



Term-Project

야구 커뮤니티 사이트

과목명	웹프로그래밍(창업)
분반	1 분반
교수	문일영
학번	2018136045
이름	박 형 진
제출일	2023년 6월 7일 수요일

목 차

I. 서론

- 1. 프로젝트 목표 및 요구사항 1
- 2. 주제 및 주제 선정 동기 1

II. 본론

- 1. 'Baseball Community'의 주요 기능소개 2
 - 1) 오늘의 프로야구 경기 정보 제공 기능 2
 - 2) 회원가입, 회원 정보수정 기능(응원팀 추가) 3
 - 3) 로그인, 로그아웃 기능 4
 - 4) 회원간 쪽지전송 기능 5
 - 5) 회원간 쪽지 답장, 삭제 기능 6
 - 6) 게시글 쓰기, 수정, 삭제 기능 7
 - 7) 게시글에 대한 댓글 쓰기 기능 9
 - 8) 자신이 남긴 댓글만 삭제할 수 있는 기능 10
 - 9) 응원하기 기능 11
- 2. 주요 기능을 위한 DB 구조 설계 12
 - 1) members 테이블 12
 - 2) message 테이블 13
 - 3) board 테이블 14
 - 4) comments 테이블 15
 - 5) team_cheer 테이블 16
- 3. 주요 기능 구현 방법 17
 - 1) 오늘의 프로야구 경기 정보 제공 기능 17
 - 2) 회원가입, 회원 정보수정 기능(응원팀 추가) 18
 - 3) 게시글에 대한 댓글 쓰기 기능 19
 - 4) 게시판과 댓글에 글쓴이의 팀 제공 기능 20
 - 5) 자신이 남긴 댓글만 삭제할 수 있는 기능 21
 - 6) 응원하기 기능 22

III. 결론

- 1. 추후 개발 계획 23
- 2. 소감 23

I. 서론

1. 프로젝트 목표 및 요구사항

본 보고서는 2023학년도 한국기술교육대학교 컴퓨터 공학부 웹프로그래밍(창업) 교과목에서 요구된 팀프로젝트에 대하여, "PHP로 프로그래밍 된 웹사이트"를 상세히 알리는 것을 목적으로 한다. 본 프로젝트의 목표는 학기 중에 배운 PHP 프로그래밍을 통해 사용자에게 여러 서비스를 제공하는 웹사이트를 구축하는 것이다.

프로젝트 요구사항

- XAMPP(잼프)를 활용한 PHP 프로그래밍으로 자신만의 사이트를 구축한다.
- 사용자에게 여러 서비스를 제공해주는 기능들을 제공한다.
- 서비스를 이용할 수 있도록 사용자 인터페이스 구현한다.
- 각 기능 구현에 필요한 데이터를 관리하기 위해 데이터베이스를 설계하여 연동한다.

2. 주제 및 주제 선정 동기

본 프로젝트의 웹사이트 주제는 야구를 좋아하는 사람들을 위한 웹사이트('Baseball Community')이다. 야구를 좋아하는 편이라 평소에 야구 커뮤니티를 자주 이용해왔다. 그러나 많은 커뮤니티 사이트가 있음에도 불구하고, 소통만을 위한 게시판이 주된 기능일 뿐 추가된 기능이 있는 사이트는 많이 찾아보기 힘들다. 그래서 쪽지, 응원하기 기능이 포함된 종합적인 야구 커뮤니티 사이트를 개발하고자 한다. 회원 정보에 응원팀 정보까지 추가하여 게시판 활동 시 응원팀까지 알 수 있도록 하여 소속감을 더해주고자 한다. 이렇게 개발된 사이트를 통해 본 웹사이트 이용자는 더 큰 즐거움을 누릴 수 있을 것이다.



II. 본론

1. 'Baseball Community'의 주요 기능소개

1) 오늘의 프로야구 경기 정보 제공 기능

The screenshot shows the 'Baseball Community' website. The header includes a logo and navigation links: Home, Message, Board, Cheer Up, Sign Up, and Login. The main content area is titled 'KBO 경기' (KBO Games) and displays a table of today's games. Below the table, there is a section for '최근 게시물' (Recent Posts) and a footer with contact information and a copyright notice.

Team 1	Time	Team 2
KT Wiz SP: Bae Je-seong	6:30pm Gwangju	Kia Tigers SP: Medina
Lotte Giants SP: Barnes	6:30pm Seoul-Jamsil	LG Twins SP: Plutko
Doosan Bears SP: Alcántara	6:30pm Changwon	NC Dinos SP: Fedde
Samsung Lions SP: Yang Chang-seop	6:30pm Incheon-Munhak	SSG Landers SP: Baek Seung-geon
Kiwoom Heroes SP: Jurado	6:30pm Daejeon	Hanwha Eagles SP: Moon Dong-ju

최근 게시물

KBO(<https://www.koreabaseball.com>)
한국 프로야구

2018136045 박정원
이메일 주소 : bark000104@koreatech.ac.kr | Gd : hyeongjinnnn

야구 커뮤니티 사이트인 만큼 오늘의 경기 정보를 알 수 있도록 오늘의 정보 데이터를 가져와서 웹사이트에 메인 화면에 출력하는 기능을 추가했다. 이용자는 홈페이지에 들어오면 오늘 예정 중인 경기 또는 하고 있다면 실시간으로 진행 중인 경기 정보를 알 수 있다.

