

PHP 練習問題. 13 ログインとログアウト

練習問題 13 では「ログイン」と「ログアウト」の機能について学習します。

- ログイン機能
 1. ログイン画面で
 - 「ユーザー名 (ユーザーID)」と「パスワード」を入力します。
 2. データベースのテーブルから、入力した「ユーザー名」と「パスワード」に一致するユーザーを検索します。
 3. 一致するユーザーがあれば、セッションにユーザーの情報を保存し、ログイン後に表示されるページへリダイレクトします。
 4. 一致するユーザーがなければ、ログイン画面へリダイレクトし、エラーメッセージを表示します。
- ログインが必要なページの処理
 1. セッションにユーザー情報があれば、表示します。
 2. セッションにユーザー情報がなければ、ログインページへリダイレクトします。
- ログアウト機能
 1. セッションに保存されたユーザーの情報を破棄し、ログインページへリダイレクトします
- パスワードについて
 1. パスワードは、悪意を持った第三者にサーバーをハッキングされてもパスワードがもれないように、暗号化処理を施します。
 2. 「いきなりプロ並み PHP」の書籍では、md5()という関数を使って暗号化処理を行っています。
 3. しかし、md5()は暗号化強度が弱いため、現在では利用されることはありません。代わりに、**password_hash()**と**password_verify()**を使います。

password_hash()

<https://www.php.net/manual/ja/function.password-hash>

password_verify()

<https://www.php.net/manual/ja/function.password-verify.php>

※ 「PHP 練習問題 12」で作成したファイルをそのまま使用します。

練習問題のファイルをコピーしてお使いください。

設問1. 「PHP 練習問題 08」で作成したデータベースに、下記のテーブルを追加してください。

テーブル名： users

デフォルトの文字コード： utf8mb4_general_ci

名前	データ型	Null 不許可	デフォルト値	
id	INT	yes		Primary Key AUTO_INCREMENT
email	VARCHAR(256)	yes		
password	VARCHAR(256)	yes		
name	VARCHAR(50)	yes		

- ✓ email はログイン ID に使います。
- ✓ E-mail の最大の長さはインターネットの仕様で 256 文字と決められています。
- ✓ パスワードはハッシュ化するので、最大 256 文字程度必要です。

設問2. Users.php を class/db のディレクトリの中に新規作成します。

Users.php では、下記の内容を記載してください。

- (1) Base クラスを継承した Users クラスを作ります。
- (2) コンストラクタで親クラス（スーパークラス）のコンストラクタを呼び出してください。
- (3) 以降、ユーザーに関する処理のメソッドは、Users クラスの中に作ります。

設問3. user_add.php を新規作成し、下記のようにフォームを作成してください。form の action は user_add_action.php にしてください。

● user_add.php

新規ユーザー登録

Email address

Password

お名前

登録

設問4. user_add_action.php を新規作成します。下記の処理を行ってください。

- (1) POST されてきたメールアドレスが既にテーブルに存在しているかどうか調べます。
 - ① 存在しているとき、エラーメッセージをセッションに保存して、user_add.php にリダイレクトし、エラーメッセージを表示します。
 - ② 存在していないとき、処理を続行します。
- (2) パスワードを password_hash()関数を使ってハッシュ化（暗号化）します。

```
$pass = password_hash($pass, PASSWORD_DEFAULT);
```

- (3) Users テーブルにインサートします。
- ① インサート処理が成功したら、login.php へリダイレクトします。
(login.php は後ほど作成します)
セッションに保存したエラーメッセージは削除します。
 - ② 処理が失敗したら、エラーメッセージをセッションに保存して、
user_add.php にリダイレクトし、エラーメッセージを表示します。
- (4) 「設問 4」の一連の処理は、Users クラスにメソッドを作り、
user_add_action.php で呼び出して使いましょう。
- (5) user_add.php を編集します。user_add_action.php で発生したエラー
メッセージの内容を user_add.php で表示するようにします。
- (6) レコードが正常にインサートされているかどうか、phpMyAdmin などの
ツールで確認しましょう。

● user_add.php

新規ユーザー登録

Email address

satomi@sample.jp

Password

.....

