스 생성 (존재하지 않으면)
CREATE DATABASE IF NOT EXISTS gsc CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

-- gsc 데이터베이스 사용
USE gsc;

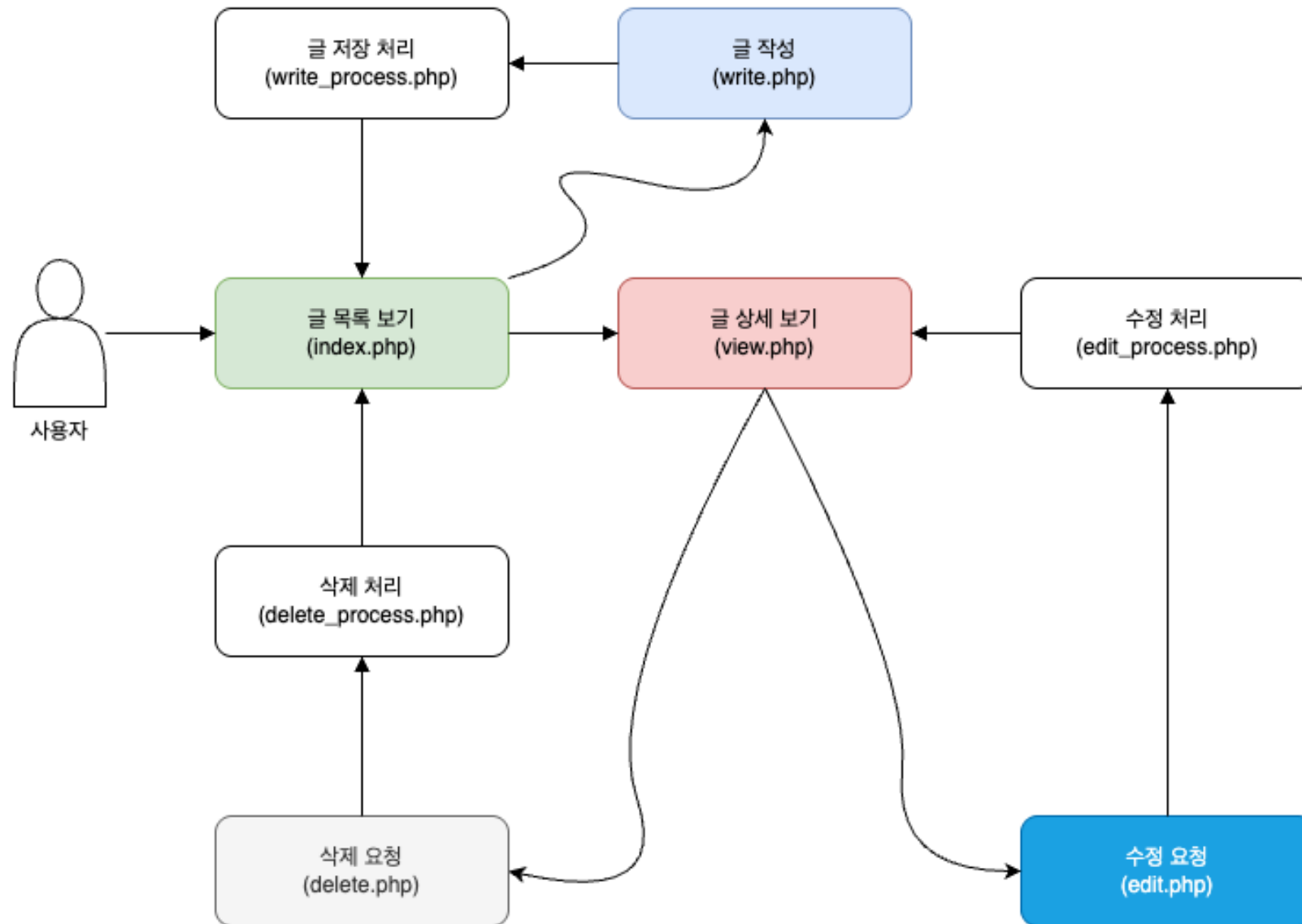
-- 게시판 테이블 생성
CREATE TABLE IF NOT EXISTS posts (
    id INT AUTO_INCREMENT PRIMARY KEY,           -- 게시글 ID
    title VARCHAR(255) NOT NULL,                 -- 제목
    name VARCHAR(100) NOT NULL,                  -- 작성자 이름
    password VARCHAR(255) NOT NULL,              -- 비밀번호 (해시값)
    content TEXT NOT NULL,                       -- 내용
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP, -- 작성 시각
    updated_at DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP -- 수정 시각 (수정 시 자동 갱신)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

• 게시판 테이블 구조 : posts

컬럼명	자료형	제약조건 및 속성	설명
id	INT	PRIMARY KEY , AUTO_INCREMENT	• 게시글 고유 ID
title	VARCHAR(255)	NOT NULL	• 게시글 제목
name	VARCHAR(100)	NOT NULL	• 작성자 이름
password	VARCHAR(255)	NOT NULL	• 게시글 비밀번호 (해시값)
content	TEXT	NOT NULL	• 게시글 내용
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	• 작성 시각
updated_at	DATETIME	DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP	• 수정 시각 (수정 시 자동 기록)

- 비회원 게시판이므로 작성자 이름, 비밀번호를 입력 받아 저장

Board 기능 흐름도



기능별 파일 구성표

기능 구분	파일명	주요 역할 및 구성 요소
글 목록 보기	index.php	• 글 리스트 출력, 페이징, 글 클릭 링크
글 작성 화면	write.php	• 제목/이름/내용/비밀번호 입력 form
글 저장 처리	write_process.php	• DB에 INSERT 후 목록으로 이동
글 상세 보기	view.php	• 제목, 내용, 작성일 등 표시, 수정/삭제 버튼
글 수정 화면	edit.php	• 기존 데이터 표시, 수정 form, 비밀번호 확인
글 수정 처리	edit_process.php	• DB UPDATE 후 상세 보기로 이동
글 삭제 요청 화면	delete.php	• 글 ID로 비밀번호 입력 form 제공
글 삭제 처리	delete_process.php	• 비밀번호 확인 후 DB 삭제 → 글 목록으로 이동

페이지네이션(Pagination) 이란?