2) 회원가입, 회원 정보수정 기능(응원팀 추가)



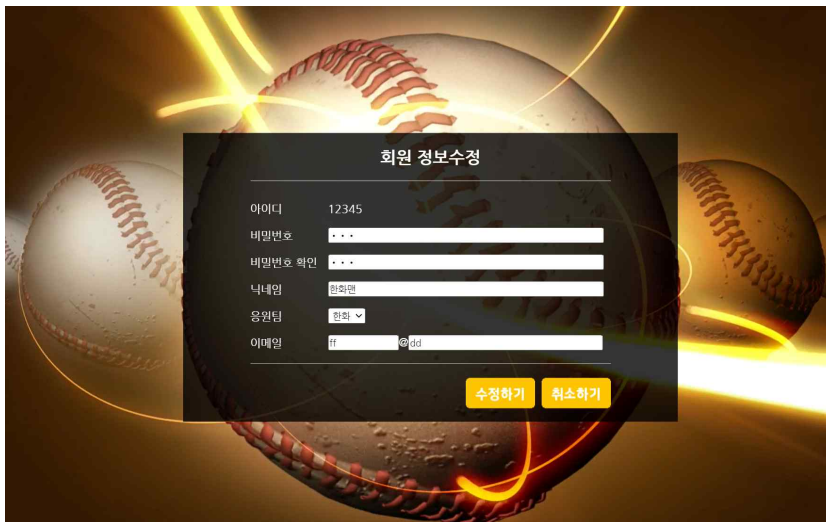
회원가입 폼은 야구공과 빛나는 선이 배경으로, 중앙에 검은색 박스 위에 흰색 텍스트와 입력 필드가 배치되어 있습니다. 제목은 '회원 가입'입니다. 필드에는 아이디, 비밀번호, 비밀번호 확인, 닉네임, 응원팀(한화 선택), 이메일이 포함되어 있습니다. 오른쪽에는 중복확인 버튼이, 아래에는 가입하기와 취소하기 버튼이 있습니다.

회원 가입	
아이디	<input type="text"/>
비밀번호	<input type="password"/>
비밀번호 확인	<input type="password"/>
닉네임	<input type="text"/>
응원팀	한화 ▼
이메일	<input type="text"/> @ <input type="text"/>

중복확인

가입하기 취소하기

회원가입은 응원팀을 선택할 수 있도록 하여 소속감을 줄 수 있도록 하였다. 현재 프로야구 구단 10개 팀 중 선택할 수 있도록 구성하였다.



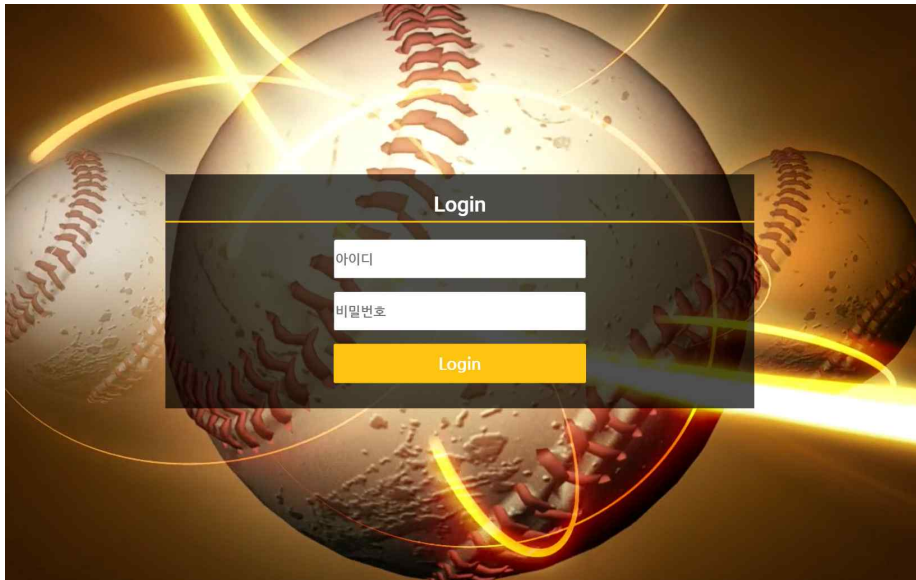
회원 정보수정 폼은 야구공과 빛나는 선이 배경으로, 중앙에 검은색 박스 위에 흰색 텍스트와 입력 필드가 배치되어 있습니다. 제목은 '회원 정보수정'입니다. 필드에는 아이디(12345), 비밀번호, 비밀번호 확인, 닉네임(한화맨), 응원팀(한화 선택), 이메일이 포함되어 있습니다. 오른쪽에는 수정하기와 취소하기 버튼이 있습니다.

회원 정보수정	
아이디	12345
비밀번호	<input type="password"/>
비밀번호 확인	<input type="password"/>
닉네임	한화맨
응원팀	한화 ▼
이메일	<input type="text"/> @ <input type="text"/>

수정하기 취소하기

회원 정보수정도 마찬가지로 자신의 응원팀까지 수정할 수 있도록 구성하였다.

3) 로그인, 로그아웃 기능

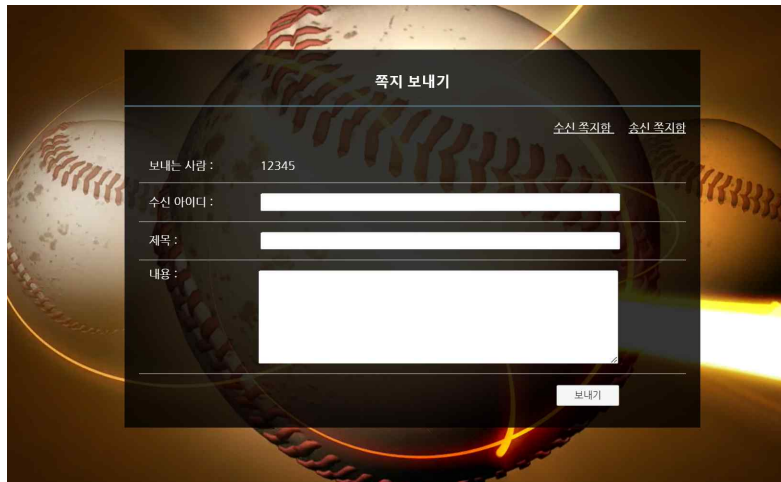


회원가입 이후 로그인이 가능하도록 구현하였다. 가입한 아이디와 비밀번호를 입력한 후 Login 버튼을 누르면 로그인이 완료된다.



