

# データベース及び演習 期末レポート

K20089 西宮 銀河,<sup>†1,a)</sup>

**概要:** 本稿は、データベースおよび演習の授業における最終課題についてのレポートである。今回はかんたん備忘録という共有可能な TODO リストを作成した。機能の詳細は以下より述べる。

K20089 NISHIMIYA GINGA,<sup>†1,a)</sup>

## 1. 機能概要

かんたん備忘録は他の人とノートを共有することができる TODO リストアプリケーションである。機能としてはシンプルに備忘録の投稿、投稿のプライベート化、自分が投稿した備忘録のチェックを行うことができる。投稿フォームはスクロールに追従するサイドバーにあるのでどの位置からでも投稿することができる。投稿フォームのチェックボックスにマークをつけると投稿がプライベート化され、最新の備忘録からは確認できなくなるが、マイ備忘録からはいつでも確認することができるので見られたくない投稿はチェックをつけて非表示にすることができる。

そして新規登録の際はユーザネームと重複防止のためのメールアドレス、そしてパスワードを設定することができる。また、これらの項目は必ず入力する必要があり、パスワードは8文字以上、メールアドレスはメールアドレスの形式のみというバリデーションの制限がある。

登録情報はログイン後サイドバーのリンクから変更することができる。入力フォームにはユーザデータが取り込まれて表示される。

## 2. 利用技術

### 2.1 PHP

PHP とは“PHP:Hypertext Preprocessor”の略で、直接 HTML に埋め込んで使用することができる汎用プログラミング言語である。構文は C 言語, Java, Perl から派生しており、それに加えて PHP 独自の構文を組み込むことで PHP 特有の機能を使用できる。PHP はサーバサイド言語

であるので、主に Web サーバ側で動作する。なのでクライアント側からブラウザが HTTP リクエストを送ったとすると、サーバ側が PHP を処理して生成された HTML 文書などをクライアント側に返すという仕組みとなっている。そして PHP はクロスプラットフォーム言語なので Linux, Windows, MacOS はもちろん、NginX や Apache など全ての web サーバ、さらに Microsoft Azure や Amazon AWS などのクラウド環境にも対応している。

また現在の最新バージョンは PHP 8 であり、これは Zend Engine 4 という PHP の実行エンジンの最新バージョンを使用している。これによってオブジェクト指向プログラミングの機能が使えるようになっている。

### 2.2 MariaDB/MySQL

まず、MySQL は 1995 年に開発されたオープンソースソフトウェアのリレーショナルデータベースである。'95 年以降は所有者も管理者も転々と変わっていたが 2010 年に所有権が Oracle 社に移るが、所有権が渡った今でもオープンソースであることは変わっていない。

MySQL はリレーショナルデータベースと呼ばれるデータの保存方法が使われている。リレーショナルデータベースとは 1 つのテーブルに全ての情報を挿入するのではなく、複数のテーブルに分けてそれぞれを関連づけて管理する方法である。例えば名前や住所などの顧客データと購入製品や価格などの注文データを扱うテーブルを作るとすると、もし一つのテーブルに全てのデータを入れておくとした際にデータの重複やデータ削除時の混乱が起こり、検索が難しくなる。その問題に対処するためにリレーショナルデータベースではキーと呼ばれるユニークな ID を利用して異なるテーブル同士を繋げ、データの取り出しや削除などが容易になっている。

<sup>†1</sup> 現在、愛知工業大学  
Presently with AICHI INSTITUTE OF TECHNOLOGY University  
<sup>a)</sup> gingin.sol@pluslab.org

そして、MariaDB はこの MySQL をベースに作られたデータベースである。MySQL と比べてパフォーマンス、安定性、開示性が高く、もっとも有名なリレーショナルデータベースの一つとされている。MySQL との完全な互換性、GPL の採用、さらには開発者コミュニティが豊富なので MariaDB は MySQL の上位互換と言える。

## 2.3 CSS

CSS とは”Cascading Style Sheets”の略で、HTML や XML 文書の見栄えなどを整えるために使われるスタイルシート言語である。HTML 内で指定されたタグや id などを選択で指定して色や段落などの設定ができる。

