# **SNSシステム開発 - ユーザー一覧**

ここで一度タイムラインの復習としてユーザー一覧の実装を進めていきます。

## **概要**

このページでは、ユーザーの一覧を表示する画面を作成します。

なお、ここで実装する内容のほとんどがこれまでの復習になります。ぜひ解説を見る前に、要件確認後に自力で解いてみてください。

## **要件確認**

ユーザー一覧の要件は以下の通りです。

* 1ページあたりの表示件数に制限はない
* 登録されているユーザーの情報が表示される
* 表示する内容はユーザーの名前、画像、登録日時、つぶやき数の四つである

## **全ユーザーの一覧を取得**

まず、ユーザー一覧を取得するSQLを考えていきましょう。

SELECT \* FROM `users`;

至ってシンプルなSELECT文でユーザーの一覧取得が可能でしょう。

PHPの実装に関してもこれまでと同様に行なうことになります。

なお、ユーザーの一覧画面においても、サインインしているユーザーの情報がnavbarで必要なので、こちらも併せて実装していきます。

**users.php**

<?php

session\_start();

require('dbconnect.php');

// サインインしているユーザーの情報を取得

$sql = 'SELECT \* FROM `users` WHERE `id` = ?';

$data = [$\_SESSION['LearnSNS']['id']];

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

$stmt->execute($data);

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

// ユーザーの一覧を取得

$sql = 'SELECT \* FROM `users`';

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

$stmt->execute();

$users = [];

while (true) {

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

if ($record == false){

break;

}

$users[] = $record;

}

?>

実装ができたらvar\_dumpを用いて$usersの中を確認しておきましょう。

## **ユーザー一覧をHTMLに出力**

上記の実装で$usersにユーザー一覧が取得できたはずです。

次は取得できたユーザー一覧をHTMLに出力していきます。

**users.php**

<div class="container">

<?php foreach ($users as $user): ?>

<div class="row">

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

<div class="thumbnail">

<div class="row">

<div class="col-xs-1">

<img src="user\_profile\_img/<?php echo $user['img\_name']; ?>" width="80">

</div>

<div class="col-xs-11">

名前 <?php echo $user['name']; ?><br>

<a href="profile.php?" style="color: #7F7F7F;">

<?php echo $user['created']; ?>からメンバー

</a>

</div>

</div>

<div class="row feed\_sub">

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

<span class="comment\_count">つぶやき数 : 10</span>

</div>

</div>

</div>

</div>

</div>

<?php endforeach; ?>

</div>

解説は不要でしょう。

取得した$usersを繰り返し文で出力すれば完成です。

## **各ユーザーのつぶやき数を取得**

ユーザーの一覧表示が出来たら続いて、各ユーザーのつぶやき数を取得します。

まずはSQLを考えていきましょう。

今回取得すべきデータは投稿情報ですので、取得元はfeedsテーブルになります。各投稿情報には誰が投稿したかを保持するuser\_idというカラムがあります。このカラムを用いて各ユーザー毎に何件投稿したかが把握できます。

SELECT COUNT(\*) AS `cnt` FROM `feeds` WHERE `user\_id` = ?;

SQLが分かったので、PHPの実装をしていきます。

が、実装する場所に注意してください。

各ユーザーのIDで絞り込む必要があるので、各ユーザーの情報にアクセスできる箇所で実装を行う必要があります。

ここまでの実装で各ユーザーの情報にアクセスできるのはどこでしょう。

ユーザー一覧を取得し、レコードを一件一件$usersに追加している場所です。

それでは実装を見てみましょう。

**users.php**

$users = [];

while(true) {

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

if ($record == false) {

break;

}

// 各ユーザーのつぶやき数を取得

$feed\_sql = 'SELECT COUNT(\*) AS `cnt` FROM `feeds` WHERE `user\_id` = ?';

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

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

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

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

$record['feed\_cnt'] = $feed['cnt'];

$users[] = $record;

}

SQLの実行結果を取得している中で、また新たなSQLの実行結果を取得しています。

これは新しいパターンではありますが、それぞれの実装内容に新しいものはありません。

もし混乱するようであれば1行1行ゆっくりと読み解いて、しっかり理解をしましょう。

なお、取得できたつぶやき数は$recordに新しいキーを設定して値を代入します。

連想配列の新しい要素の追加の方法は、下記の通りです。

$連想配列名[新しいキー] = 値;

## **各ユーザーのつぶやき数を表示**

さて、各ユーザーのつぶやき数が取得できたので、HTMLに出力していきましょう。

**users.php**

<div class="row feed\_sub">

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

<span class="comment\_count">つぶやき数 : <?php echo $user['feed\_cnt'] ?></span>

</div>

</div>

各ユーザーの情報出力は既に実装が出来ているので、上記の一箇所を修正すれば完成です。

## **ナビゲーションの修正**

最後に、ナビゲーションの修正を行なっていきます。

現状、タイムライン画面にいてもユーザー一覧画面にいても、タイムラインが活性化しているように表示されています。

ユーザー一覧画面を表示している場合は、ユーザー一覧が活性化するように修正していきましょう。

**navbar.php**

<?php if (strpos($\_SERVER['REQUEST\_URI'], 'timeline.php') !== false): ?>

<li class="active"><a href="timeline.php">タイムライン</a></li>

<li><a href="users.php">ユーザー一覧</a></li>

<?php else: ?>

<li><a href="timeline.php">タイムライン</a></li>

<li class="active"><a href="users.php">ユーザー一覧</a></li>

<?php endif; ?>

ここで、strpos関数と$\_SERVERという新しいものが出てきました。

##### **strpos()**

strpos関数は文字列内の部分文字列が最初に現れる場所を見つける関数です。

例えば、$pos = strpos('abc', 'a');とすると、$posには0が代入されます。

文字列の開始位置が0から始まることに注意しましょう。

##### **$\_SERVER**

$\_SERVERはスーパーグローバル変数の一つで、Webサーバの情報やプログラム実行時の環境情報などが格納されています。例えば、ホスト名やIPアドレス、リクエストのプロトコル、リクエスト元のIPアドレスなどが挙げられます。

今回は$\_SERVER['REQUEST\_URI']、つまりリクエストされたURLを利用して制御しています。

if文の比較演算子が!==となっているのは、strpos関数の結果が0であった場合に、falseとして扱われてしまう可能性があるため、型のチェックを含めて判別する必要があるからです。

以上で、ユーザー一覧の実装は完成です。