

# Cykor 3주차

## APM을 이용한 CRUD 게시판 구현 보고서

학번 : 2025350213

이름 : 승창민

### 1. 과제 목적

본 과제는 웹 개발의 기초를 학습하고, PHP와 MySQL을 활용하여 동적인 게시판 시스템을 구현하는 데 있다. 특히, 사용자 인증, 데이터베이스 연동, 입력 검증, 보안 처리 등 실전에서 필수적인 웹 개발 요소들을 경험하는 것이 핵심 목표다.

### 2. 프로그램 개발 과정

#### [1] 환경설정

- Docker-Compose

Docker 데스크탑 설치 & 환경설정을 챗GPT의 도움을 받았더니 나도 모르는 사이에 완성이 되어 버림. YAML 언어를 사용한다는 것 같은데, 다음에 기회가 되면 공부를 해봐야겠다.

- scr 폴더 생성

게시판 구현을 어떻게 시작할지 막막해 gpt에게 물어보았고, 하나의 파일에 php로 이루어진 핵심 로직들을 담아 구조화하는 방식을 추천받음.

#### [2]

#### ● 회원가입 (register.php & register\_process.php)

##### “register.php”

- 회원가입 양식을 사용자에게 제공. 대체로 html 사용.
- register.html로 만들었는데 제대로 작동이 안 돼서 .php로 바꿈.
- utf-8 통해 한글이 안 깨지게 방지.

##### “register\_process.php”

- register.php에서 입력된 값을 받기
- 입력값 확인
- 데이터베이스에 사용자 정보 저장
- 비밀번호 암호화 (GPT 추천)

#### ● 로그인(login.php & login\_process.php)

##### “login.php”

- 유사할 것으로 생각해 register.php의 양식을 많이 참조함

#### “login\_process.php”

- login에서 id와 password를 받음 -> \$username과 \$password에 저장
- DB서 아이디 비교하고, 아이디 있을 시 비밀번호 비교
- 맞는 경우 세션에 로그인 상태 저장.
- 틀린 경우 에러 메시지 출력

### [3]

#### ● 메인 페이지(main.php), 로그아웃(logout.php)

##### “main.php”

- 로그인 이외의 경로로 들어올 시 로그인 창으로 돌려보냄
- html 위주의 게시판 본체

##### “logout.php”

- 세션 종료
- 인터넷에 있는 다른 게시판 양식을 참고했음.

##### “login.php”

- 로그인 성공 시 바로 main.php로 이어지도록 만듦

#### ● 게시글 작성(write.php & write\_process.php)

##### “write.php”

- 원래 이름은 writing.php였는데, 만들다 보니까 헛갈려서 나중에 강 파일명을 write로 바꿈
- 로그인 없이는 이용 불가
- html 위주의 양식 제공

##### “writing\_process.php”

- 제목, 내용, 글쓴이 입력값을 받아옴
- 빈값이 있으면 에러 메시지를 출력.
- SQL 삽입 처리 (이 부분은 익숙하지 않아 검색결과 참조)
- 글을 다 썼으면 list라는 게시판으로 보내기로 했음 (아직 list는 안 만들었었지만)
- 원래 bind\_param(“sss”....) 부분의 sss를 아무 변수로나 설정했었는데 작동이 잘 안됨.  
GPT에게 물어보니 s가 string을 의미하는 거라고 함.
- 작성 성공 시 알림이 오도록 함 (GPT 추천)

### [4]

#### ● 글 수정(eddit.php & eddit\_process.php)

##### “eddit.php”

- 글 작성자 외에는 수정 권한 x

- 게시글을 불러와서 글 수정 기능 넣음
- write.php를 참조한 html 위주의 양식 제공

#### “eddit\_process.php”

- author 조회 통해 작성자 확인 (일치 시에만 수정 가능)
- sql 쿼리 준비하고 title, content 수정
- 수정 끝나면 수정 성공 메시지 출력

#### ● 게시판 열람 (view.php)

##### “view.php”

- 원래 이름은 viewer.php였지만 역시 헛갈려서 그냥 view로 바꿈

### [5]

#### ● 글 삭제(delete\_process.php)

##### “delete\_process.php”

- 글 작성자 혹은 관리자인지 확인 (관리자 권한을 설정 이전이라 대충 admin으로 설정)  
= 삭제 권한 있는지 확인
- 권한 있을시 삭제

#### ● 관리자 권한 추가

- 원래는 다른 커뮤니티처럼 관리자가 관리자를 임명하고, 관리자 list만 확인하는 기능을 넣고 싶었는데....어떻게 해야 할지 생각이 안 나서 포기
- phpmyadmin에 가서 관리자 권한을 추가하는 sql 작성

##### “login.php”

- 잘 되던 게시판이 갑자기 먹통이 됨
- GPT에게 이유 문의하며 이것저것 물어봄
- 실제 이유는 docker-compose의 충돌이었으며 잠시 down했다 올리니 해결
- 번외로 gpt가 login 방식을 \$row 방식으로 바꾸는 걸 추천함

### [6]

#### ● 회원 목록(user\_list.php) & 회원 삭제 기능(delete\_user.php)

##### “user\_list.php” & “delete\_user.php”

- 관리자 권한이 있는 경우에만 접근 가능
- html로 회원 목록과 가입 일시를 띄우는 user\_list.php를 기반으로 삭제 기능 구현
- 관리자는 자기 자신을 제외한 다른 user를 삭제 가능