로그인이 완료된 상태면 Sign Up과 Login이 있던 헤더 자리에 Logout과 Edit Profile이 생긴다. 여기서 Logout을 누르면 로그아웃을 누르면 로그아웃이 완료된다.

4) 회원간 쪽지전송 기능



쪽지 보내기

[수신 쪽지함](#) [송신 쪽지함](#)

보내는 사람: 12345

수신 아이디:

제목:

내용:



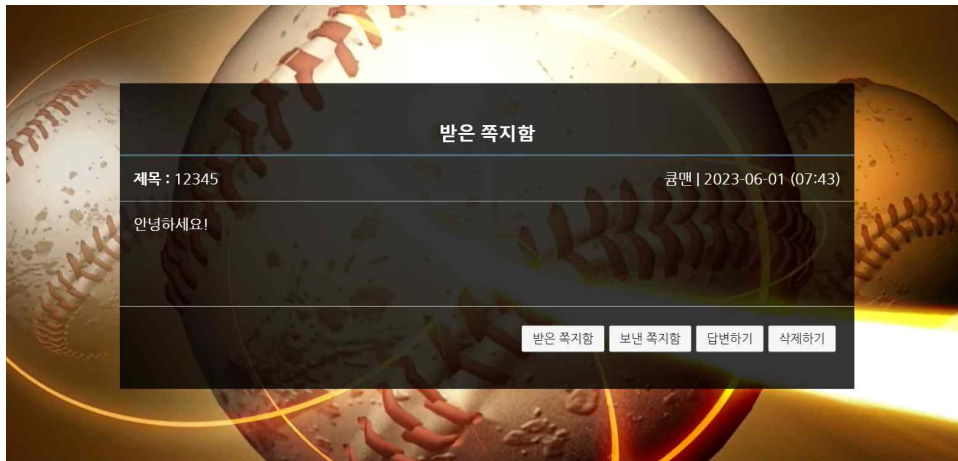
받은 쪽지함

번호	제목	보낸이	등록일
1	12345	쿵맨(abc)	2023-06-01 (07:43)

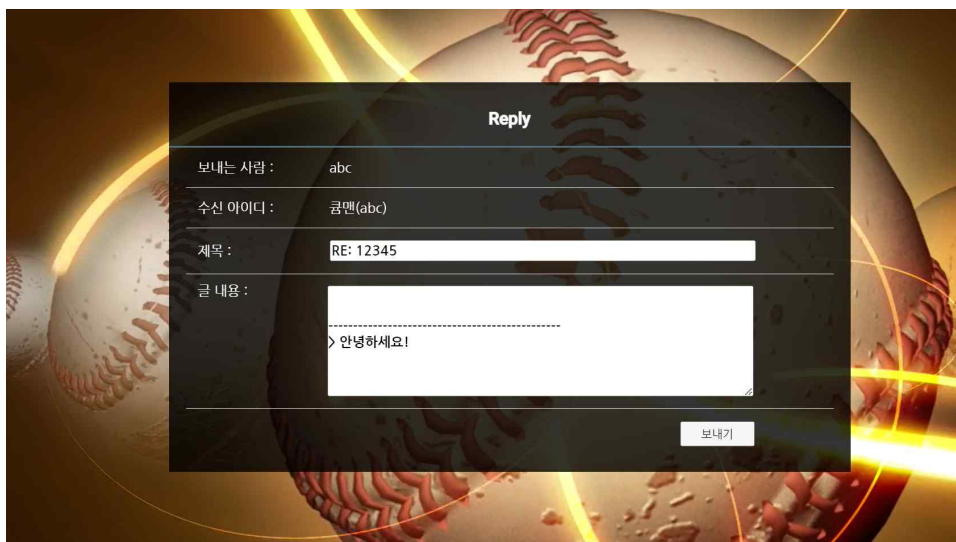
1

회원 간의 공개되지 않는 개인적 소통 공간을 위한 쪽지전송 기능이다. 전송된 쪽지는 보낸 쪽지함에서 확인할 수 있고 전송받은 쪽지는 받은 쪽지함에서 확인할 수 있다.

5) 회원간 쪽지 답장, 삭제 기능



쪽지함에 있는 쪽지 열람 시 우측 하단의 답변하기를 통해 쪽지를 보낸 사람에게 답장할 수 있다. 삭제하기 버튼을 통해 현재 열람 중인 쪽지를 삭제할 수 있다.

