

web_1 김동훈

기본 설정

Apache, PHP, MySQL 설치 확인

```
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1$ apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built: 2025-04-03T09:05:48
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1$ php -v
PHP 8.1.2-1ubuntu2.21 (cli) (built: Mar 24 2025 19:04:23) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
with Zend OPcache v8.1.2-1ubuntu2.21, Copyright (c), by Zend Technologies
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1$ mysql --version
mysql Ver 8.0.42-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1$
```

```
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1$ sudo ln -s /mnt/c/ku/cykor/2_1/web1 /var/www/html/web1
[sudo] password for racon:
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1$
```

database 생성

- create_database.sql

```
CREATE DATABASE IF NOT EXISTS cykor_web1_db;
USE cykor_web1_db;

-- user 정보 데이터 저장 테이블 생성
CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY, -- id 저장
    username VARCHAR(50) NOT NULL UNIQUE, -- 사용자 이름 저장, 동일한 이름 저장 불가
    password VARCHAR(255) NOT NULL, -- 비밀번호 저장
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP -- 계정 생성 시간 저장
);

-- 게시물 데이터 저장 테이블 생성
CREATE TABLE IF NOT EXISTS posts (
    id INT AUTO_INCREMENT PRIMARY KEY, -- 게시물 고유번호, 자동 증가
    title VARCHAR(100) NOT NULL, -- 게시물 제목, 공란 불가
    content TEXT NOT NULL, -- 게시물 내용, 공란 불가
    user_id INT NOT NULL, -- user id
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP, -- 글 작성 시간
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, -- 글 수정 시간, 현재 시간으로
    FOREIGN KEY (user_id) REFERENCES users(id) -- user_id 요소는 항상 user 테이블에 존재해야한다
);
```

my sql에서 해당 프로그램 실행

이후 정상적으로 sql 데이터베이스가 형성되어있는 것을 터미널에서 확인할 수 있었다

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'racon123!';
```

```
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1/cykor_web_1$ sudo mysql  
[sudo] password for racon:
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'racon123!';  
Query OK, 0 rows affected (0.03 sec)  
  
mysql> SELECT user, host, plugin FROM mysql.user WHERE user = 'root';  
+-----+-----+-----+  
| user | host      | plugin |  
+-----+-----+-----+  
| root | localhost | mysql_native_password |  
+-----+-----+-----+  
1 row in set (0.01 sec)  
  
mysql> █
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'racon123!';
```

생성된 mysql 테이블 확인

```
mysql> SOURCE create_database.sql  
Query OK, 1 row affected, 1 warning (0.01 sec)
```

Database changed

```
Query OK, 0 rows affected (0.15 sec)
```

```
Query OK, 0 rows affected (0.05 sec)
```

```
mysql>
```

```
mysql> SHOW TABLES;  
+-----+  
| Tables_in_cykor_web1_db |  
+-----+  
| posts                    |  
| users                    |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql> █
```

DB 접속 - php

db.php ⇒ sql 접속을 위한 코드

이를 통하여 각 php 파일에서 위에서 생성한 db에 접근하는 것이 가능해진다

```
<?php
$host = 'localhost';
$dbname = 'board_db';
$user = 'root';
$pass = 'racon123!';

$conn = new mysqli($host, $user, $pass, $dbname); // mysql 접속

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error); //에러 발생시 출력
}
?>
```

로그인 관련 코드

회원가입

```
<?php
include '../db/db.php'; // db.php를 통해 접속

if ($_SERVER['REQUEST_METHOD'] === 'POST') { //사용자가 <form method="POST">로 요청을 보낸 경우 데이터 저장장
    $username = $_POST['username']; //사용자 입력 받아오기
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT); // 비밀번호 해시화

    // 중복 사용자 확인
    $check = $connection->prepare("SELECT id FROM users WHERE username = ?"); //? 부분에 username을 넣을 예정이다
    $check->bind_param("s", $username); //문자열이 ?자리에 들어간다 => username요소를 binding
    $check->execute(); //쿼리 실행
    $check->store_result(); //쿼리 결과를 저장

    if ($check->num_rows > 0) { //이미 해당 요소가 존재하는 경우(찾은 개수가 0보다 클때)
        echo "이미 존재하는 사용자";
    } else {
        // 새 사용자 등록
        $statement = $connection->prepare("INSERT INTO users (username, password) VALUES (?, ?)"); //DB에 저장하기 위한
        $statement->bind_param("ss", $username, $password); //모두 문자열로 binding
        $statement->execute(); //쿼리문 실행
        echo "회원가입 완료";
    }
}
?>

