



# (仮) アイデア倉庫

## 各種機能説明

# 本システムの概要

- 目的

「利用者のブラウザから、質問やアイデアの発信、質問等に関する意見の発信や閲覧を行う」

- 期待効果

「基システムにより、利用者の利便性向上とアイデア募集、質問の回答募集を円滑に行い、社内改善や社内コミュニケーションの向上を目指す」

# 要求について

- とことんシンプルな質問サイトとする。
- 利用者は、質問や回答を見ることが出来る。
- 回答は、質問に紐つく。
- 1 質問に対する回答は、複数存在する。
- 利用者は、質問や回答を投稿することが出来る。
- 質問や回答は、誰でも見ることが出来る。
- 質問や回答を投稿する場合は、ユーザー認証が必要。
- ログインは、IDとパスワードで行う。
- ログインするには、ユーザー登録が必要。
- ログイン後は、ログアウトすることが出来る。
- 質問や回答には、画像は不要
- 質問や回答は、投稿したら、編集できなくてよい
- 但し、削除はできる。
- 回答の削除は、指定した回答のみを削除する。
- 質問の削除は、回答と共に削除する。
- 削除したものは、データベースには、残しておく。
- 投稿した利用者のみ削除することが出来る。
- 質問の削除は、質問を投稿した利用者のみ可能で、他者の回答も併せて削除する
- 回答の削除は、投稿した自身の回答のみ
- 質問に、カテゴリのような区分は不要
- ユーザーの退会やパスワード再発行は不要  
テキストのみ投稿できる。

## 追加機能 15個

- 新規登録注意事項切り替え表示機能
- パスワード変更機能
- パスワードの文字制限数変更
- 長い文章の短縮表示機能
- 質問検索機能
- 質問者検索機能
- 複数単語検索機能
- 文字入力制限機能
- 入力フォーム注意事項表示機能
- 入力フォーム文字数カウント機能
- 入力フォーム保持機能
- ソート機能
- 自他の質問抽出機能
- 各検索、ソート、抽出同時使用機能
- コメント返信機能



# 個人発表

```
// リーダー  
if(isset(川満 達也)) {  
    $work = 質問一覧, 追加機能,  
}
```



# 作成演習における所感

- リーダー業務

全体作業における全体像の確認、チームメンバーの掌握・作業配分など

「自分正面の作業に没頭すると掌握が疎かになり、チーム作業の効率が低下する。作業指揮と自己作業の配分に気を遣う」

- タスク作業

質問一覧の作成作業、遷移するページ担当との調整

「リーダーの権限で変数やパラメーター名の統一が容易だった。作業自体は短時間で終了し、リーダー業務やメンバーのヘルプ、追加機能の実装とバグ修正に注力できた」

- 追加機能

各追加機能の処理構想、機能の実装とバグ修正

「入れ込みたい機能と教科書の内容などを分析して実装。一つ一つは単純な仕組みであるが、組み合わせた際の予期せぬ挙動やバグの解消が面倒」

\$\_SESSION( 浅野 裕子 ) = [  
    work1 => ログインページ,  
    work2 => 質問入力  
];



# ◆Index.php：ログイン画面◆

## 【苦戦した点】

- ・未ログイン時、各ページからの値を保ちながらログイン処理を促す時に、リダイレクトさせる時の処理

↓

\$GET\_\_[ ] 送信で行う

↓

\$POST\_\_[ ] 送信で上書きされる

↓

ログイン成功時はすぐに遷移できるが、**ログイン失敗**した時に、また値が消えてしまうので、エラー時にも再びエラー内容を格納し、表示させる必要があった

<!--\$\_GET[ 'loginFlg' ]で送られてきた値を変数へ代入-->

```
<?php
$loginFlg = $_GET['loginFlg'] ?? '';
$id = $_GET['id'] ?? '';
```

## 【次回への課題】

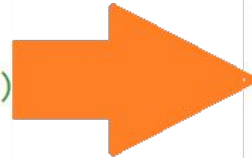
- ・細かい遷移先、連携先の理解、各コードの理解など、個人としての知識・技術力の向上
- ・チームとして動く中での自分自身の立ち回り方や、自分の意図を伝えるだけでなく、仲間の考えをくみ取ることのできるコミュニケーションスキルの習得 && 楽しむ心☆

// デバック隊長  
var\_dump( 佐々木 美智子 );  
work(27) “新規登録、追加機能”