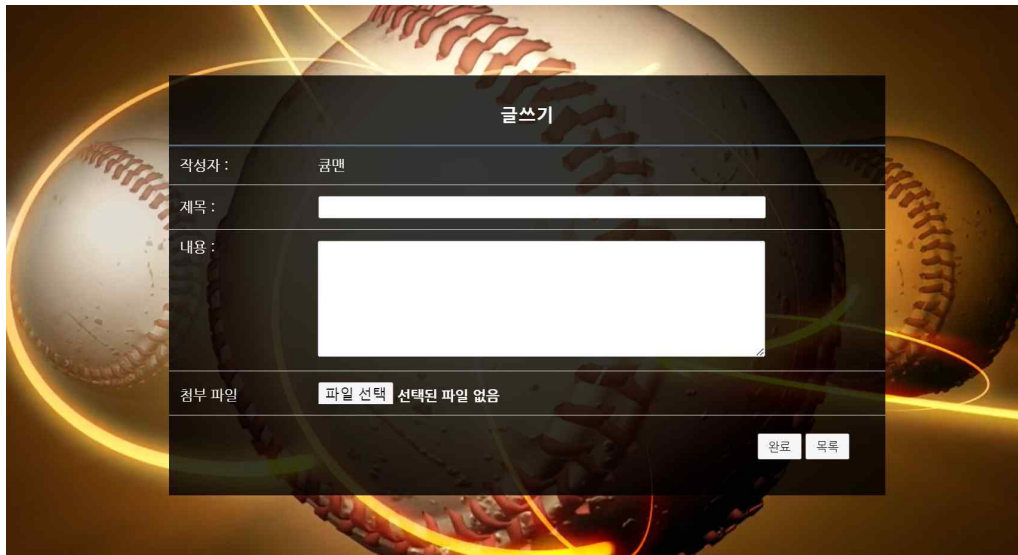


기본으로 제목과 글 내용이 받은 쪽지의 내용을 가지고 자동으로 채워지는 것을 확인할 수 있다.

6) 게시글 쓰기, 수정, 삭제 기능



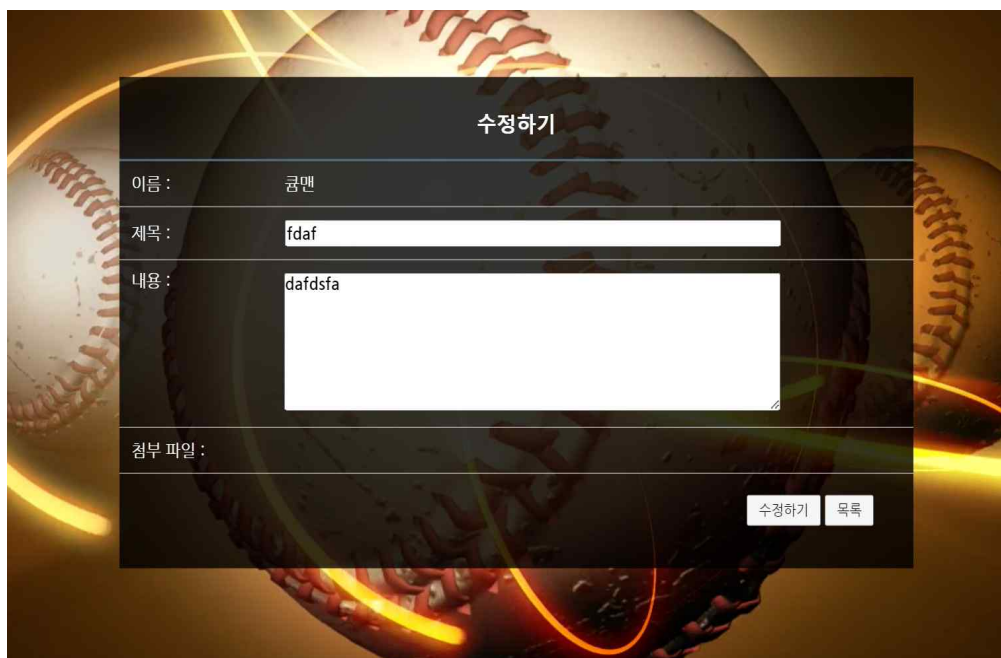
Header에서 board를 클릭하면 게시판 페이지로 이동한다. 본 사이트 게시판의 추가적인 특징은 글쓴이의 정보에 팀까지 보여준다는 점이다. 우측 하단의 글쓰기 버튼을 클릭하면 아래의 사진의 페이지로 이동한다.



이처럼 글을 쓸 수 있는 form이 나타나고, 입력 후 완료 버튼을 누르면 게시글이 작성된다.