<!-- 회원가입 폼 -->
<?php include '../header.php'; ?>
<h2>회원가입</h2>
<form method="POST">
    사용자명: <input name="username" required><br>
    비밀번호: <input type="password" name="password" required><br>
```

```
<button type="submit">회원가입</button>
</form>
```

로그아웃

```
<?php
session_start();    // 기존 세션 불러오기
session_unset();    // 세션 변수 모두 삭제
session_destroy();  // 세션 파괴

// 로그인 페이지로 리디렉션
header("Location: /web1/index.php");
exit;
?>
```

로그인 상태 확인

```
<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    echo "로그인 필요";
    exit;
}
?>
```

로그인

```
<?php
session_start(); // 세션 시작 -> 로그인 성공시 $_SESSION에 사용자 정보가 저장된다
include '../db/db.php'; // DB 연결

if ($_SERVER['REQUEST_METHOD'] === 'POST') { //post를 통해서 데이터를 받아온 경우
    $username = $_POST['username'];
    $password = $_POST['password'];

    $statement = $connection->prepare("SELECT id, password FROM users WHERE username = ?"); //쿼리문 준비
    $statement->bind_param("s", $username); //username 정보 문자열로 저장장
    $statement->execute(); //쿼리문 실행
    $statement->store_result(); //실행결과 저장

    if ($statement->num_rows === 1) { //해당 사용자가 1명 존재하는 경우 로그인 절차 확인
        $statement->bind_result($id, $hashedPassword); //위의 select 결과를 저장
        $statement->fetch();

        if (password_verify($password, $hashedPassword)) { //비밀번호 비교
            $_SESSION['user_id'] = $id;
            $_SESSION['username'] = $username;
            header('Location: ../post/list_post.php');
            exit;
        } else {
```

```

        echo "비밀번호가 틀림";
    }
} else {
    echo "사용자 없음";
}
}
?>

<?php include '../header.php'; ?>

<!-- 로그인 폼 -->
<form method="POST">
    사용자명: <input name="username" required><br>
    비밀번호: <input type="password" name="password" required><br>
    <button type="submit">로그인</button>
</form>

```

중간 테스트 - 링크 설정

```

racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1/cykor_web_1$ sudo ln -s /mnt/c/ku/cykor/2_1/web1/cykor_web_1/va
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1/cykor_web_1$

```

사용자명:
 비밀번호:

```

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'ls' at line 1
mysql> USE cykor_web1_db;
Database changed
mysql> SELECT * FROM users;
+----+-----+-----+-----+
| id | username | password | created_at |
+----+-----+-----+-----+
| 1 | racon | $2y$10$s29VH2mqmG1x5ng64OUwembcH28LXsb1aBnAZofc46BjL2dmPSie | 2025-05-21 15:56:57 |
+----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

racon, racon으로 회원가입 테스트 실행 → 저장되어 있는것을 확인할 수 있다

로그인 성공!
 사용자명:
 비밀번호:

로그인 또한 정상적으로 작동한다

게시글 기능

게시글 작성

```
<?php
include '../auth/session.php'; // 로그인 상태 확인
include '../db/db.php';
include '../header.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') { //post로 받은 경우
    $title = $_POST['title']; //제목
    $content = $_POST['content']; //내용
    $user_id = $_SESSION['user_id']; //작성자의 id

    $statement = $connection->prepare("INSERT INTO posts (title, content, user_id) VALUES (?, ?, ?)");
    $statement->bind_param("ssi", $title, $content, $user_id); //각각 문자열, 문자열, int 이다
    $statement->execute(); //쿼리문 실행
    echo "게시글 등록 완료";
    header('Location: list_post.php');
    exit;
}
?>
<h2>글 작성</h2>
<form method="POST">
    제목: <input name="title" required><br>
    내용: <textarea name="content" required></textarea><br>
    <button type="submit">글 등록</button>
</form>
```