#### ● Cykor 이미지 & 이스터에그 넣기

- 심심해서 그냥 하나 넣음

- 버튼을 하나 만들고, 버튼을 정해진 횟수 만큼 누르면 문구가 나오는 기능 넣음
- txt 파일에 숫자 저장하고, 간단한 조건 & 반복문으로 구현

[7]

#### ● Id 중복 체크

“register\_process”

- 동일 아이디로 로그인 가능할 수도 있겠단 생각이 들어서 막아야겠다고 생각함
- check 결과가 존재하는 경우 “이미 존재하는 아이디입니다”라는 경고 띄우기

#### ● 내 게시물 모아보기(my\_post.php)

“my\_post.php”

- 전반적인 양식은 view.php 내용을 참조
- 사용자의 게시글을 최신순으로 가져오는 sql 쿼리 구문 추가

#### ● main.php 개선

- 글 작성을 하지 않으면 글 목록을 볼 수가 없는 이슈가 있어서 바로가기에 넣어둠.

### 3. 주요 기능

<< 사용자 기능 >>

- 회원가입 / 로그인 / 로그아웃
- 로그인 상태 유지 (session 사용)
- 게시글 작성, 수정, 삭제 (본인만 가능)
- 본인이 작성한 글 목록 보기
- 전체 게시판 목록 보기
- 아이디 중복 검사

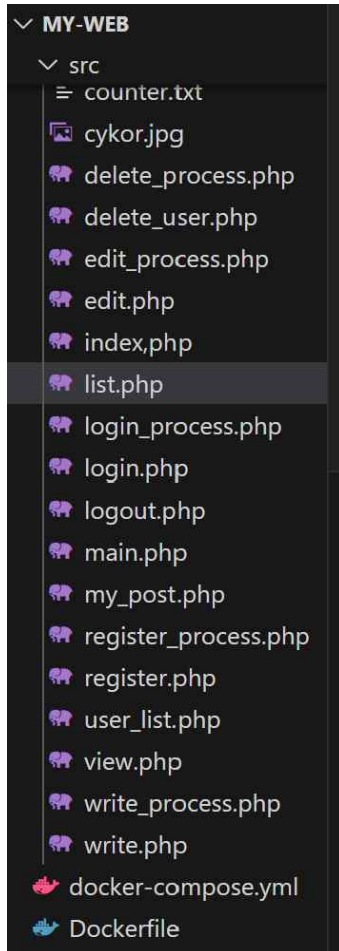
<< 관리자 기능 >>

- 관리자 계정 식별 및 권한 부여 (is\_admin 컬럼)
- 전체 회원 목록 보기
- 회원 삭제 기능
- 모든 게시글 삭제 가능

<< 기타 기능 >>

- 메인페이지에 버튼 클릭 이스터에그
- Docker + Docker Compose로 서비스 구성
- htmlspecialchars() 및 password\_hash()로 기본 보안 적용

#### 4. 디렉토리 구조



#### 5. 실행 예시

안녕하세요, administrator님!

페이지에 방문하신 걸 환영합니다!!

[로그아웃](#)

- [게시글 작성하기](#)
- [내가 쓴 글 보기](#)
- [게시글 보기](#)
- [회원 목록 보기](#)



- Cykor -

[돌러보세요](#)

[그림 1]

## 게시글 목록

| 글 번호 | 제목                            | 글쓴이           | 작성 시간               | 관리                                    |
|------|-------------------------------|---------------|---------------------|---------------------------------------|
| 6    | <a href="#">아하</a>            | guest2        | 2025-06-04 19:28:58 | <a href="#">수정</a> <a href="#">삭제</a> |
| 5    | <a href="#">게시판 이용자들에게 안내</a> | administrator | 2025-06-04 19:28:17 | <a href="#">수정</a> <a href="#">삭제</a> |
| 3    | <a href="#">commit 3</a>      | guest1        | 2025-06-04 18:18:19 | <a href="#">수정</a> <a href="#">삭제</a> |
| 2    | <a href="#">commit</a>        | guest1        | 2025-06-04 18:14:35 | <a href="#">수정</a> <a href="#">삭제</a> |

[글쓰기](#)

[메인 화면으로](#)

[그림 2]

## 전체 회원 목록

| 아이디                 | 가입일                 |                    |
|---------------------|---------------------|--------------------|
| guest2              | 2025-06-04 18:49:13 | <a href="#">삭제</a> |
| administrator (관리자) | 2025-06-04 18:44:06 | (본인)               |
| guest1              | 2025-06-04 18:00:20 | <a href="#">삭제</a> |

[메인으로 돌아가기](#)

[그림 3]

## 6. 느낀 점

과제 시작 전에 html과 php를 공부하고 들어갔음에도 어떻게 구현해야 할지 막막해서, 초반 부에는 돌아다니는 양식이나 gpt를 많이 참고하였다. 하지만 진행하며 각 php 파일에 있는 공통점이 눈에 보이고, 확신을 가지고 작성한 코드가 실제로 작동되는 걸 보았을 때 웹 개발 실력이 발전하고 있음을 느낄 수 있었다. 특히 과제 시작 전에 예시 코드들을 볼 때는 전혀 이해되지 않았던 코드들이, 공부 이후 조금씩 해석이 되는 걸 보며 뿌듯함을 느꼈다.