第１２回　総合問題②　＋　ログインによる表示切替

**※総合問題のため、11回目までの復習範囲を含む。**

**※ファイルの置き場所なども、11回目までの講義資料から調べて配置すること。**

**【課題は６問】**

**ユーザーが投稿できる簡易SNSを作成する。**

**■ログイン認証にはBreezeを用いる**。

ログイン認証自体は実装する必要はない。

**※ただし、第5回目の講義資料「５－５」を実施していることを前提とする。**

■ＵＲＬは「<http://localhost:8000/kadai12_1>」でアクセスするものとする。

**【使用ファイル】**

■配布

・Kadai12\_1.blade.php

・originalTheme.scss　　※スタイルシート

■編集（既存ファイルに追加、もしくは一部編集）

・navigation.blade.php　※共通部分ナビゲーション

・web.php

■新規作成

・Kadai12\_1Controller

・create\_snspost\_table.php　　※マイグレーションファイル

・SnspostModel.php　　　　　　　※snsPostsテーブルのモデル

課題１２－１　マイグレーションファイルとモデルの作成

1. 以下の仕様のテーブルを作成すること。
2. テーブル用のモデルクラスも作成すること。

＊テーブル定義にはマイグレーションファイルを用いること。MySQLで直接createTableしたり、プログラムからSQL文を発行したりすることはNGとする。

■ＤＢ　新規作成

snspostsテーブル

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 型 | 制約 | 備考 |
| id | AUTO\_INCREMENT | UNSIGNED |  |
| user\_id | INT | NOT NULL  FK:users.id | ユーザーid |
| comment | TEXT | NOT NULL | 本文 |
| img\_path | VARCHAR(255) | NULL 許可 | 画像ファイル名 |
| created\_at | TIMESTAMP | CURRENT\_TIMESTAMPを  デフォルト値として使用 | 作成日時 |
| updated\_at | TIMESTAMP | ●CURRENT\_TIMESTAMPを  デフォルト値として使用  ●レコードが更新されたときに  CURRENT\_TIMESTAMPを使用 | 更新日時 |
| deleted\_at | ソフトデリート用  TIMESTAMPカラム | ------ | 論理削除用  （＊） |

課題１２－２　コントローラの処理（GETメソッド）

1. kadai12\_1ビューが下記のURLで表示できるようにすること。

※DBからの情報抽出処理は現時点では行わない。

1. URL「<http://localhost:8000>」でアクセスしたときも、今回のURLにリダイレクトするように設定すること。
2. 設定が完了したら、この時点で実行して、下記のエラーが出ることを確認すること。

下記エラーの解消は次の１２－３で行います。

（下記エラーのみ現時点での正常動作。その他エラーは解消しておくこと）

＊エラー内容

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

■Kadai12Controller.phpindexメソッド仕様

**※コントローラのファイルは[ターミナル]からコマンドで作成すること。（以降の課題も同じ）**

|  |  |
| --- | --- |
| アクセス修飾子 | public |
| メソッド名 | index |
| 引数 | なし |
| 処理 | viewヘルパー関数を使用し、Kadai12\_1.blade.phpを　呼び出す。 |

　■ルーティング設定①　＊続いて②もあります。

|  |  |
| --- | --- |
| リスエスト形式 | get |
| URI | http://localhost:8000**/kadai12\_1** |
| 呼び出しクラス・メソッド | Kadai12Controller の　index　メソッド |
| ルーティング名 | kadai12\_1.index |

　■ルーティング設定②

**※既存のルーティングへリダイレクト処理を追加**

|  |  |
| --- | --- |
| リスエスト形式 | get |
| URI | http://localhost:8000 |
| リダイレクト | リダイレクトのルーティング名「kadai12\_1.index」 |