게시물 확인

```
<?php
include '../auth/session.php'; // 로그인 확인
include '../db/db.php'; // DB 연결
include '../header.php'; // 공통 헤더
?>

<?php

// 게시글 + 작성자 이름을 가져오는 쿼리
//게시물 id,제목, 내용, 생성 시간등 표시
$sql = "SELECT posts.id, posts.title, posts.content, posts.created_at, users.username
        FROM posts
        JOIN users ON posts.user_id = users.id
        ORDER BY posts.created_at DESC";

//게시글의 작성자 id와 실재 user_id가 같은 행만 가져옴
//작성 시간순 내림차순

$result = $connection->query($sql); //쿼리문 전송 및 저장
```

```

?>

<h2>게시글 목록</h2>

<?php if ($result->num_rows > 0): ?> <!-- 게시물이 존재하는 경우 -->
    <?php while ($row = $result->fetch_assoc()): ?> <!-- 전체 게시물을 반복하며 내용 출력 -->
        <div style="border:1px solid #ccc; padding:10px; margin:10px 0;">
            <h3>
                <a href="view_post.php?id=<?php echo $row['id']; ?>">
                    <?php echo htmlspecialchars($row['title']); ?>
                </a>
            </h3>
            <p><?php echo nl2br(htmlspecialchars($row['content'])); ?></p>
            <p><strong>작성자:</strong> <?php echo htmlspecialchars($row['username']); ?></p>
            <p><small><?php echo $row['created_at']; ?></small></p>
        </div> <!-- content의 내용을 html 태그 방지, nl2br로 줄바꿈 표시 -->
    <?php endwhile; ?>
<?php else: ?>
    <p>게시글이 없습니다.</p>
<?php endif; ?>

```

게시물 상세보기

게시물에 포함된 user id와 현재 로그인된 id가 같은 경우 수정, 삭제가 가능하도록 버튼을 추가한다

```

<?php
include '../auth/session.php'; // 로그인 확인
include '../db/db.php'; // DB 연결
include '../header.php';

$id = $_GET['id']; //문서 번호를 가져온다

$stmt = $connection->prepare("SELECT posts.title, posts.content, posts.created_at, posts.user_id, users.username
FROM posts
JOIN users ON posts.user_id = users.id
WHERE posts.id = ?"); //게시글, 작성자
$stmt->bind_param("i", $id); //게시물 id 바인딩
$stmt->execute(); //sql 쿼리 실행
$result = $stmt->get_result();

if ($row = $result->fetch_assoc()): //실행 결과가 존재하는 경우 row에 정보 저장
    ?>
    <h2><?php echo htmlspecialchars($row['title']); ?></h2> <!-- 특수문자 escape 처리 -->
    <p><?php echo nl2br(htmlspecialchars($row['content'])); ?></p>
    <p><strong>작성자:</strong> <?php echo htmlspecialchars($row['username']); ?></p>
    <p><small><?php echo $row['created_at']; ?></small></p>

    <!-- 로그인한 사용자와 글 작성자가 같을 때만 수정/삭제 버튼 표시 -->
    <?php if ($_SESSION['user_id'] == $row['user_id']): ?>
        <form action="edit_post.php" method="GET" style="display:inline;">
            <input type="hidden" name="id" value="<?php echo $id; ?>">
            <button type="submit">수정</button>
        </form>

        <form action="delete_post.php" method="POST" style="display:inline;" onsubmit="return confirm('정말 삭제하시겠습니까?');">
            <input type="hidden" name="id" value="<?php echo $id; ?>">
            <button type="submit">삭제</button>
        </form>
    <?php endif; ?>

```

```

<?php endif; ?>
<?php else: ?>
    <p>게시글을 찾을 수 없습니다</p>
<?php endif; ?>
<p><a href="list_post.php"><- 목록으로 돌아가기</a></p>

