

CYKOR 웹 1주차 개발 일지

2025350003 최동주

5/15: 기초 세팅 및 가입기능 구현

개발환경은 xampp와 mysql extension을 활용한 vscode이다.

우선 cykor라는 데이터베이스를 만들어주었고, 인터넷을 뒤지며 배운 문법으로 간단한 코드를 작성해보았다.

```
USEcykor;
CREATE TABLEuserinfo (
  id VARCHAR(100),
  VARCHAR(100)
);
--예제 입력
INSERT INTOuserinfo(id,pw) VALUES('dongjoo','1122');
```

이후 extension에 들어가보니 해당 값들이 잘 들어가져있었다. commit을 하겠다.

프론트엔드 개발은, 우선 html로 전체적인 인터페이스 및 페이지들을 제작한 뒤, form 태그를 통해 php파일을 불러올까 한다.

우선은 login, main, signin 페이지를 다음과 같이 구현했다.

```
<!DOCTYPE html>
<html>
<head>
  <title>CRUD</title>
</head>
<body>
  <div>
    <span><a href="main.html">main</a></span>
    <span><a href="login.html">login</a></span>
    <span><a href="signin.html">sign in</a></span>
  </div>
</body>
</html>
```

해당하는 글자를 클릭시 각 페이지로 넘어갈 수 있도록 하였다. 이후 form을 이용해서 php와 연결이 되도록 할 예정이다. commit을 하겠다.

코드를 돌아보니 span이 없어도 다른 div에 있는게 아니면 가로정렬이

자동으로 되는것을 확인했다(...) span을 없애고 다시 commit했다.

이제 로그인 페이지와 가입하는 페이지의 내용을 만들 예정이다.

우선, 두 페이지 모두 id와 pw 입력이 필요하니, 똑같이 해당 부분을 html로 만들어줄것이다.

```
<div>
    <span>ID</span>
    <input type="text"placeholder="ID 입력"maxlength="100">
</div>
<div>
    <span>PW</span>
    <input type="text"placeholder="PW 입력"maxlength="100">
</div>
<div>
    <button>로그인 하기</button>
</div>
```

위와 같이 코드를 짜주었다. signin.html에는 버튼의 표시를 가입하기로 바꾸었다. h3나 p로 ID와 PW를 적으니 자동으로 줄바꿈을 하길래 span으로 감싸주었다. localhost로 보니 세팅이 잘 돼있어 commit을 하겠다.

이제 signin 페이지에서 입력을 받아 db에 저장이 가능하도록 코드를 짜보았다

```
<?php
$host'localhost';
$username'root';
$password'';
$databas'cykor';
$dbconnnew mysqli($host,$username,$password,$database);
if(!$dbconn) die("DB연결 실패");
//id,pw 입력 받고 변수에 저장
$sid$_POST['sid'];
$spw$_POST['spw'];
//id 중복 체크
$jungbok"SELECT *FROMuserinfo WHEREid ='$sid'";
$result$dbconn->query($jungbok);
if($result->num_rows>0) echo '<script>alert("이미 사용 중인 ID입니
다.");</script>';
```

```

else{
    $signin"INSERT INTOuserinfo (id,pw) VALUES('$sid','$spw')";
    if($dbconn->query($signin) == TRUE) {
        echo '<script>alert("가입이 완료되었습니다.");</script>';
    }
}
$dbconn->close();
?>

```

우선 signin.html에서 id, pw입력에 각각 sid, spw라는 name을 부여해주었다. 이후 php 코드에서 new mysqli를 이용해 cykor db와 연결을 해주었다. 연결하는 함수?가 많아서 찾아보니 그냥 mysql함수는 지원 종료되었고, 향상된 버전의 mysqli를 쓴다고 해 mysqli를 사용했다.

이후 post로 sid spw를 입력받고, 기존의 db에서 중복되는 id가 있으면 찾아 경고문 발생 및 db입력을 안하게 만들었다.

만약 중복이 없을 시, signin 변수를 만들어다 db에 새로운 데이터를 넣게끔 하였고, dbconn의 객체? 기능에 접근하여 만약 signin이 잘 되었다면 가입이 완료되었다고 알람이 뜨게 하였다. ->를 통해 객체의 속성에 접근해 연산을 한다는 개념이 너무 신기했다

최대한 javascript없이 php만 이용하려 했는데, alert는 script만의 기능이라 어쩔 수 없이 사용하게 되었다.

