# **SNSシステム開発 - コメント機能**

検索機能の次はコメント機能の実装を進めていきます。

## **概要**

今回は、コメント機能の実装します。この項も復習的な内容となっています。

これまで覚えた内容を思い出しながら取り組んでください。

## **要件確認**

コメント機能の要件は以下の通りです。

* 投稿：コメントは１：ｎの関係である
* 表示する内容はコメントしたユーザーの名前と画像、コメントの三つである

## 

## **テンプレートの確認**

コメント機能の実装前にテンプレートの確認を行いましょう。

コメント機能はタイムライン画面の一部機能として実装していきます。

**timeline.php**

<div class="row feed\_sub">

<div class="col-xs-12">

<button class="btn btn-default">いいね！</button>

いいね数：

<span class="like-count">10</span>

<a href="#collapseComment" data-toggle="collapse" aria-expanded="false"><span>コメントする</span></a>

<span class="comment-count">コメント数：5</span>

<?php if($signin\_user['id'] == $feed['user\_id']): ?>

<a href="edit.php?feed\_id=<?php echo $feed['id']; ?>" class="btn btn-success btn-xs">編集</a>

<a onclick="return confirm('ほんとに消すの？');" href="delete.php?feed\_id=<?php echo $feed['id']; ?>" class="btn btn-danger btn-xs">削除</a>

<?php endif; ?>

</div>

<?php include('comment\_view.php'); ?>

</div>

タイムライン画面の一部として実装をしますが、コメント画面はcomment\_view.phpに切り出しています。その切り出したcomment\_view.phpを各投稿情報を表示している箇所でincludeして呼び出しています。

なお、このincludeしているファイルは上部の「コメントする」をクリック時にトグルオープンされます。

## **投稿とコメントの紐づけ**

ただし、この時、オープンされるものが特定されていなければなりません。つまり、どの投稿に対するコメントかが分かるようにしなければなりません。

タイムライン画面とコメント画面をそれぞれ紐づくように修正しましょう。

**timeline.php**

<span class="like-count">10</span>

<a href="#collapseComment<?php echo feed['id']; ?>" data-toggle="collapse" aria-expanded="false"><span>コメントする</span></a>

<span class="comment-count">コメント数：5</span>

**comment\_view.php**

<div class="collapse" id="collapseComment<?php echo $feed['id']; ?>">

<div class="col-xs-12" style="margin-top:10px">

<form method="post" class="form-inline" action="comment.php" role="comment">

既に習ってきたように、includeされたファイルはinclude元で定義された変数を利用することが出来ます。

どの投稿か判別するために投稿のIDを利用して、コメントと連動させます。

ブラウザで各投稿に対してトグルが開かれることを確認してください。

## **コメント画面の修正**

続いてcomment\_view.phpの修正を行います。

コメント入力欄横のユーザーの画像をサインインしているユーザーのものに変更し、投稿のIDをサーバー側に送信できるように修正しましょう。

comment\_view.php

<form method="post" class="form-inline" action="comment.php" role="comment">

<div class="form-group">

<img src="user\_profile\_img/<?php echo $signin\_user['img\_name']; ?>" width="40" class="img-circle">

</div>

<div class="form-group">

<input type="text" name="write\_comment" class="form-control" style="width:400px;border-radius: 100px!important; -webkit-appearance:none;" placeholder="コメントを書く">

</div>

<input type="hidden" name="feed\_id" value="<?php echo $feed['id']; ?>">

<div class="form-group">

<button type="submit" class="btn btn-sm btn-primary">投稿する</button>

</div>

</form>

## **コメントの保存**

次にコメントを保存する機能を実装します。

コメントを保存する機能はcomment.phpに実装していきます。

必要な値は、下記の三点になります。

* コメントしたユーザーのID
* コメントする投稿のID
* コメント内容

ユーザーのIDはセッションから、他二つはポストパラメータから取得できます。

**comment.php**

<?php

session\_start();

require('dbconnect.php');

