# **SNSシステム開発 - プロフィール機能**

次はユーザー周りの機能に移り、プロフィール機能の実装を進めていきます。

## **概要**

今回は、プロフィール機能の実装をします。各ユーザーの詳細情報を表示するページになります。

なお、profile.phpでもサインインしているユーザーの情報が必要になるので、これまで通り$signin\_userという変数を用意しておきましょう。

## **要件確認**

プロフィール機能の要件は以下の通りです。

* ユーザー毎の詳細情報が表示出来る
* 表示する内容はユーザーの名前と画像、フォローしているユーザー一覧、フォローされているユーザー一覧である
* 各ユーザーのページからはフォローをすることが出来る

## **選択したユーザーの表示**

まずは選択されたユーザーの情報を画面左部に表示します。

**選択されたユーザーの情報取得**

プロフィール機能はprofile.phpに実装していきますが、その前にまずユーザー一覧を修正しておきましょう。

**users.php**

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

名前 <?php echo $user['name']; ?><br>

<a href="profile.php?user\_id=<?php echo $user['id']; ?>" style="color: #7F7F7F;">

<?php echo $user['created']; ?>からメンバー

</a>

</div>

ユーザー一覧からどのユーザーが選択されたかを示すために、GETパラメータを追加しました。送るべき値はどのユーザーかなので、一意に絞り込めるIDをパラメータに利用します。

これでprofile.phpに遷移した際に、選択されたユーザーが分かるようになりました。

**profile.php**

<?php

require('dbconnect.php');

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

$data = [$\_GET['user\_id']];

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

$stmt->execute($data);

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

?>

選択されたユーザーの情報を取得するので、CRUD処理のREAD処理を行っていきます。

GETパラメータで送られた値を利用することに注意してください。

**サインインユーザーの情報表示**

選択されたユーザーの情報が取得できたので表示を行います。

**profile.php**

<div class="col-xs-3 text-center">

<img src="user\_profile\_img/<?php echo $profile['img\_name']; ?>" class="img-thumbnail" />

<h2><?php echo $profile['name']; ?></h2>

<a href="follow.php">

<button class="btn btn-default btn-block">フォローする</button>

</a>

</div>

## **フォロー機能**

次にフォロー機能を実装していきます。「誰をフォローしているか」「誰にフォローされているか」を示すものです。

**テーブルの追加作成**

実装をする上で、現在のDBにはこのフォローを保持するテーブルが存在しません。

まずは、下記のSQL文をphpMyAdminから実行してテーブルを追加しましょう。

**SQL**

CREATE TABLE `followers` (

`id` int(11) UNSIGNED NOT NULL,

`user\_id` int(11) DEFAULT NULL,

`follower\_id` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `followers`

ADD PRIMARY KEY (`id`);

ALTER TABLE `followers`

MODIFY `id` int(11) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9;

**フォロー処理の実装**

次に「フォローする」ボタンを押した時の処理を作成します。

**profile.php**

<a href="follow.php?following\_id=<?php echo profile['id']; ?>">

<button class="btn btn-default btn-block">フォローする</button>

</a>

誰をフォローするのかをサーバ側に送信する必要があるので、ここでもGET送信を利用します。

フォロー機能自体はfollow.phpに記述していきます。

フォローするユーザーとフォローされるユーザーのIDをDBに保存すれば、それがフォロー状態を示します。

フォロー状態をDBに保存ができたら元のプロフィール画面に戻ります。

**follow.php**

<?php

session\_start();

require('dbconnect.php');

$user\_id = $\_GET['following\_id'];

$follower\_id = $\_SESSION['id'];

$sql = 'INSERT INTO `followers` (`user\_id`, `follower\_id`) VALUES (?, ?)';

$data = [$user\_id, $follower\_id];

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

$stmt->execute($data);

header('Location: profile.php?user\_id=' . $user\_id);

exit();

## **フォローボタンの制御**

次にフォローボタンについて処理を加えていきます。

**自分自身へのフォローを制限**

例えば、サインインしているユーザーが自分自身をフォローできるでしょうか？現状の作りでは可能ですが、システムとして不自然なものになりますね。

**profile.php**

<?php if ($signin\_uesr['id'] != $profile['id']): ?>

<a href="follow.php?following\_id=<?php echo profile['id']; ?>">

<button class="btn btn-default btn-block">フォローする</button>

</a>

<?php endif; ?>

制御は単純にサインインしているユーザーのIDと選択されたユーザーのIDを比較し、同一だった場合表示されない、つまり自分以外の場合ボタンが表示されるようになります。

実装が出来たらブラウザで挙動を確認しましょう。

**フォローする/解除するボタンの切り替え**

次は、フォロー済みのユーザーを表示した場合です。

既にフォローをしている場合は、今度は逆にフォローを解除する処理が行えるべきです。

まずはフォロー済みかどうかを取得していきましょう。

**profile.php**

$sql = 'SELECT `id` FROM `followers` WHERE `user\_id` = ? AND `follower\_id` = ?';

$data = [$profile['id'], $signin\_user['id']];

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

$stmt->execute($data);