위의 코드를 테스트 해보니, 데이터가 db에 정상적으로 입력이 되긴했다...그러나 웹에서 기존 페이지에 머무르는게 아닌 url이 signin.php로 옮겨져 빈 페이지만 나오게 됐다. 이것을 해결하기 위해 리서칭 해보니, header라는 기능을 사용해 원하는 html로 이동이 가능하다는 것을 알게 되었다. 따라서 header를 사용해 이동하려고 하니, 계속해서 alert와 db가 씹힌다. 아무리 코드를 보고 수정해봐도 계속 똑같이 씹히길래, 결국 내키지는 않았지만 script 내부에 `window.location.href="login.html"` 와 같이 추가하여 php의 header를 통해서가 아닌 스크립트를 통해 이동하게 하였다. 이것을 실행 해보니 잘 되었다. commit을 하겠다.

5/16: 로그인 기능 구현

로그인 기능을 구현하고자 한다. 우선 이전에 signin페이지에서 썼던 코드들을 대거 가져왔고, 거기에 session관련 코드를 조금씩 삽입해 1차적인 코드를 만들었다.

```
<?php
//세션 시작
session_start();
$host'localhost';
$username'root';
$password'';
$databas'cykor';
$dbconnnew mysqli($host,$username,$password,$database);
if(!$dbconn) die("DB연결 실패");
//id,pw 입력 받고 변수에 저장
$lid$_POST['lid'];
$lpw$_POST['lpw'];
//로그인 정보 확인
$login"SELECT *FROMuserinfo WHEREid='$lid' ANDpw='$lpw'";
$result$dbconn->query($login);
//로그인 완료시 세션에 사용자 저장
if($result->num_rows > 0) {
    $_SESSION['userid'] = $lid; // 세션에 사용자 ID 저장
    echo ' <script>alert("로그인이 완료되었습니다.");
window.location.href="main.html";</script>';
} else{
    echo ' <script>alert("아이디 또는 비밀번호가 올바르지 않습니다.");
window.location.href="login.html";</script>';
}

$dbconn->close();
?>
```

위의 코드를 실행시키고 로그인을 해보니, 제대로 alert이 뜨면서 session이 생긴것을 확인했다. 일단은 commit을 하겠다.

이후 로그인 상태를 확인 할 수 있도록 main.html에 로그인 상태가 뜨게끔 하려 한다. 근데 알아보니 html내에서 php코드를 작성할 수는 있어도 실행이 안된다고 하여 main.html을 main.php로 바꾸었다. 이후 조건문과 위에서 저장한 session의 아이디를 이용해

```
<div>
    <?php if(isset($_SESSION['userid'])): ?>
        <p>환영합니다, <?php echo($_SESSION['userid']); ?>님!</p>
    <?php else: ?>
        <p>로그인이 필요합니다.</p>
    <?php endif; ?>
</div>
```

와 같이 코드를 짜니, 세션 정보가 있을때는 인삿말이, 없을때는 로그인이 필요하다는 말이 뜬다. commit을 하겠다.

이후 로그아웃 기능도 구현해보겠다. 우선 각 페이지마다

```
<a href="logout.php">log out</a>
```

같은 코드를 추가해 클릭 가능한 logout 글자를 만들었다.

이후 php의 조건과 세션 정보를 이용해

```
<?php
session_start();
if(isset($_SESSION['userid'])) {
    session_destroy();
    echo ' <script>alert("로그아웃 되었습니다.");
window.location.href="main.php";</script>';
} else{
    echo ' <script>alert("로그인이 되어 있지 않습니다.");
window.location.href="login.html";</script>';
}
exit();
?>
```

만약 로그인이 되어있다면 로그아웃을 시키고 메인화면으로, 아니라면 로그인이 안되어있다고 안내 후 로그인 페이지로 이동시키게 하였다. 이후 실행시켜보니 잘 작동이 된다. commit을 하겠다.

이제 본격적으로 crud기능을 구현해보고자 한다. 우선은, 과제 조건에 따라 새로운 테이블을 생성했다.

```
CREATE TABLE board (
title VARCHAR(100),
writer VARCHAR(100),
    context VARCHAR(100)
);
```

생각해보니 글마다 고유 아이디도 있으면 좋을 것 같아 테이블 삭제 후 다시 만들었다.

```
CREATE TABLE board (
    uid INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100),
    writer VARCHAR(100),
    context VARCHAR(1000)
);
```

자동으로 index의 값을 늘려주는 코드를 auto_increment를 통해 작성했다. 또한 이왕 다시 만드는 겸 context의 크기를 1000으로 늘려줬다.