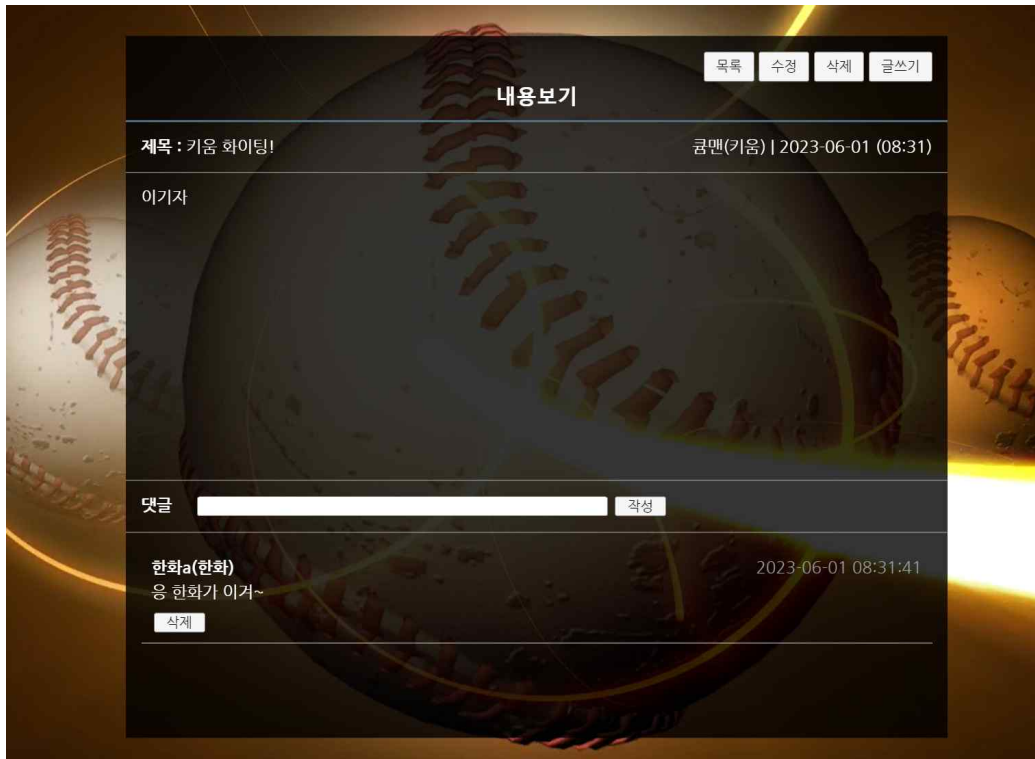


우측 상단의 수정 버튼과 삭제 버튼을 통해 글을 수정하거나 삭제할 수 있다.



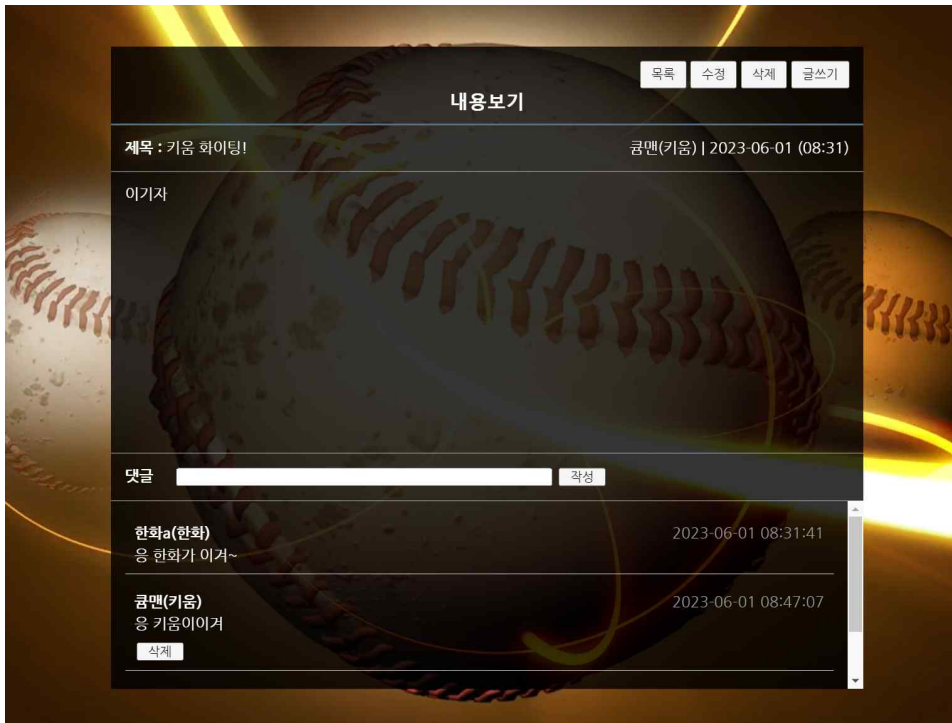
수정하기 버튼을 누르면 수정이 완료된다.

7) 게시글에 대한 댓글 쓰기 기능



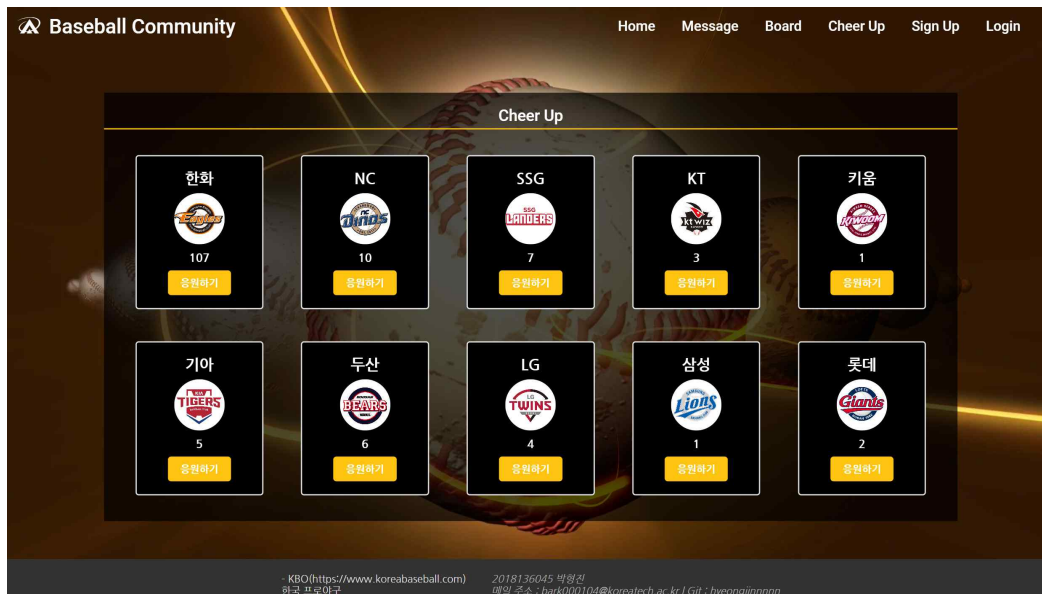
각 게시글에 댓글을 작성할 수 있다. 각 게시글에 있는 댓글들은 해당 게시글에 종속된다. 댓글 input에 글씨를 쓰고 작성 버튼을 누르면 댓글 입력이 완료된다. 댓글 역시 마찬가지로 작성자의 팀까지 나타나도록 하였다. 추가로 작성된 시간까지도 나타나도록 하였다.