```

게시물 수정

```

<?php
include '../db/db.php';
include '../auth/session.php';
include '../header.php';

if ($_SERVER['REQUEST_METHOD'] === 'GET') {
    // 수정할 게시글 ID를 가져옴
    $id = $_GET['id'];

    // 게시글 정보 조회
    $stmt = $connection->prepare("SELECT title, content, user_id FROM posts WHERE id = ?");
    $stmt->bind_param("i", $id);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($row = $result->fetch_assoc()) {
        // 로그인된 id와 게시물의 id 비교
        if ($_SESSION['user_id'] != $row['user_id']) {
            echo "권한 없음";
            exit;
        }

        $title = $row['title']; //게시물 내용 확인
        $content = $row['content'];
    } else {
        echo "게시글 없음";
        exit;
    }
} elseif ($_SERVER['REQUEST_METHOD'] === 'POST') { //수정된 데이터 post
    // 수정된 데이터를 받아옴
    $id = $_POST['id'];
    $title = $_POST['title'];
    $content = $_POST['content'];

    // 게시글 소유자 확인
    $check = $connection->prepare("SELECT user_id FROM posts WHERE id = ?"); //소유자 정보 확인
    $check->bind_param("i", $id);
    $check->execute();
    $check->bind_result($author_id);
    $check->fetch();
    $check->close();

    if ($_SESSION['user_id'] != $author_id) {
        echo "권한이 없음";
        exit;
    }

    // 수정 쿼리 실행
    $stmt = $connection->prepare("UPDATE posts SET title = ?, content = ? WHERE id = ?"); //게시물 수정 쿼리

```



```

$stmt->bind_param("ssi", $title, $content, $id); //문자, 문자, int
$stmt->execute();

header("Location: view_post.php?id=$id");
exit;
}
?>

<!-- 게시물 수정 폼 -->
<h2>게시글 수정</h2>
<form method="POST">
  <input type="hidden" name="id" value="<?php echo htmlspecialchars($id); ?>">
  제목: <input name="title" value="<?php echo htmlspecialchars($title); ?>" required><br>
  내용: <textarea name="content" required><?php echo htmlspecialchars($content); ?></textarea><br>
  <button type="submit">수정 완료</button>
</form>

```

게시물 삭제

```

<?php
include '../db/db.php';
include '../auth/session.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $post_id = $_POST['id']; //문서 데이터 받아오기

    // 게시물 소유자 확인 (보안)
    $check = $connection->prepare("SELECT user_id FROM posts WHERE id = ?"); //게시글의 user id 받아오기
    $check->bind_param("i", $post_id);
    $check->execute();
    $check->bind_result($author_id);
    $check->fetch();
    $check->close();

    if ($_SESSION['user_id'] == $author_id) {
        $statement = $connection->prepare("DELETE FROM posts WHERE id = ?");
        $statement->bind_param("i", $post_id);
        $statement->execute();
    }

    header('Location: list_post.php');
    exit;
}
?>

```

게시글 작성 테스트

로그인이 되어있는 상태에서 글을 작성

제목:

내용:

게시글 목록

test2
작성자: racon
2025-05-21 17:29:46

test
작성자: racon
2025-05-21 17:26:36

정상적으로 삽입되어 있음을 확인 가능하다

[게시글 목록](#) | [글 작성](#) | [마이페이지](#) | [로그아웃](#)

게시글 목록

[test2](#)
test2_change
작성자: racon
2025-05-21 17:29:46

[test 수정](#)
test 수정
작성자: racon
2025-05-21 17:26:36

마찬가지로 수정, 삭제 또한 정상적으로 작동하고

test2

test2_change

작성자: racon

2025-05-21 17:29:46

[← 목록으로 돌아가기](#)

test2

test2_change

작성자: racon

2025-05-21 17:29:46

[수정](#) | [삭제](#)

[← 목록으로 돌아가기](#)

자식이 작성한 글에서만 수정, 삭제가 작동하도록 설정된 것을 확인할 수 있다.