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

この$is\_followedがフォロー状態を示します。fetchを行って、レコードが取得できない場合にfalseが入ることをしっかり覚えておきましょう。

続いて、ボタンの切り替えを実装していきます。

**profile.php**

<?php if ($signin\_user['id'] != $profile['id']): ?>

<?php if ($is\_followed): ?>

<a href=""><button class="btn btn-default btn-block">フォロー解除する</button></a>

<?php else: ?>

<a href="follow.php?following\_id=<?= $profile\_user['id']; ?>"><button class="btn btn-default btn-block">フォローする</button></a>

<?php endif; ?>

<?php endif; ?>

if文のネスト(入れ子)になっていますが、一つ目のif文は自分と同一かを条件分岐しているif文で、二つ目のif文がフォローしているかどうかの条件分岐になります。

フォロー後にボタンの内容が変わることを確認しましょう。

## **フォロー解除の処理**

続いてフォロー解除の処理を実装します。この実装もfollow.phpに実装を行っていきます。

GETパラメータを複数送信する方法を復習しておきましょう。

URL?キー1=値1&キー2=値2

それでは、フォロー解除のリンクを修正していきます。

**profile.php**

<?php if ($signin\_user['id'] != $profile['id']): ?>

<?php if ($is\_followed): ?>

<a href="follow.php?following\_id=<?php echo $profile['id']; ?>&unfollow=true"><button class="btn btn-default btn-block">フォロー解除する</button></a>

<?php else: ?>

<a href="follow.php?following\_id=<?= $profile\_user['id']; ?>"><button class="btn btn-default btn-block">フォローする</button></a>

<?php endif; ?>

<?php endif; ?>

リンクの修正が終わったので、今度はフォローするボタンが押された場合と、フォロー解除するボタンが押された場合で、処理を分けるようにします。

**follow.php**

if (isset($\_GET['unfollow'])) {

// フォロー解除する

} else {

// フォローする

}

フォローする場合は、CRUD処理のC、つまりINSERT処理でしたが、解除する場合は何になるでしょうか。

フォロー状態はfollowersテーブルのデータの有無で判別できます。レコードがあればフォロー状態になり、なければフォローしていない状態になります。

上記のことから、フォローを解除するということは、followersテーブルからレコードを削除することを意味します。つまり、CRUDのD、DELETE処理を利用します。

なお、今回の処理では、INSERT文でもDELETE文でも実行するSQL文が変わるだけで、その後の流れは変わらないことに注意してください。

**follow.php**

if (isset($\_GET['unfollow'])) {

// フォロー解除する

$sql = 'DELETE FROM `followers` WHERE `user\_id` = ? AND `follower\_id` = ?';

} else {

// フォローする

$sql = 'INSERT INTO `followers` (`user\_id`, `follower\_id`) VALUES (?, ?)';

}

$data = [$user\_id, $follower\_id];

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

$stmt->execute($data);

実装が出来たら、画面を更新して、ボタンを押すたびにボタンの内容が切り替わることと、レコードに変更が及んでいることを確認しましょう。

## **ユーザーをフォローしているユーザーの一覧**

続いてフォロワーの一覧表示を実装していきます。

フォロー情報を保持しているのはfollowersテーブルですが、ユーザー情報を保持しているのはusersテーブルになります。followersテーブルの情報を元にusersテーブルの情報を取得しましょう。

**profile.php**

$sql = 'SELECT `u`.\* FROM `followers` AS `f` LEFT JOIN `users` AS `u` ON `u`.`id` = `f`.`follower\_id` WHERE `f`.`user\_id` = ?';

$data = [$profile['id']];

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

$stmt->execute($data);

$followers = [];

while (true) {

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

if ($record == false) {

break;

}

$followers[] = $record;

}

一覧を取得出来たら表示を行っていきましょう。

**profile.php**

<div id="tab1" class="tab-pane fade in active">

<?php foreach ($followers as $follower): ?>

<div class="thumbnail">

<div class="row">

<div class="col-xs-2">

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

</div>

<div class="col-xs-10">

名前 <a href="profile.php?user\_id=<?php echo $follower['id']; ?>" style="color:#7F7F7F;"><?php echo $follower['name']; ?></a>

<br>

<?php echo $follower['created']; ?>からメンバー

</div>

</div>

</div>

<?php endforeach; ?>

</div>

## **ユーザーがフォローしているユーザーの一覧**

最後に、フォロワーとは逆にフォローイングの一覧表示を実装していきます。

**profile.php**

$sql = 'SELECT `u`.\* FROM `followers` AS `f` LEFT JOIN `users` AS `u` ON `u`.`id` = `f`.`user\_id` WHERE `f`.`follower\_id` = ?';

$data = [$profile['id']];

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

$stmt->execute($data);

$followings = [];

while (true) {

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

if ($record == false) {

break;

}

$followings[] = $record;

}

フォロワーを取得した先程のSELECT文とは、ON句とWHERE句に違いがあることに注意してください。

<div id="tab2" class="tab-pane fade">

<?php foreach ($followings as $following): ?>

<div class="thumbnail">

<div class="row">

<div class="col-xs-2">

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

</div>

<div class="col-xs-10">

名前 <a href="profile.php?user\_id=<?php echo $following['id']; ?>" style="color: #7F7F7F;"><?php echo $following['name']; ?></a>

<br>

<?php echo $following['created']; ?>からメンバー

</div>

</div>

</div>

<?php endforeach; ?>

</div>

実装が出来たらブラウザでタブを切り替え、正しく表示されることを確認しましょう。

以上でプロフィール機能の実装が完了です。