お名前

さとみ

登録

インサート処理に成功したら、login.php にリダイレクトされます。(が、
ファイルはまだないので、Not Found になります)

Not Found

The requested URL /php_exam/13/login.php was not found on this server.

すでに同じメールアドレスが登録されているときは user_add.php へリダイレクトし、エラーメッセージを表示します。

新規ユーザー登録

既に同じメールアドレスが登録されています。

Email address

satomi@sample.jp

Password

お名前

さとみ

登録

例外スローされたとき（データベースに接続できないなど）は、user_add.php へリダイレクトし、エラーメッセージを表示します。（MySQL を Stop するなどしてお試しください）

新規ユーザー登録

申し訳ございません。エラーが発生しました。

Email address

satomi@sample.jp

Password

お名前

さとみ

登録

インサート処理に成功しているかどうかは、phpMyAdmin などのツールで確認してください。パスワードはハッシュ化（暗号化）されているので、複雑な文字列になっています。

id	email	password	name
1	satomi@sample.jp	\$2y\$10\$I/dC4m8irxjDZTiGQsKEO.IN2B9lq/t3y6W9v65rkab...	さとみ

設問5. login.php を新規作成します。下記のように、ログインフォームを作成してください。form の action は login_action.php にします。

- login.php

ログイン

Email address

Password

ログイン

設問6. login_action.php を新規作成します。下記の処理を行います。

- (1). 入力された Email のユーザーを検索します。
 - ① 該当のユーザーがなかったら、login.php にリダイレクトし、エラーメッセージを login.php で表示します。
- (2). 該当のユーザーが存在したら、入力されたパスワードとレコードに保存されたハッシュ化されたパスワードを password_verify()関数を使って照合します。
 - ① パスワードの照合ができなかったときは、login.php にリダイレクトし、エラーメッセージを login.php に表示します。
- (3). パスワードの照合ができたときは、ユーザーの情報をセッションに保存し、login.php にリダイレクトします。

ログインに成功したとき、トップページが表示されます。

TODOリスト

2020/03/15

TODO項目を入力してください

追加

期限日	TODO項目
2020-03-05	てすと <div><div><input checked="" type="radio"/> 未完了</div><div><input type="radio"/> 完了</div><div><input type="checkbox"/> 削除</div><div>実行</div></div>

CSVファイルに変換

CSVファイルをダウンロード

CSVファイルをアップロードして更新

ログインに失敗したとき、login.php にリダイレクトし、エラーメッセージを表示します。

ログイン

メールアドレス、または、パスワードに誤りがあります。

Email address

satomi@sample.jp

Password

ログイン

※ 「メールアドレスが間違っています」「パスワードに誤りがあります」などと、個別のエラー表示はしません。なぜでしょうか？

設問7. index.php でセッションに保存されているユーザーの情報を表示しましょう。また、ログインしていないときは、login.php へリダイレクトするようにします。

※ ログインしていないときに login.php へリダイレクトする機能は、下記のフィルにも実装します。

- action.php
- add.php
- csv.php
- download.php
- update.php
- upload.php

logout.php へのリンクを設置します。

● index.php

The screenshot displays a web application interface for managing a TODO list. At the top, there is a header bar with the text 'TODOリスト' on the left, 'ログインユーザー：さとみ' in the center, and a 'ログアウト' button on the right. Below the header, there is a form for adding a new TODO item. It includes a date input field containing '2020/03/15', a text input field with the placeholder 'TODO項目を入力してください', and a blue '追加' button. Underneath the form is a table with two columns: '期限日' (Deadline) and 'TODO項目' (TODO Item). The table contains one row with the deadline '2020-03-05' and the item 'てすと'. To the right of the item, there are radio buttons for '未完了' (selected) and '完了' (unselected), followed by a checkbox for '削除' (delete), and a blue '実行' button. At the bottom of the interface, there are three buttons: 'CSVファイルに変換', 'CSVファイルをダウンロード', and 'CSVファイルをアップロードして更新'.

設問8. ログアウト機能を実装します。logout.php にアクセスすると、セッションに保存されているユーザーの情報を削除します。その後、login.php へリダイレクトします。

※ ログアウトしたあとに、index.php など、ログインが必要なページにアクセスしたときに、login.php へリダイレクトされることを確認してください。