기타 기능

header

각 페이지를 보다 쉽게 이동하기 위해서 로그인된 경우 위에 각 페이지의 주소를 헤더로 쉽게 이동할 수 있게 설정한다

```
<?php
session_start();
?>

<?php if (isset($_SESSION['user_id'])): ?>
    <nav>
        <a href="/web1/post/list_post.php">게시글 목록</a> |
        <a href="/web1/post/create_post.php">글 작성</a> |
        <a href="/web1/mypage.php">마이페이지</a> |
        <a href="/web1/auth/logout.php">로그아웃</a>
    </nav>
    <hr>
<?php endif; ?>
```

마이페이지

```
<?php
include 'auth/session.php';
include 'header.php';
?>

<h2>마이페이지</h2>
```

```
<p>안녕안녕 <?php echo htmlspecialchars($_SESSION['username']); ?> 님!</p>
<p><a href="/web1/auth/logout.php">로그아웃</a></p>
```

admin 기능 추가

```
CREATE DATABASE IF NOT EXISTS cykor_web1_db;
USE cykor_web1_db;

-- user 정보 데이터 저장 테이블 생성
CREATE TABLE IF NOT EXISTS users (
  id INT AUTO_INCREMENT PRIMARY KEY, -- id 저장
  username VARCHAR(50) NOT NULL UNIQUE, -- 사용자 이름 저장, 동일한 이름 저장 불가
  password VARCHAR(255) NOT NULL, -- 비밀번호 저장
  role ENUM('user', 'admin') NOT NULL DEFAULT 'user',
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP -- 계정 생성 시간 저장
);

-- 게시물 데이터 저장 테이블 생성
CREATE TABLE IF NOT EXISTS posts (
  id INT AUTO_INCREMENT PRIMARY KEY, -- 게시물 고유번호, 자동 증가
  title VARCHAR(100) NOT NULL, -- 게시물 제목, 공란 불가
  content TEXT NOT NULL, -- 게시물 내용, 공란 불가
  user_id INT NOT NULL, -- user id
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP, -- 글 작성 시간
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, -- 글 수정 시간, 현재 시간으로
  FOREIGN KEY (user_id) REFERENCES users(id) -- user_id 요소는 항상 user 테이블에 존재해야한다
);
```

sql 테이블을 변경하여 user와 admin을 추가한다

이때 role ENUM('user', 'admin') NOT NULL DEFAULT 'user', 항목에서 기본설정은 user가 되도록 한다

코드 수정

1. login.php 파일 수정 → role 정보도 확인해서 세션에 저장되도록 체크한다

```
<?php
session_start(); // 세션 시작 -> 로그인 성공시 $_SESSION에 사용자 정보가 저장된다
include '../db/db.php'; // DB 연결

if ($_SERVER['REQUEST_METHOD'] === 'POST') { //post를 통해서 데이터를 받아온 경우
  $username = $_POST['username'];
  $password = $_POST['password'];

  $statement = $connection->prepare("SELECT id, password, role FROM users WHERE username = ?"); //쿼리문 준비
  $statement->bind_param("s", $username); //username 정보 문자열로 저장장
  $statement->execute(); //쿼리문 실행
  $statement->store_result(); //실행결과 저장
```

```

if ($statement->num_rows === 1) { //해당 사용자가 1명 존재하는 경우 로그인 절차 확인
    $statement->bind_result($id, $hashedPassword, $role); //위의 select 결과를 저장
    $statement->fetch();

    if (password_verify($password, $hashedPassword)) { //비밀번호 비교
        $_SESSION['user_id'] = $id;
        $_SESSION['username'] = $username;
        $_SESSION['role'] = $role;
        header('Location: ../post/list_post.php');
        exit;
    } else {
        echo "비밀번호가 틀림";
    }
} else {
    echo "사용자 없음";
}
}
?>

<?php include '../header.php'; ?>

<!-- 로그인 폼 -->
<form method="POST">
    사용자명: <input name="username" required><br>
    비밀번호: <input type="password" name="password" required><br>
    <button type="submit">로그인</button>
</form>

```

admin 계정 삽입

admin, admin123

```

<?php
include './db/db.php'; // 경로 확인 필요: 현재 같은 디렉토리면 ./db.php

$username = 'admin';
$raw_password = 'admin123';
$password = password_hash($raw_password, PASSWORD_DEFAULT); // 안전하게 해시
$role = 'admin';

// 중복 방지용 기존 admin 제거 (있을 경우)
$conn->query("DELETE FROM users WHERE username = 'admin'");

// admin 계정 삽입
$stmt = $conn->prepare("INSERT INTO users (username, password, role) VALUES (?, ?, ?)");
$stmt->bind_param("sss", $username, $password, $role);
$stmt->execute();

echo "✅ admin 계정 등록 완료 (비밀번호: admin123)";
?>