8) 자신이 남긴 댓글만 삭제할 수 있는 기능



닉네임이 쿠팡맨인 아이디로 로그인하여 댓글을 작성했을 때 모습이다. 쿠팡맨인 사람에게 자신이 쓴 댓글만 삭제 버튼이 나타나는 것을 확인할 수 있다. 따라서 비 로그인 상태라면 모든 댓글에 대해 삭제 버튼이 나오지 않는다. 삭제 버튼을 누르면 자신이 쓴 댓글을 삭제할 수 있다.

9) 응원하기 기능



응원하기 기능은 로그인하지 않은 상태에서도 응원 버튼을 클릭할 수 있도록 구현해놓았다. 각각의 팀들에 대한 응원하기 버튼을 누르면 실시간으로 응원 수가 1씩 올라가면서 반영되는 것을 확인할 수 있다.

2. 주요 기능을 위한 DB 구조 설계

DB 테이블은 회원 정보를 관리하기 위한 members 테이블, 쪽지 정보를 관리하기 위한 message 테이블, 게시판을 관리하기 위한 board 테이블, 댓글을 관리하기 위한 comments 테이블, 응원 정보를 관리하기 위한 team_cheer 테이블로 나누어 구성했다. 이 테이블을 포함하는 데이터베이스의 이름은 term_project이다.

1) members 테이블

데이터 항목으로 team이 추가되어 있는 것을 확인할 수 있다.

Field	Type	Null	Key	Default	Extra
id	char(20)	NO	PRI	NULL	
pass	char(20)	NO		NULL	
name	char(10)	NO	PRI	NULL	
team	char(20)	NO		NULL	
email	char(80)	YES		NULL	
regist_day	char(20)	YES		NULL	

실제로 쿼리문을 통해 members 테이블을 생성하는 sql문이다.

```
create table members (  
    id char(20) not null,  
    pass char(20) not null,  
    name char(10) not null,  
    team char(20) not null,  
    email char(80),  
    regist_day char(20),  
    primary key(id, name)  
)
```


2) message 테이블

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
send_id	char(20)	YES	MUL	NULL	
rv_id	char(20)	YES	MUL	NULL	
subject	char(200)	NO		NULL	
content	text	NO		NULL	
regist_day	char(20)	YES		NULL	

send_id와 rv_id는 foreign key로 members 테이블의 id를 참조하도록 하였다. message 테이블을 생성하는 쿼리문은 다음과 같다.

```
create table message (  
    num int not null auto_increment,  
    send_id char(20),  
    rv_id char(20),  
    subject char(200) not null,  
    content text not null,  
    regist_day char(20),  
    primary key(num),  
    foreign key (send_id) references members(id),  
    foreign key (rv_id) references members(id)  
)
```

3) board 테이블

Field	Type	Null	Key	Default	Extra
num	int(11)	NO	PRI	NULL	auto_increment
id	char(20)	NO		NULL	
name	char(10)	NO		NULL	
subject	char(200)	NO		NULL	
content	text	NO		NULL	
regist_day	char(20)	NO		NULL	
hit	int(11)	NO		NULL	
file_name	char(40)	YES		NULL	
file_type	char(40)	YES		NULL	
file_copied	char(40)	YES		NULL	

board 테이블을 생성하는 쿼리문은 다음과 같다.

```
create table board (  
    num int not null auto_increment,  
    id char(20) not null,  
    name char(10) not null,  
    subject char(200) not null,  
    content text not null,  
    regist_day char(20) not null,  
    hit int not null,  
    file_name char(40),  
    file_type char(40),  
    file_copied char(40),  
    primary key(num)  
)
```


4) comments 테이블

comments 테이블은 새로 추가한 테이블로 댓글 정보를 가지는 테이블이다.

댓글 id, 해당 댓글이 속해있는 게시판 id, 댓글 작성자의 닉네임, 댓글 내용, 등록일을 데이터 항목으로 가진다.

Field	Type	Null	Key	Default	Extra
comment_id	int(11)	NO	PRI	NULL	auto_increment
board_id	char(20)	YES		NULL	
writer_name	char(10)	YES		NULL	
content	text	NO		NULL	
regist_day	char(20)	NO		NULL	

comments 테이블을 생성하는 쿼리문은 다음과 같다.

```
create table comments (  
  comment_id int not null auto_increment,  
  board_id char(20),  
  writer_name char(10),  
  content text not null,  
  regist_day char(20) not null,  
  primary key (comment_id)  
)
```

5) team_cheer 테이블

응원하기 기능을 위한 테이블이다. 팀 이름과 그 팀의 응원 수를 데이터 항목으로 가진다. 응원 수는 0으로 초기화되도록 설정한다.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
team_name	varchar(50)	NO		NULL	
cheer_count	int(11)	YES		0	

실제로 이 테이블을 만든 후에는 프로 구단 10개를 삽입해주어야 한다. 테이블을 만들고 삽입하는 sql문은 다음과 같다.

```
create table team_cheer (  
  id int auto_increment primary key,  
  team_name varchar(50) NOT NULL,  
  cheer_count int default 0  
);  
  
insert into team_cheer (team_name, cheer_count)  
values ('한화', 0),  
      ('NC', 0),  
      ('SSG', 0),  
      ('KT', 0),  
      ('키움', 0),  
      ('기아', 0),  
      ('두산', 0),  
      ('LG', 0),  
      ('삼성', 0),  
      ('롯데', 0);
```