이제 db에 존재하는 게시글들을 메인 페이지에 나오게끔 해야한다. 찾아보니 fetch?기능을 사용해서 하는것 같다. 조금 더 알아보니 fetch assoc을 통해서 컬럼명중 원하는 값을 뽑을 수 있는것 같다. 이것을 참고하여 main.html에서 다음과 같이 코드를 짰다.

```
<div>
    <?php if($result->num_rows>0){
        while($row=$result->fetch_assoc()) {
            $title=$row['title'];
            echo "<a href='board.php?title=".urlencode($title)."'>".$title."</a>";
        }
    }else{
        echo"게시글이 없습니다.";
    }

    ?>
</div>
```

실행시켜보니, 로그인 현황 아래에 클릭 가능한 글씨로 나온다. 내가 생각한 알고리즘은 다음과 같다. 만약 해당하는 게시글 클릭시, board.php에 ? title의 값까지 함께 전달을 하고, 전달 받은 php가 해당 title값을 토대로 db에서 해당 데이터들을 찾아 필요한 정보들을 출력하게끔 할 것이다. 이 부분을 조사해보니 php로 보낼 때 \$title을 그대로 보내면 특수기호들이나 공백에 따라 깨질수도 있다고 해 urlencode를 붙여주었다. commit하겠다.

이제 게시글 작성 버튼을 만들 것이다. 페이지 제일 하단에 나오게 할 것이며, 우선 기능 구현 후 로그인 유무에 따른 접근을 구현할 것이다.

일단은 클릭버튼을 만들어주었다.

```
<div>
    <form action="write.php"method"post">
        <button>글 작성하기</button>
    </form>
</div>
```

이전과 같이 form을 통해 php파일로 이동하여 작동하게끔 할 것이다.

세션과 php, script를 이용해 만약 로그인이 안되어있다면 alert와 함께 login페이지로 이동시키는 코드도 작성했다.

```
echo          "<script>alert('로그인이          필요합니다.');"
window.location.href='login.html';</script>";
exit;
```

```
<div>
    <form action="write2.php"method"post">
        <span>글 제목</span>
        <input type="text"name"title"maxlength="100"><br>
        <span>글 내용</span>
        <input type="text"name"context"maxlength="1000"><br>
        <button>게시하기</button>
    </form>
</div>
```

해당 코드에서서는 글 제목과 글 내용을 입력받고, button 클릭시 form을 통해 write2.php로 넘어가진다.

write2.php에서는 이전 write1.php에서 입력받은 title과 context, 그리고 현재 세션의 id를 토대로

```
$title=$_POST['title'];
$context=$_POST['context'];
$writer=$_SESSION['userid'];
```

세가지 변수를 만들어주고, 이전에 signin할때 사용해줬던, php로 spl에 데이터를 넣는 코드를 가져와 조금 변형시켜 적었다.

```
$create"INSERT          INTOboard          (title,          writer,          context)
```

```
VALUES('$title','$writer','$context');"
if($dbconn->query($create) == TRUE) {
    echo    '<script>alert("게시글    작성이    완료되었습니다.");window.location.href="main.php";</script>';
} else{
    echo    "<script>alert('게시물    등록에    실패했습니다');
history.back();</script>";
}
```

실행시켜보니, 게시글을 작성하고 나면 alert를 통해 알람이 뜨고, 이후 메인화면으로 이동이 된다. 이때 이전에 작성한 글이 정상적으로 출력되는 것을 확인 할 수 있었다!
다만 게시글들이 세로정렬이 아닌 가로정렬로 나오기에, 출력코드에
을 붙여 줄 넘기기가 되도록 설정했다.

```
echo    "<a
href='board.php?title=".urlencode($title)."'>".$title."</a><br>";
```

이후 다시 실행시켜보면

[main](#) [login](#) [sign in](#) [log out](#)

로그인이 필요합니다.

[cykor 화이팅](#)

[L L](#)

[화이팅](#)

글 작성하기

잘 나오는 것을 확인할 수 있다!! 기쁜마음으로 commit을 하겠다!

5/18: 게시판 관련 개발

우선 게시글이 잘 나오는 것을 확인 했으니, 간단하게 게시글 목록 위에 "게시글 목록"이라는 글귀를 추가하기로 했다

```
<br><h3>게시글 목록</h3>
```

이제 게시글 조회 기능을 구현하면 된다.