$user\_id = $\_SESSION['LearnSNS']['id'];

$feed\_id = $\_POST['feed\_id'];

$comment = $\_POST['write\_comment'];

$sql = 'INSERT INTO `comments` (`user\_id`, `feed\_id`, `comment`, `created`) VALUES (?, ?, ?, now())';

$data = [$user\_id, $feed\_id, $comment];

$stmt = $dbh->prepare($sql);

$stmt->execute($data);

header('Location: timeline.php');

exit();

特別新しい内容はありません。必要な値を取り出し、Create処理を行なっているだけです。

実装ができたらコメント画面からコメントを行い、DBに値が保存されることを確認してみてください。

## **コメントの取得**

保存が出来たので、次は取得処理を実装していきます。

各投稿毎にコメントを取得していくので、timeline.phpの投稿情報を取得している中で行なっていきます。

実行するSQLは下記のようになります。

SELECT `c`.\*, `u`.`name`, `u`.`img\_name` FROM `comments` AS `c`

LEFT JOIN `users` AS `u` ON `c`.`user\_id` = `u`.`id` WHERE `c`.`feed\_id` = ?;

それでは、記述する場所に気を付けて実装していきましょう。

**timeline.php**

$feeds = [];

while (true) {

$record = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($record == false) {

break;

}

$comment\_sql = 'SELECT `c`.\*,`u`.`name`,`u`.`img\_name` FROM `comments` AS `c` JOIN `users` AS `u` ON `c`.`user\_id` = `u`.`id` WHERE `feed\_id`=?';

$comment\_data = [$record['id']];

$comment\_stmt = $dbh->prepare($comment\_sql);

$comment\_stmt->execute($comment\_data);

$comments = [];

while (true) {

$comment = $comment\_stmt->fetch(PDO::FETCH\_ASSOC);

if ($comment == false) {

break;

}

$comments[] = $comment;

}

$record['comments'] = $comments;

$feeds[] = $record;

}

## **コメントの表示**

各投稿に対するコメントの一覧が取得出来たので、表示を行なっていきます。

投稿一覧を表示した時と同じようにコメント一覧も繰り返し文を用いて出力します。

**comment\_view.php**

<?php foreach ($feed['comments'] as $comment): ?>

<p style="margin-top: 30px; margin-bottom: 30px">

<img src="user\_profile\_img/<?php echo $comment['img\_name']; ?>" width="40" class="img-circle">

<span style="border-radius: 100px!important; -webkit-appearance:none;background-color:#eff1f3;padding:10px;margin-top:10px;">

<a href="profile.php"><?php echo $comment['name']; ?></a>

<?php echo $comment['comment’]; ?>

</span>

</p>

<?php endforeach; ?>

実装が出来たらブラウザを更新してみましょう。

DBに保存された値が正しく表示されていることを確認してください。

## **コメント数の取得と表示**

最後にコメント数の取得と表示を行います。

各投稿に対するコメント数の取得は下記のSQLで行えます。

SELECT COUNT(\*) AS `comment\_cnt` FROM `comments` WHERE `feed\_id` = ?;

実装する場所は先程のコメント取得の次が良いでしょう。

**timeline.php**

$record['comments'] = $comments;

$comment\_cnt\_sql = 'SELECT COUNT(\*) AS `comment\_cnt` FROM `comments` WHERE `feed\_id` = ?';

$comment\_cnt\_data = [$record['id']];

$comment\_cnt\_stmt = $dbh->prepare($comment\_cnt\_sql);

$comment\_cnt\_stmt->execute($comment\_cnt\_data);

$comment\_cnt\_result = $comment\_cnt\_stmt->fetch(PDO::FETCH\_ASSOC);

$record['comment\_cnt'] = $comment\_cnt\_result['comment\_cnt'];

$feeds[] = $record;

コメント数が取得できたので、画面に出力しましょう。

**timeline.php**

<span class="comment-count">コメント数：<?php echo $feed['comment\_cnt']; ?></span>

以上で、コメント機能が完成しました。