## 2.4 Laravel

Laravel は MVC 方式の PHP フレームワークである。図 1 に示すようにプロジェクトファイルをデータベースと接続するモデル、HTML 表示を行うためにクライアントにレスポンスを返すビュー、これらモデルとビューをつなぐコントローラの 3 つの役割に分けることでフロントエンドとバックエンドそれぞれの開発がしやすくなっている。CodeIgniter や Yii などの PHP フレームワークや Ruby on Rails などのプログラム言語の基本的な機能を組み込んであるので豊富な機能が使用できることが特徴である。なのでスクラッチ開発でもセキュリティ対策などが簡単に実装できるので容易に Web 開発に取り組むことができる。さらに vue.js などの他のフレームワークからコンポーネントを持ってくることも可能なのでより Web アプリケーションの開発が容易になっている。

また Laravel にはプロジェクトの操作に便利な Artisan コマンドというものが搭載されている。このコマンドは Symfony というフレームワークを使って作られており、これを使用してデータベースの操作、ローカルサーバの立ち上げ、モデル・ビュー・コントローラの生成が簡単にできるようになっている。

さらにプログラムからデータベースに楽にアクセスするために Eloquent と呼ばれる ORM(Object Relational Mapper) が組み込まれている。これによって関数やモデルクラスの使用で SQL の操作が可能となっている。

またビューでは blade と呼ばれる Laravel 独自のテンプレートファイルが使用可能で、これを使用することで php ファイル内で条件に応じた処理や別ファイルの継承などを行って便利に表示をさせることができる。

## 2.5 HTTP

HTTP とは”Hypertext Transfer Protocol”の略でハイパーテキストを転送するためのアプリケーション層プロトコルである。主にサーバ側とクライアント側との通信の目的のために使われるが別の用途でも使用されることがあ

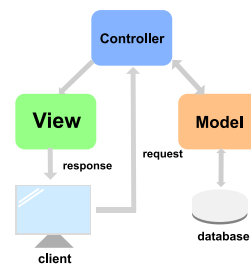


図 1 MVC の動作

る。基本的な動作としてはクライアント側がサーバ側にリクエストを送信するためにポートを開き、サーバ側からレスポンスが帰ってくるまで待機するという仕組みとなっている。また、HTTP は主に TCP/IP 層の上の通信で使用されるが、UDP のようなトランスポート層でも使用されることがある。

## 3. システム設計

### 3.1 システム概要

Laravel を使用して 3 つのコントローラ、2 つのモデル、11 のビューによって構成されている。

SimpleNoteController クラスは主に特にデータベースとの連携などを行わない処理の際に呼び出されるコントローラである。ここではトップページの表示、新規登録の確認画面の表示、ログアウトなどでセッションの操作を行うことがほとんどである。

次に UserController クラスである。ここではユーザーデータの操作を行っており、新規登録の完了、ログイン情報のチェック、登録情報の編集が機能となっている。またデータベースにユーザーデータを挿入する際のパスワードのハッシュ化もここで行っている。

最後に NoteController クラスである。ここでは備忘録の操作を行っている。トップページでの表示の際に id の降順にページネーションしての表示、備忘録の投稿、マイページでの表示が主な機能となっている。さらにプライベート投稿であるかのチェックもここで行っており、プライベート投稿であればフラグを立てたものをデータベースに登録している。そして備忘録を表示する際にはビュー側でこのフラグを検知して非表示にするかを判断している。また、これら UserController と NoteController はバリデーションもここで行っており、それぞれに独自の Request クラスを実装してリクエストが渡された際にクラスの機能に応じたバリデーションを行うことでユーザ入力のルール違反をチェックしている。主にパスワードは 8 文字以上、投稿文字数は 400 文字以内などのルール設定がここで行われ、チェックされている。

そしてモデルは User モデルクラスと Note モデルクラスがあり、それぞれ User モデルが Note モデルの主テーブルであることの宣言やデータベース挿入時のバリデーション



- モデル
  - User.php
  - Notebook.php
- ビュー
  - index.blade.php
  - login.blade.php
  - register.blade.php
  - registerConfirm.blade.php
  - registerComplete.blade.php
  - editUser.blade.php
  - editConfirm.blade.php
  - editComplete.blade.php
  - mypage.blade.php
  - uploadConfirm.blade.php
  - template.blade.php
- リクエストクラス
  - EditRequest.php
  - RegisterRequest.php
  - UploadRequest.php

SimpleNoteController クラスでは特に深くデータベースとの連携がない処理を行っているので index メソッドや login メソッド, register メソッドではそのままトップページや登録画面を表示するのみとなっている。また, regist.confirm メソッドでは新規登録の確認画面の際に html の方に表示するデータの転送とセッションへの格納により入力したデータを確認として表示させている。