```

post 관련 코드를 수정

edit.php 코드에서 문서의 id비교 과정에서 admin 인 경우도 통과되도록 설정

```
<?php
include '../db/db.php';
include '../auth/session.php';
include '../header.php';

if ($_SERVER['REQUEST_METHOD'] === 'GET') {
    // 수정할 게시물 ID를 가져옴
    $id = $_GET['id'];

    // 게시물 정보 조회
    $stmt = $connection->prepare("SELECT title, content, user_id FROM posts WHERE id = ?");
    $stmt->bind_param("i", $id);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($row = $result->fetch_assoc()) {
        // 로그인된 id와 게시물의 id 비교
        if ($_SESSION['user_id'] != $row['user_id'] && $_SESSION['role'] != 'admin') {
            echo "권한 없음";
            exit;
        }

        $title = $row['title']; //게시물 내용 확인
        $content = $row['content'];
    } else {
        echo "게시글 없음";
        exit;
    }
} elseif ($_SERVER['REQUEST_METHOD'] === 'POST') { //수정된 데이터 post
    // 수정된 데이터를 받아옴
    $id = $_POST['id'];
    $title = $_POST['title'];
    $content = $_POST['content'];

    // 게시물 소유자 확인
    $check = $connection->prepare("SELECT user_id FROM posts WHERE id = ?"); //소유자 정보 확인
    $check->bind_param("i", $id);
    $check->execute();
    $check->bind_result($author_id);
    $check->fetch();
    $check->close();

    if ($_SESSION['user_id'] != $author_id && $_SESSION['role'] != 'admin') {
        echo "권한이 없음";
        exit;
    }

    // 수정 쿼리 실행
    $stmt = $connection->prepare("UPDATE posts SET title = ?, content = ? WHERE id = ?"); //게시물 수정 쿼리
    $stmt->bind_param("ssi", $title, $content, $id); //문자, 문자, int
    $stmt->execute();

    header("Location: view_post.php?id=$id");
    exit;
}
?>
```

```

<!-- 게시물 수정 폼 -->
<h2>게시글 수정</h2>
<form method="POST">
  <input type="hidden" name="id" value="<?php echo htmlspecialchars($id); ?>">
  제목: <input name="title" value="<?php echo htmlspecialchars($title); ?>" required><br>
  내용: <textarea name="content" required><?php echo htmlspecialchars($content); ?></textarea><br>
  <button type="submit">수정 완료</button>
</form>

```

delete_post.php 또한 관리자 권한을 추가한다

```

<?php
include '../db/db.php';
include '../auth/session.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $post_id = $_POST['id']; //문서 데이터 받아오기

    // 게시물 소유자 확인 (보안)
    $check = $connection->prepare("SELECT user_id FROM posts WHERE id = ?"); //게시글의 user id 받아오기
    $check->bind_param("i", $post_id);
    $check->execute();
    $check->bind_result($author_id);
    $check->fetch();
    $check->close();

    if ($_SESSION['user_id'] == $author_id || $_SESSION['role'] === 'admin') {
        $statement = $connection->prepare("DELETE FROM posts WHERE id = ?");
        $statement->bind_param("i", $post_id);
        $statement->execute();
    }

    header('Location: list_post.php');
    exit;
}
?>

```

view_post.php 또한 다음과 같이 수정

```

<?php if ($_SESSION['user_id'] == $row['user_id'] || $_SESSION['role'] === 'admin'): ?>