그리고 생각을 해보니, 게시글을 보여주는 php파일로 제목을 보내는 것보다는 고유 아이디를 보내는 것이 낫다고 판단했다. 왜냐하면 이런 경우 제목을 굳이 인코딩해서 보내는 비효율과, 같은 제목의 글이 있을 때 생길 수 있는 미연의 버그를 예방 할 수 있다는 생각이 들었기 때문이다. 따라서 코드를


```

while($row$result->fetch_assoc()) {
    $title$row['title'];
    $unid$row['unid'];
    echo "<a
href='board.php?unid=".$unid."'>".$title."</a><br>";
}

```

과 같이 수정했다. 웹에서 확인해보니,

<http://localhost/cykor/board.php?title=5> 처럼 잘 나온다. 커밋을 하겠다.

이제 board.php로 가겠다. 이 php에서는 전달받은 unid를 토대로 db에서 해당하는 게시글을 가져와 내용을 출력하는 방식으로 하고자 한다.

```

$unid=$_GET['unid'];
echo "$unid"
?>

```

우선 해당 코드를 통해 php파일에서 unid를 잘 가져오는 것을 확인했다.

이제 이것을 활용해 table에서 unid의 값을 토대로 내용을 불러와 출력하는 코드를 작성하였다.

```

<div>
    <?php
    $page"SELECT *FROMboard WHEREunid =$unid";
    $result$dbconn->query($page);
    $row$result->fetch_assoc();
    echo "<h4>".$row['title']."</h4>";
    echo "<p>".$row['context']."</p>";
    ?>
</div>

```

돌려보니,

[main](#) [login](#) [sign in](#) [log out](#)

cykor 화이팅

정회원 가즈아~

처럼 잘 출력되는 것을 확인하였다. commit하겠다

이제 세션 아이디와 writer의 값이 일치할 때만 수정 및 삭제 버튼이 나오도록 조정해보겠다. 우선 메인php로 가서 내가 세션 id를 어떻게 출력했었는지 확인해보았다.

이후 간단한 코드를 짜보았다.

```
<?php
    if($_SESSION['userid'] == $row['writer']) {
        echo "수정 삭제";
    }

?>
```

근데 이렇게 하니, userid가 있을때는 잘 작동하지만 로그인 안되어있을 때는 Warning: Undefined array key "userid" in C:\xampp\whtdocs\cykor\board.php on line 41처럼 뜬다. 왜이런가 찾아보니, 말 그대로 userid가 정의되지 않아서 그랬었다. 따라서 main에서 했던것처럼 isset함수를 이용해 다시 짜보았다. 그러니 깔끔하게 사라졌다. 우선은 commit을 하겠다.

일단 버튼을 생성한 후, 삭제부터 구현하기 위해 대충 코드를 짜보았다.

```
if(isset($_SESSION['userid'])){
    if($_SESSION['userid'] == $row['writer']) {
        echo "<button>수정</button>";
        echo "<button>삭제</button>";
        onclick="\location.href='delete.php?unid=$unid'\">>삭제</button>";
    }
}
```

"가 이스케이프 되는거를 눈채못해서 저 한줄 쓰는데 피나 애를 먹었다. 암튼 이것이 클릭 되었을 때 unid와 함께 delete.php로 전달 되는 것을 확인했다. 이제 delete.php에 <?php

```
session_start();
$host'localhost';
$username'root';
$password'';
$databse'cykor';
$dbconnnew mysqli($host,$username,$password,$database);
if(!$dbconn) die("DB연결 실패");
//게시글 가져오기
$list"SELECT unid, title, context, writer FROM board";
$result$dbconn->query($list);
$unid$_GET['unid'];
$delete = "DELETE FROM board WHERE unid = $unid";
```

```
echo          '<script>alert("게시글이          삭제되었습니다.");
window.location.href="main.php";</script>';
?>
```

다음과 같이 코드를 짜보고 돌려보았는데, alert까진 뜨지만 db에서는 아무것도 삭제되지 않았다. 왜 그런가 생각해보니 delete를 dbconn의 쿼리에 넣지 않아서 그랬다...

```
$unid=$_GET['unid'];
$delete"DELETE FROMboard WHEREunid = $unid";
if($dbconn->query($delete) == TRUE) {
    echo          '<script>alert("게시글이          삭제되었습니다.");
window.location.href="main.php";</script>';
} else{
    echo          "<script>alert('게시글          삭제에          실패했습니다');
history.back();</script>";
}
```

다음과 같이 수정하고 돌려보니, 정상적으로 게시글이 삭제되며 작동이 된다. commit을 하겠다!