# 苦労した点：エラーメッセージ表示方式の改善について

## ■ 変更前

```
if (preg_match('/^.{1,10}$/', $viewName)) {
if (preg_match('/^.{8,10}$/', $userId)) {
if (preg_match('/^.{6,10}$/', $pass)) {
    $sql=$pdo->prepare(
        'select * from user where loginId=? or password=?' );
    $sql->execute([$userId,$pass]);
    $existing = $sql->fetch(PDO::FETCH_ASSOC);
    if ($existing) {
        // --- どちらか一方でも既に存在する場合 ---
        echo 'ログインIDまたはパスワードがすでに使用されています。別の
    } else {
        // --- 新規登録処理 ---
        $sql=$pdo->prepare('insert into user values(null,?,?,?)');
        $sql->execute([$userId,$pass,$viewName]);
        echo 'お客様情報を登録しました。';
    } else {
        $err="パスワードは6文字以上10文字以下で入力してください。";
    }
    } else {
        $err= "ユーザーIDは8文字以上10文字以内で入力してください。";
    }
    } else {
        $err= "表示名は1文字以上10文字以内で入力してください。";
    }
}
```



## ■ 変更後

```
$errors = [];
if (!preg_match('/^.{1,10}$/u', $viewName)) {
    $errors[] = "表示名は1～10文字";
}
if (!preg_match('/^[a-zA-Z0-9]{8,10}$/', $userId)) {
    $errors[] = "ユーザーIDは半角英数字8～10文字";
}
if (!preg_match('/^[a-zA-Z0-9]{6,20}$/', $pass)) {
    $errors[] = "パスワードは半角英数字6～20文字";
}
if (!empty($errors)) {
    $errors = implode(" ", $errors). "で入力してください"; // 結合
} else {
    $sql=$pdo->prepare(
        'SELECT * from user where loginId=? or password=?' );
    $sql->execute([$userId,$pass]);
    if (empty($sql->fetchAll())) {
        // 新規登録処理
        $sql=$pdo->prepare('INSERT into user values(null,?,?,?)');
        $sql->execute([$userId,$pass,$viewName]);
        header("Location: ../index.php");
    } else {
        // どちらか一方でも既に存在する場合
        $errors= 'ログインIDまたはパスワードがすでに使用されています。別
    }
}
```

```
$pdo = new PDO( 池田 喜一  
郎);  
$sql = $pdo-> work (   
データベース, 回答入力  
);
```



## 頑張ったところや、難しかったところ：池田

**SELECT**

```
answer.id AS 回答番号,  
answer.questionId AS 質問番号 (=question.id) ,  
answer.userId AS 回答者 (利用者) 番号 (=user.id) ,  
question.userId AS 質問者 (利用者) 番号 (=user.id) ,  
answer.answer AS 回答内容
```

**FROM answer**

```
INNER JOIN question ON answer.questionId = question.id  
INNER JOIN user ON answer.userId = user.id");
```

- デバッグの為に、テーブルを3つ結合し、データベースを確認しにいく手間を軽減。

- 参照する「id」を誤って、データベースからの参照ができなくなった

構造等の理解が薄いまま作成していたので、次の演習では把握して進めていきたい！



```
// サブリーダー  
require    ‘猪本 千尋’ ;  
header( work:質問詳細, CSS, );
```



```

33 + <?php
34 + // 既存の質問一覧をファイルから読み込む（ファイルがない場合は空の配列）
35 + $board=file_exists('board.txt') ? file('board.txt',FILE_IGNORE_NEW_LINES) : [];
36 +
37 + // フォームなどから送信されたデータを受け取る
38 + if(isset($_REQUEST['question'])){
39 +     //HTML特殊文字をエスケープ（xxs対策）
40 +     $question=htmlspecialchars($_REQUEST['question'], ENT_QUOTES, 'utf8');
41 +     $board[]= $question;
42 +
43 +     // 質問一覧をファイルに保存
44 +     file_put_contents('board.txt', implode("\n",$board));
45 + }
46 +
47 + // 保存されている質問をすべて表示
48 + foreach($board as $question){
49 +     echo '<p>', $question, '</p>';
50 + }

```

## ●苦勞したこと：画面の仕様を理解すること、コミュニケーション

- ・はじめは理解が甘くて、的外れなコードを書いていた

→日本語で処理を整理してから、コードに書く

- ・「相手が言っていることが分からない」「どう言ったらうまく伝わるか分からない」

## ●改善点

構造理解のための時間をしっかり確保することと、コミュニケーションの中でより良い伝え方・聞き方を練習することを意識して臨みたいと思います。