- 많은 양의 데이터를 한 번에 모두 보여주지 않고, 여러 페이지로 나누어 일부씩 보여주는 UI 및 로직 기법
- 왜 필요한가?
 - 사용자 UX 개선
 - 서버 부하 감소
 - 네트워크 효율 향상
- 예시
 - ✓ 게시판, 쇼핑몰 상품 목록, 검색 결과 등
 - ✓ 1 2 3 ... 형태의 페이지 링크 제공

번호	제목	작성자	작성일
30	25		2025-07-31
29	24		2025-07-31
28	23		2025-07-31
27	22		2025-07-31
26	21		2025-07-31

페이지: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#)

[글쓰기](#)

Pagination

번호	제목	작성자	작성일
25	20		2025-07-31
24	19		2025-07-31
23	18		2025-07-31
22	17		2025-07-31
21	16		2025-07-31

페이지: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#)

[글쓰기](#)

페이지네이션(Pagination) 알고리즘

```
$page = isset($_GET['page']) ? max(1, intval($_GET['page'])) : 1;
```

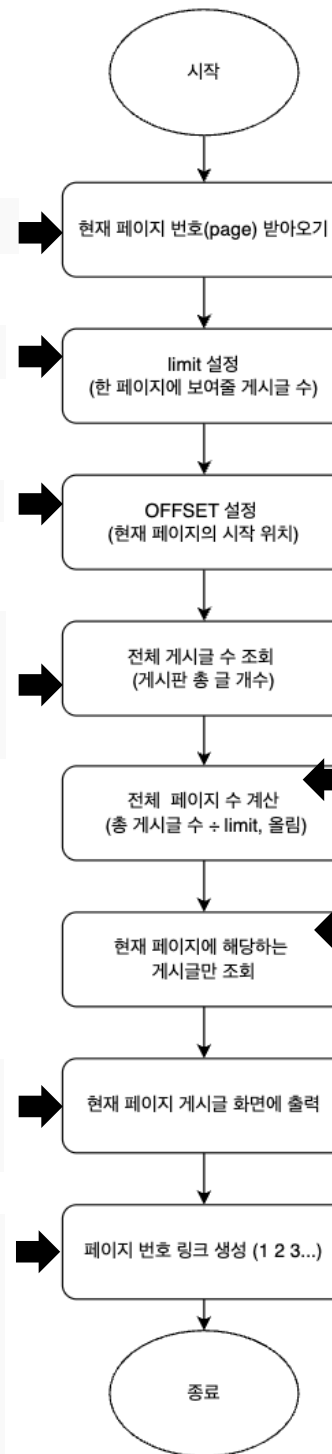
```
$limit = 5; // 예: 한 페이지에 5개 게시물
```

```
$offset = ($page - 1) * $limit;
```

```
$sql = "SELECT COUNT(*) AS total FROM posts";
$result = $conn->query($sql);
$total_posts = $result->fetch_assoc()['total'];
```

```
for ($i = 1; $i <= $total_pages; $i++) {
    echo ($i == $page) ? "<strong>$i</strong> " : "<a href='?page=$i'>$i</a> ";
}
```

```
while ($post = $posts->fetch_assoc()) {
    echo "<tr>";
    echo "<td>" . htmlspecialchars($post['id']) . "</td>";
    echo "<td>" . htmlspecialchars($post['title']) . "</td>";
    echo "</tr>";
}
```



```
$total_pages = ceil($total_posts / $limit);
```

```
$sql = "SELECT id, title, name, created_at
FROM posts
ORDER BY id DESC
LIMIT $limit OFFSET $offset";
$posts = $conn->query($sql);
```


SQL로 특정 페이지 게시물만 불러오기 (LIMIT & OFFSET)

```
-- 1페이지 (page = 1)
SELECT * FROM posts ORDER BY id DESC
LIMIT 5 OFFSET 0;
```

정렬된 결과에서

첫 번째 행부터 5개의 행 가져옴 : **1행 ~ 5행**

```
-- 2페이지 (page = 2)
SELECT * FROM posts ORDER BY id DESC
LIMIT 5 OFFSET 5;
```

정렬된 결과에서 앞의 5개 행을 건너뛰고,

그 다음 5개 행을 가져옴 : **6행 ~ 10행**

```
-- 3페이지 (page = 3)
SELECT * FROM posts ORDER BY id DESC
LIMIT 5 OFFSET 10;
```

정렬된 결과에서 앞의 10개 행을 건너뛰고,

그 다음 5개 행을 가져옴 : **11행 ~ 15행**

◆ 현재 페이지 값에 따른 OFFSET 계산 식

OFFSET = (page - 1) * limit

Extra Slides

Q/A

감사합니다



주문식교육의 산실
영진전문대학교