5/19: 게시글 수정 구현

이제 수정 기능을 구현하면 기본적인 CRUD 기능 구현은 끝난다. 우선 update.php를 만들어준 후, 글쓰기 템플릿이 있는 write1.php을 복붙해다가 가져왔다.

이전 write와는 다르게 한 페이지에서 끝내고 싶다는 생각이 들었다. 그래서 우선은 리서칭을 한 뒤, 나름의 방법을 찾은것 같다. 우선은 아래에 html부분을 다음과 같이 post 메소드로 수정해줬다.

```
<form action=""method"post">
    <span>글 제목</span>
    <input type"text"name"title"maxlength="100"><br>
    <span>글 내용</span>
    <input type"text"name"context"maxlength="1000"><br>
    <button type= "submit">수정하기</button>
</form>
```

이렇게 했을 때 위의 php문에서 request_method를 통해 post를 확인할 수 있다고 확인했다.

이후

```
if($_SERVER["REQUEST_METHOD"] == "POST") {
    $title$_POST['title'];
```

```

    $context$_POST['context'];
    $update"UPDATEboard SETtitle='$title', context='$context'
WHEREunid='$unid'";
    $result$dbconn->query($update);
    if($dbconn->query($result)==TRUE) {
        echo '<script>alert("게시글 수정이 완료되었습니다.");window.location.href="board.php?unid=$unid";</script>';
    } else{
        echo '<script>alert("게시글 수정에 실패하였습니다.");window.location.href="board.php?unid=$unid";</script>';
    }
}
}

```

이런 형식으로 업데이트 가능하게끔 코드를 작성해서 실행을 했더니...

```

Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL
syntax; check the manual that corresponds to your MariaDB server
version for the right syntax to use near '1' at line 1 in
C:\xampp\Whtdocs\Wcykor\update.php:20 Stack trace: #0
C:\xampp\Whtdocs\Wcykor\update.php(20): mysqli->query('1') #1 {main}
thrown inC:\xampp\Whtdocs\Wcykor\update.phpon line20

```

와 같은 에러가 났다(???) 무엇이 문제인가 보니 if문에 \$update 대신 \$result를 넣어 생긴 오류였다. 이것을 수정해주니 db가 수정되긴 했지만, 또 오류가 발생했다.

```

Fatal error: Uncaught mysqli_sql_exception: Unknown column '$unid' in 'where clause'
in C:\xampp\Whtdocs\Wcykor\board.php:33 Stack trace: #0
C:\xampp\Whtdocs\Wcykor\board.php(33): mysqli->query('SELECT * FROM b...') #1
{main} thrown in C:\xampp\Whtdocs\Wcykor\board.php on line 33

```

이건 또 뭔가 하니 위의 echo에서 \$unid가 변수로 인식이 되지 않아 생긴 오류같았다. 이것을 수정하고 돌려보니 오류 없이 제대로 작동했다. 이제 커밋을 하겠다~~

이제 CRUD 기능은 전부 구현했다. 이제는 무엇을 구현할까 고민을 해보아야겠다.

5/25: 개발 구조화 및 추가 기능 고안

구현 이후 과제 제출 형식 및 도커 compose가 가능하도록 환경을 조금 바꾸어보고자 한다. 일단 assignment_template의 형식을 참고하였으며, 도커로 실행가능하고자 도커에 대한 아주 약간의 공부를 하려고 한다. 우선 형식은

맞추었으니 commit을 하였다.

5/26: 도커 환경 설정

도커를 통해 돌리기 위해 compose와 빌드를 해보니, db-1 | 2025-05-26 00:19:18+00:00 [Note] [Entrypoint]: /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/init.sql

db-1 | ERROR 1049 (42000) at line 1: Unknown database 'cykor' 라는 에러가 떴다. 따라서 init에 CREATE DATABASE IF NOT EXISTS cykor; 라는 문장을 추가하고 다시 돌려보니 이 외에도 여러 에러가 떴다. compose의 정보와 맞게 수정하고, init.sql 내부 내용(자동으로 테이블 생성)등을 하고 다시 돌려보니 또 오류가 떴다.

그러던 중 exampleuser에게 권한이 주어지지 않는 것이 치명적인 오류라는 점을 발견하였고, init.sql에

```
GRANT ALL PRIVILEGES ON cykor.* TO 'exampleuser'@'%';  
FLUSH PRIVILEGES;
```

을 추가해주고, 컨테이너를 다시 빌드하니 정상적으로 돌아간다!!!! 이제 commit을 하겠다.