```

admin 계정의 경우에도 수정, 삭제 표시가 되도록 한다

비공개 게시물 기능

db 수정

```

CREATE DATABASE IF NOT EXISTS cykor_web1_db;
USE cykor_web1_db;

-- user 정보 데이터 저장 테이블 생성
CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY, -- id 저장
    username VARCHAR(50) NOT NULL UNIQUE, -- 사용자 이름 저장, 동일한 이름 저장 불가
    password VARCHAR(255) NOT NULL, -- 비밀번호 저장
    role ENUM('user', 'admin') NOT NULL DEFAULT 'user',
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP -- 계정 생성 시간 저장
);

-- 게시물 데이터 저장 테이블 생성
CREATE TABLE IF NOT EXISTS posts (
    id INT AUTO_INCREMENT PRIMARY KEY, -- 게시물 고유번호, 자동 증가
    title VARCHAR(100) NOT NULL, -- 게시물 제목, 공란 불가
    content TEXT NOT NULL, -- 게시물 내용, 공란 불가
    user_id INT NOT NULL, -- user id
    is_private BOOLEAN DEFAULT FALSE, -- 나만보기 기능 추가
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP, -- 글 작성 시간
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, -- 글 수정 시간, 현재 시간으로
    FOREIGN KEY (user_id) REFERENCES users(id) -- user_id 요소는 항상 user 테이블에 존재해야한다
);

```

is_private 항목을 boolean 요소로 추가하여 나만보기 기능을 추가한다

글 작성 변경

- 글을 작성할때 나만보기 기능을 추가한다

```

<?php
include '../auth/session.php'; // 로그인 상태 확인
include '../db/db.php';
include '../header.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') { //post로 받은 경우
    $title = $_POST['title']; //제목
    $content = $_POST['content']; //내용
    $user_id = $_SESSION['user_id']; //작성자의 id

    $statement = $connection->prepare("INSERT INTO posts (title, content, user_id, is_private) VALUES (?, ?, ?, ?)");
    $statement->bind_param("ssii", $title, $content, $user_id, $is_private); //각각 문자열, 문자열, int 이다
    $statement->execute(); //쿼리문 실행
    echo "게시글 등록 완료";
    header('Location: list_post.php');
    exit;
}
?>
<h2>글 작성</h2>
<form method="POST">
    제목: <input name="title" required><br>
    내용: <textarea name="content" required></textarea><br>
    <label><input type="checkbox" name="is_private" value="1"> 나만 보기</label><br>

```



```
<button type="submit">글 등록</button>
</form>
```

상세보기 화면에서도 해당 설정을 추가한다

```
if ($row = $result->fetch_assoc())://실행 결과가 존재하는 경우 row에 정보 저장
    if ($row['is_private'] && $_SESSION['user_id'] != $row['user_id'] && $_SESSION['role'] != 'admin') { //비공개 게시물은 a
        echo "<p>이 게시글은 비공개입니다.</p>";
        exit;
    }
}
```

게시글 수정 화면에서도 해당 기능을 설정할 수 있도록 코드를 추가한다

```
<?php
include '../db/db.php';
include '../auth/session.php';
include '../header.php';

if ($_SERVER['REQUEST_METHOD'] === 'GET') {
    // 수정할 게시글 ID를 가져옴
    $id = $_GET['id'];

    // 게시글 정보 조회
    $statement = $connection->prepare("SELECT title, content, user_id, is_private FROM posts WHERE id = ?");
    $statement->bind_param("i", $id);
    $statement->execute();
    $result = $statement->get_result();

    if ($row = $result->fetch_assoc()) {
        // 로그인된 id와 게시물의 id 비교
        if ($_SESSION['user_id'] != $row['user_id'] && $_SESSION['role'] != 'admin') {
            echo "권한 없음";
            exit;
        }

        $title = $row['title']; //게시물 내용 확인
        $content = $row['content'];
    } else {
        echo "게시글 없음";
        exit;
    }
} elseif ($_SERVER['REQUEST_METHOD'] === 'POST') { //수정된 데이터 post
    // 수정된 데이터를 받아옴
    $id = $_POST['id'];
    $title = $_POST['title'];
    $content = $_POST['content'];
    $is_private = isset($_POST['is_private']) ? 1 : 0;

    // 게시글 소유자 확인
    $check = $connection->prepare("SELECT user_id FROM posts WHERE id = ?"); //소유자 정보 확인
    $check->bind_param("i", $id);
    $check->execute();
    $check->bind_result($author_id);
```

```

$check->fetch();
$check->close();

if ($_SESSION['user_id'] != $author_id && $_SESSION['role'] != 'admin') {
    echo "권한이 없음";
    exit;
}

// 수정 쿼리 실행
$stmt = $connection->prepare("UPDATE posts SET title = ?, content = ?, is_private = ? WHERE id = ?"); //게시물 수
$stmt->bind_param("ssii", $title, $content, $is_private, $id);
$stmt->execute();

header("Location: view_post.php?id=$id");
exit;
}
?>

<!-- 게시물 수정 폼 -->
<h2>게시글 수정</h2>
<form method="POST">
    <input type="hidden" name="id" value="<?php echo htmlspecialchars($id); ?>">
    제목: <input name="title" value="<?php echo htmlspecialchars($title); ?>" required><br>
    내용: <textarea name="content" required><?php echo htmlspecialchars($content); ?></textarea><br>
    <label>
        <input type="checkbox" name="is_private" <?php echo ($is_private ? 'checked' : ''); ?>>
        나만 보기
    </label><br>
    <button type="submit">수정 완료</button>
</form>