3. 주요 기능 구현 방법

1) 오늘의 프로야구 경기 정보 제공 기능

웹 스크래핑 기술을 이용하여 경기 정보를 가져오는 데 성공했다. 이 기술을 위해 필요한 simple_html_dom.php 파일을 다운 받은 후 같은 폴더 내에 위치시켜야 한다. 밑에 코드를 보면 해당 php 파일을 include 시킨 것을 확인할 수 있다.

```
<section class="baseballResult">
  <h1>KBO 경기</h1>
</section>
<div class="play">
  <?php
    header("Content-Type: text/html; charset=UTF-8");
    include_once "simple_html_dom.php";
    $url = "https://mykbostats.com/";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $res = curl_exec($ch);
    $html = new simple_html_dom();
    $html->load($res);
  >>
  <div id="home__games__today" class="game-line-container">
    <?php echo $html->find("div#home__games__today", 0); >>
  </div>
</div>
```

가져오고자 하는 정보가 있는 홈페이지의 url을 적고 그 url을 통해 해당 페이지를 스크래핑한다. find를 통해서 그 페이지 안에 있는 가져오고자 하는 컨테이너 정보를 입력하여 해당 정보만을 가져올 수 있다.

2) 회원가입, 회원 정보수정 기능(응원팀 추가)

실제로 회원가입 과정에 팀을 추가하여 팀까지 데이터베이스에 삽입될 수 있도록 하였다.

```
<?php
    $id = $_POST["id"];
    $pass = $_POST["pass"];
    $name = $_POST["name"];
    $team = $_POST["team"];
    $email1 = $_POST["email1"];
    $email2 = $_POST["email2"];

    $email = $email1."@".$email2;
    $regist_day = date("Y-m-d (H:i)"); // 현재의 '년-월-일-시-분'을 저장

    $con = mysqli_connect("localhost", "user1", "12345", "term_project");

    $sql = "insert into members(id, pass, name, team, email, regist_day) ";
    $sql .= "values('$id', '$pass', '$name', '$team', '$email', '$regist_day')";

    mysqli_query($con, $sql); // $sql 에 저장된 명령 실행
    mysqli_close($con);

    echo "
    <script>
        location.href = 'index.php';
    </script>
    ";
?>
```

회원 정보수정 시 참조 무결성 원칙을 지키기 위해 sql2 sql3문을 만들어 board와 comments의 글쓴이의 이름도 아래와 같이 수정되도록 하였다.

```
$original_name = $row["name"];

$id = $_GET["id"];
$pass = $_POST["pass"];
$name = $_POST["name"];
$team = $_POST["team"];
$email1 = $_POST["email1"];
$email2 = $_POST["email2"];

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

$sql = "update members set pass='$pass', name='$name', team='$team', email='$email'";
$sql .= " where id='$id'";
mysqli_query($con, $sql);
$sql2 = "update board set name='$name' where name='$original_name'";
mysqli_query($con, $sql2);
$sql3 = "update comments set writer_name='$name' where writer_name='$original_name'";
mysqli_query($con, $sql3);
mysqli_query($con, $sql3);
mysqli_close($con);
```

3) 게시물에 대한 댓글 쓰기 기능

```
<?php
// 댓글 작성 처리
$con = mysqli_connect("localhost", "user1", "12345", "term_project");
if (isset($_POST['comment_content'])) {
    $comment_content = $_POST['comment_content'];
    $regist_day = date("Y-m-d H:i:s");

    $userid = $_SESSION["userid"];
    $query = "SELECT * FROM members WHERE id='$userid'";
    $result = mysqli_query($con, $query);
    $row = mysqli_fetch_assoc($result);

    $name = $row["name"];
    $team = $row["team"];

    // comments 테이블에 댓글 데이터를 삽입하는 쿼리
    $query = "INSERT INTO comments (board_id, writer_name, content, regist_day)
VALUES ('$num', '$name', '$comment_content', '$regist_day')";
    $result = mysqli_query($con, $query);
}
```

댓글 작성 버튼을 누르면 POST 방식을 통해 해당 정보를 받고 쿼리문을 통해 정보를 데이터베이스에 삽입한다. 현재 로그인된 userid의 이름을 추가로 넣어주기 위해 쿼리문을 통해 name을 얻는다. \$num은 board_id로써 위의 코드에서 보이지 않지만, board_id의 정보를 담고 있는 변수이다.

4) 게시판, 댓글 글쓴이의 팀 추가 제공 기능

아래 사진은 쿼리문을 통해 게시판 글쓴이의 응원 팀을 가져오는 코드이다.

```
// 글쓴이의 응원 팀 정보 가져오기
$user_sql = "select team from members where name = '$name'";
$user_result = mysqli_query($con, $user_sql);

if ($user_result) {
    $user_row = mysqli_fetch_array($user_result);
    $team = $user_row["team"];
} else {
    $team = "";
}
```

게시판 글쓴이의 팀 정보를 \$team에 넣어두고 팀까지 출력해준다.

```
<li>
    <span class="col1"><?=$number?></span>
    <span class="col2"><a href="board_view.php?num=<?=$num?>&page=<?=$page?>"><?
    <span class="col3"><?=$name."<?={$team}>"></span>
    <span class="col4"><?=$file_image?></span>
    <span class="col5"><?=$regist_day?></span>
    <span class="col6"><?=$hit?></span>
</li>
```