また logout メソッドではセッションに格納してあるログイン ID とユーザネームを消してリダイレクトすることでログアウト機能を実現している。

UserController クラスではユーザ情報をモデルとビュー間でやり取りする処理が行われている。regist.complete メソッドではユーザ情報をデータベースに格納させる処理を行っている。前ページからセッションで受け取ったユーザデータにバリデーションを行ったのちにユーザネームとメールアドレス, さらにパスワードをハッシュ化した状態で格納させている。そして login.check メソッドではログイン時のデータチェックを行っている。データベース内から入力メールアドレスと登録メールアドレスを比較して一致したものがない, またはメールアドレスは一致していたとしてもパスワードが一致していない場合はその趣旨のエラーをビューに返すようにしている。

edit メソッドは登録情報を編集する際に入力フォームに登録データを入力しておくために呼び出され, edit.confirm メソッドで regist.confirm メソッドと同じく確認画面のためのデータ転送, セッションへの格納を行っている。そして edit.complete メソッドで入力された変更情報を再挿入している。

NoteController クラスでは備忘録情報をモデルとビュー

間でやり取りする処理が行われている。upload メソッドではトップページで入力された備忘録情報を受け取り, プライベート投稿であるかのフラグを boolean に変換した状態でセッションに格納した上で確認画面に飛ばしている。そして upload.complete メソッドで実際に入力されたデータをデータベースに格納するという仕組みになっている。ユーザデータの格納と違うところはプライベート投稿であるかの boolean を判別する部分のみで他はほとんど同じとなっている。そして mypage メソッドではマイページで表示する備忘録の条件指定を行っている。現在ログインしているユーザの ID のみで条件を絞り, 降順で表示させることで自分の投稿のみを表示させるようにしている。

User モデルクラスでは自動繰上げである id カラムの保護, notebooks テーブルが従テーブルであることの宣言, バリデーションを指定している。また, Notebook モデルクラスでは自動繰上げ, 自動更新である id カラムと created\_at カラムの保護, users テーブルが主テーブルであるのことの宣言, 同じくバリデーションを指定している。

リクエストクラスは確認画面に飛ぶ前に入力情報が条件に合っているかのバリデーション機能を使用するために実装している。今回は新規登録, 登録情報の変更, 備忘録の投稿の際にバリデーションを使用している。regist.confirm メソッドでは RegisterRequest クラスを呼び出して新規登録時のバリデーション, edit.confirm メソッドでは EditRequest クラスを呼び出して登録情報変更時のバリデーション, upload メソッドでは UploadRequest クラスを呼び出して備忘録投稿時のバリデーションを行って, 機能ごとそれぞれのバリデーションを実装している。

### 3.5 変数設計

今回はセッション変数と POST 変数を多く使用してデータのやり取りをしている。ここで使用されている変数の一覧は以下になっている。

- セッション変数
  - \*user\_name : ユーザネームの格納
  - \*mail : メールアドレスの格納
  - \*password : パスワードの格納
  - login\_id : ログインしているユーザ ID の格納
  - login\_user : ログインしているユーザネームの格納
  - \*title : 備忘録の件名の格納
  - \*texts : 備忘録の本文の格納
  - \*isPrivate : プライベート投稿であるかのフラグの格納
- POST 変数
  - user : ユーザネームの格納
  - mail : メールアドレスの格納

- pass : パスワードの格納
- isLoginError : ログインエラーである時のフラグの格納
- notes : 備忘録データの格納
- title : 備忘録の件名の格納
- texts : 備忘録の本文の格納
- isPrivate : プライベート投稿であるかのフラグの格納

また \* マークの付いているセッション変数は Laravel の機能であるフラッシュデータと呼ばれるものであることを示している。これは常にサーバに残り続ける通常のセッション変数とは違い、リダイレクト後に自動的にセッションから削除される一時的な変数である。このフラッシュデータを使用してデータベースに格納するデータを保存して格納完了後にセッションから消去されるようにしている。

さらに login\_id はログインしている状態を保つために使用されており、login\_user は備忘録投稿の際のユーザ名の表示のために使用されている。なのでログアウトの際にはこれらの変数を削除することでログアウト機能を実装している。

そして POST 変数は主に新規登録、登録内容の編集、投稿の確認画面にデータを転送する際に使用されている。他にも notes という全ての備忘録データをビューに転送する変数や isLoginError というログインエラーであった時のみ転送されるフラグ変数も導入されている。