```

docker,doecker-compose 설정

Dockerfile

admin 계정 생성을 위해 [entry.sh](#) 파일을 추가하였다

```

FROM php:8.1-apache

RUN apt-get update \
    && apt-get install -y default-mysql-client \
    && docker-php-ext-install mysqli

COPY . /var/www/html/

COPY entry.sh /entry.sh
RUN chmod +x /entry.sh

ENTRYPOINT ["/entry.sh"]

```

```
#!/bin/bash

echo "Waiting for MySQL to be ready..."

# 포트 3306에 연결 가능할 때까지 기다림
until mysqladmin ping -h db -u root -p"$MYSQL_ROOT_PASSWORD" --silent; do
    echo "Still waiting..."
    sleep 2
done

echo "Running init_admin.php..."
php /var/www/html/init_admin.php

echo "Starting Apache..."
exec apache2-foreground
```

entry.sh 같은 경우 init_admin.php가 자동으로 한번 실행될 수 있도록 하여 위의 admin생성 php 파일을 실행해 admin계정을 추가한다

docker-compose.yml

```
version: '3.8'

services:
  web:
    build: .
    ports:
      - "8080:80"
    volumes:
      - ../var/www/html
    depends_on:
      - db

  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: racon123!
      MYSQL_DATABASE: cykor_web1_db
    ports:
      - "3307:3306"
    volumes:
      - ./schema:/docker-entrypoint-initdb.d

volumes:
  db_data:
```

비밀번호는 다음과 같이 사전에 설정해줬다

```
MYSQL_ROOT_PASSWORD=racon123!
```

```
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1/cykor_web_1$ docker compose up --build
```

```
docker compose down
```

```
racon@DESKTOP-H67TV7J:/mnt/c/ku/cykor/2_1/web1/cykor_web_1$ docker compose up --build
WARN[0000] /mnt/c/ku/cykor/2_1/web1/cykor_web_1/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove
ential confusion
[+] Running 12/12
 ✓ db Pulled
 ✓ 20e4dcae4c69 Pull complete
 ✓ 1c56c3d4ce74 Pull complete
 ✓ e9f03a1c24ce Pull complete
 ✓ 68c3898c2015 Pull complete
 ✓ 6b95a940e7b6 Pull complete
```

테스트

[게시글 목록](#) | [글 작성](#) | [마이페이지](#) | [로그아웃](#)

마이페이지

안녕안녕 admin 님!

[로그아웃](#)

정상적으로 admin 계정을 통해 로그인이 가능하다

[게시글 목록](#) | [글 작성](#) | [마이페이지](#) | [로그아웃](#)

게시글 목록

[test](#)

test

작성자: racon

2025-05-25 00:12:39

[test](#)

test

작성자: admin

2025-05-25 00:08:48

게시글 목록 또한 정상적으로 보인다

test

test

작성자: racon

2025-05-25 00:12:39

[수정](#) | [삭제](#)

[← 목록으로 돌아가기](#)

상세 페이지의 경우 현재 racon 계정으로 접속했기 때문에 정상적으로 보인다

이 게시글은 비공개입니다.

racon2 계정으로 접속하게 되면 다음과 같이 비어있는 것을 확인할 수 있다