아래 사진은 쿼리문을 통해 댓글 작성자의 정보를 가져오는 코드이다.

```
<?php
// comments 테이블에서 해당 게시글의 댓글을 조회하는 쿼리
$query = "select * from comments where board_id = '$num'";
$result = mysqli_query($con, $query);

// 댓글이 존재하는 경우
if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        $comment_id = $row['comment_id'];
        $comment_writer = $row['writer_name'];
        $query2 = "SELECT * FROM members WHERE name='$comment_writer'";
        $result2 = mysqli_query($con, $query2);
        $row2 = mysqli_fetch_assoc($result2);
        $comment_writer_id = $row2['id'];
        $comment_writer_team = $row2['team'];
        $comment_content = $row['content'];
        $comment_regist_day = $row['regist_day'];
    }
}
```

\$comment_writer_team에 댓글 작성자의 팀을 넣어두고 함께 출력한다.

```
<span class="col1"><b><?=$comment_writer."<?={$comment_writer_team}>"></b></span>
<span class="col2"><?=$comment_regist_day; ?></span>
<p><?=$comment_content; ?></p>
```

5) 자신이 남긴 댓글만 삭제할 수 있는 기능

```
<?php if (isset($_SESSION["userid"]))  
    if ($_SESSION["userid"] == $comment_writer_id) {?  
        <form method="POST" action="board_view.php?num=?=$num?>&page=?=$page?>>  
            <button type="submit" name="delete_comment" value=?=$comment_id?>>삭제</button>  
        </form>  
    }  
<?php } ?>
```

위의 사진과 같이 현재 로그인되어 있다면 \$_SESSION을 통해 받는 아이디와 댓글 작성자의 아이디가 일치하는 경우에만 삭제 버튼이 생성된다. 따라서 자신이 댓글 작성자가 아니라면 삭제 버튼을 볼 수 없도록 구현했다. 삭제 버튼을 누르면 POST 방식을 통해 삭제되는 comment_id를 전달한다. 아래의 사진은 실제로 넘겨준 후 받은 comment_id 정보를 활용해 삭제하는 쿼리문이다.

```
// 댓글 삭제 처리  
if (isset($_POST['delete_comment'])) {  
    $delete_comment_id = $_POST['delete_comment'];  
  
    // comments 테이블에서 해당 댓글을 삭제하는 쿼리  
    $query = "DELETE FROM comments WHERE comment_id = '$delete_comment_id'";  
    $result = mysqli_query($con, $query);  
}  
?>
```

6) 응원하기 기능

```
<form name="cheer_form" method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
<div id="cheer_content">
    <div class="cheer_card">
        <h2>한화</h2>
        <div class="team_logo">
            
        </div>
        <div class="cheer_count"><?=$cheer_count[0]?></div>
        <button class="cheer_button" name="team_name" value="한화">응원하기</button>
    </div>
    <div class="cheer_card">
```

위의 코드와 같이 각 각의 버튼들은 POST 방식을 통해 팀 정보를 전달한다. 아래의 코드를 보면 알 수 있듯이 POST 방식을 통해 넘겨받은 팀 정보를 이용해 해당 팀의 cheer_count를 1씩 업데이트한다. cheer_count 출력 또한 쿼리문을 통해 모든 팀의 cheer_count를 \$cheer_count 배열로 저장해두고 위의 코드의 \$cheer_count[0]과 같이 활용함으로써 출력해줄 수 있다.

```
<?php
$con = mysqli_connect("localhost", "user1", "12345", "term_project");
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // 응원 버튼이 눌린 팀 이름 가져오기
    $team_name = $_POST["team_name"];

    // 해당 팀의 cheer_count 값을 1 증가시키는 쿼리문
    $sql = "update team_cheer set cheer_count = cheer_count + 1 where team_name = '$team_name'";

    $con->query($sql);
}

// cheer_count 값을 가져와서 배열에 저장
$cheer_count = array();
$sql = "select cheer_count from team_cheer order by id";
$result = $con->query($sql);

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $cheer_count[] = $row['cheer_count'];
    }
}
?>
```


Ⅲ. 결론

1. 추후 개발 계획

웹 스크래핑 기술을 활용하여 KBO 경기 정보를 가져오는 데 성공했지만, 이를 실제 상용화한다고 한다면 KBO 홈페이지에서 제공하는 API를 활용하는 편이 안전할 것으로 판단된다. 개인적인 홈페이지 제작 수준으로는 KBO에서 제공해주지 않을 것 같아서 웹 스크래핑으로 정보를 가져왔다. 프로젝트의 규모가 커지면 추후 API로 개발하는 것도 고려할 수 있을 것 같다. CSS를 통해 최대한 이쁘게 꾸미고자 노력했지만, 아직 많이 부족하다고 생각된다. 앞으로 추후 더 노력해서 미디어 쿼리를 이용한 동적 레이아웃까지도 수정해보고 싶다. 여러 기능을 최대한 넣어보고자 노력했고, 결과물로 잘 나온 것 같아 만족스럽다. 앞으로도 여기서 안주하지 않고 세세한 기능들, 더 활용도가 높고 즐거움을 줄 수 있고, 편리함까지 더해주는 여러 기능을 더 넣어서 프로젝트의 완성물을 더 높이고 싶다.

2. 개인적 소감

강의시간에 배웠던 SESSION이나 POST, GET 방식, 데이터베이스 관리 등을 직접 활용하며 프로그램을 만들어가면서 더 확실하게 내 지식으로 흡수할 수 있었다. 앱이나 GUI 프로그램 개발은 많이 해봤지만, 웹 프로그램 제작은 사실상 처음이어서 처음에는 막막하기도 하고, 도중에 막히는 부분도 많았지만 끝내 완성까지 해보면서 웹 프론트엔드, 백엔드의 실력 모두 업그레이드한 좋은 계기가 되었다.