## 4. 実装

### 4.1 実装環境

本システムの実装環境は以下の通りである。

- MariaDB 10.4.21
- PHP 8.1.4
- Laravel 4.2.10
- Composer 2.3.9

### 4.2 環境設定

今回の開発では XAMPP は一切使わずに全て Laravel で環境設定を行っている。Laravel の env ファイルにデータベースやメールの設定を追記することで使用可能になる。今回は mysql で localhost を使用の上、ポート番号は 3306, 使用データベースは simple\_notebooks, ユーザネームはテスト用のユーザとして ginging を設定している。

### 4.3 動作検証

ログインを完了させるまでの画面遷移は図 3 のようになっている。まずトップページのサイドメニューからログインを選ぶと、メールアドレスとパスワードを入力するフォームが表示されるので登録情報を入力することでログインが完了し、トップページに遷移する。また、入力し

たユーザ情報が存在していなかったりメールアドレスやパスワードが間違っていたりすると「メールアドレスかパスワードが違います」と言ったエラーメッセージが表示される。



図 3 ログイン完了までの流れ

そして備忘録の投稿を完了させるまでの画面遷移は図 4 のようになっている。トップページのサイドバーに入力フォームが設置されているので、件名と本文を入力して送信すると確認画面に飛ぶことができ、そこから投稿完了となる。このときに件名と本文が入力されていなければその趣旨のエラー文が表示される。



図 4 備忘録投稿完了までの流れ

最後に新規登録完了までの画面遷移は図 5 のようになっ



ている。ログイン画面の左上のリンクから新規登録をすることができる。ユーザネームとメールアドレス、パスワードと確認用のパスワードを入力することで登録情報の確認画面に遷移することができる。この際、パスワードが8文字未満であったり、メールアドレスがアドレスの形式でない、またこれらが入力されていない場合はこの場でエラー文が表示される。確認が終わり、次に進めば新規登録完了となり、登録した内容でログイン可能となる。



図 5 新規登録完了までの流れ

## 5. まとめ

今回はかんたん備忘録というシンプルな TODOWeb アプリケーションを作成した。Laravel について勉強しながら同時に制作したが、思い通りの動作をさせることができた。この制作で Laravel についての理解をさらに深めることができ、今後の Web アプリケーションの制作にも活かせると考えたので大きな勉強になった。しかしその反面、Laravel の認証などの様々な機能を完全に使いこなすことはできなかったので今後制作するときはしっかり学んだ上で作りたい。

## 参考文献

- [1] PHP Group . "一般的な情報" . php . <https://www.php.net/manual/ja/faq.general.php> . (閲覧日:2022/05/29)
- [2] PHP Group . "Zend API: PHP のコアをハックする" . php . <http://php.adamharvey.name/manual/ja/internals2.ze1.zendapi.php> . (閲覧日:2022/05/29)
- [3] PHP TUTORIAL . "What is PHP" . phptutorial.net . <https://www.phptutorial.net/php-tutorial/what-is-php/> . (閲覧日:2022/05/29)
- [4] KINSTA . "MySQL とは？初心者にわかりや

- すい説明" . KINSTA . (更新日:2020/07/03) . <https://kinsta.com/jp/knowledgebase/what-is-mysql/> . (閲覧日:2022/06/01)
- [5] MariaDB Foundation . "MariaDB Server: The open source relational database" . MariaDB Foundation . <https://mariadb.org/> . (閲覧日:2022/06/01)
  - [6] Mark Smallcombe . "MariaDB vs MySQL: 徹底比較" . integrate.io . (更新日:2020/09/03) . <https://www.integrate.io/jp/blog/mariadb-vs-mysql-everything-you-need-to-know-ja/> . (閲覧日:2022/06/01)
  - [7] MDN contributors . "CSS: カスケーディングスタイルシート" . mdn web docs . (更新日:2021/07/18) . <https://developer.mozilla.org/ja/docs/Web/CSS> . (閲覧日:2022/06/01)
  - [8] Tutorials Point . "Laravel Overview" . tutorialspoint simply easy learning . [https://www.tutorialspoint.com/laravel/laravel\\_overview.htm](https://www.tutorialspoint.com/laravel/laravel_overview.htm) . (閲覧日:2022/07/08)
  - [9] MDN contributors . "HTTP" . mdn web docs . (更新日:2021/09/18) . <https://developer.mozilla.org/ja/docs/Web/HTTP> . (閲覧日:2022/06/01)