【補足説明】　課題１２－２　実装時点でのエラーについて

課題１２－２を実行すると、エラーが出ています。

＊エラー内容

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

＊エラー箇所

　今回、kadai12\_1.blade.phpでナビゲーションのテンプレートを使用しています。

　ナビゲーションでは、ユーザー名を表示する処理をしていますが、そもそも現状ではログインしていないので、このエラーが出ます。

テキスト が含まれている画像

自動的に生成された説明

　　ログインしていない（ログアウト状態）の場合でも、ナビゲーションを表示できるように、一部変更を行います。変更は課題１２－３で行います。

課題１２－３　ナビゲーション　テンプレートの修正

ナビゲーションのテンプレートをログイン状態により、分岐できるように変更します。

通常の表示とレスポンシブ対応の表示の2箇所の変更が必要です。

　＊変更点

　　●ログイン時→デフォルトのナビゲーション

　　●ログアウト時→ナビゲーションのメニューを「ログイン」「新規登録」とする

　　　＊ログアウト時のメニューで「ログイン」「新規登録」それぞれをクリック時、

各画面へ遷移し、デフォルトの機能（ログインやユーザー登録）が正しく使用できること。

　■対象ファイル

|  |  |
| --- | --- |
| ナビゲーション | resources\views\layouts\navigation.blade.php |

　■追加①　全容　※各行の説明は後述。

テキスト

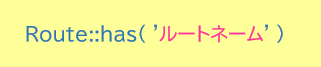
自動的に生成された説明

　■追加①　詳細内容

　　21行目辺りドロップダウンの表示に追加。

1. 23行目：「ログインのルートネームが存在した場合」の条件を追加。

■Route::hasメソッド



戻り値：ルーティング情報が存在すればtrue、しなければfalse

1. 2４行目：ログインしている場合の条件追加。

<x-dropdown>タグ内（デフォルトで記述あり）はログインしている場合のみ表示。

■@authディレクティブ



　戻り値：ログイン認証が行われてログインしている場合はtrue、そうでなければfalse

1. 59行目～67行目：ログインしていない場合の処理追加。（下記のコード）

「register」リンクの表示にもルーティングネームregisterが存在するかの条件を付加している。

@else

    <a href="{{ route('login') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Log in</a>

    @if (Route::has('register'))

        <a href="{{ route('register') }}"

            class="ml-4 text-sm text-gray-700 dark:text-gray-500 underline">Register</a>

    @endif

@endauth

1. ６８行目：23行目の条件の終了範囲

　■追加②　全容

　　レスポンシブ対応のデザインにもログインによる分岐を追加します。

**＊下記の行数は追加①のコードを追加後の行数。**

**＊追加している内容は追加①と同じ。**



【動作確認】　課題１２－３．完了時点

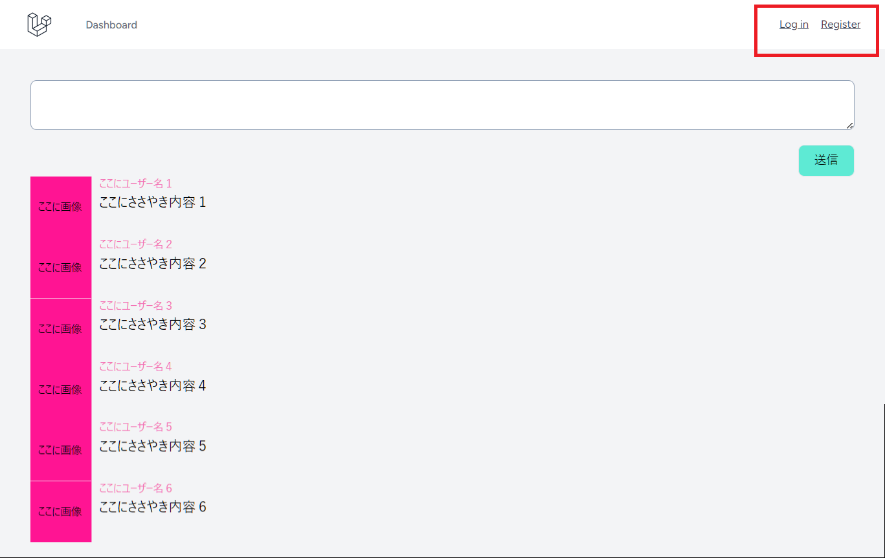
1. 「<http://localhost:8000>」にアクセス時、リダイレクトで、

「<http://localhost:8000/kadai12_1>」が表示されること。

1. ①の実行時、エラーが解消され、下記の画面が表示されること。

■ログアウト状態（初回）

　　＊追加したナビゲーションメニュー（赤枠部分）が表示されること。

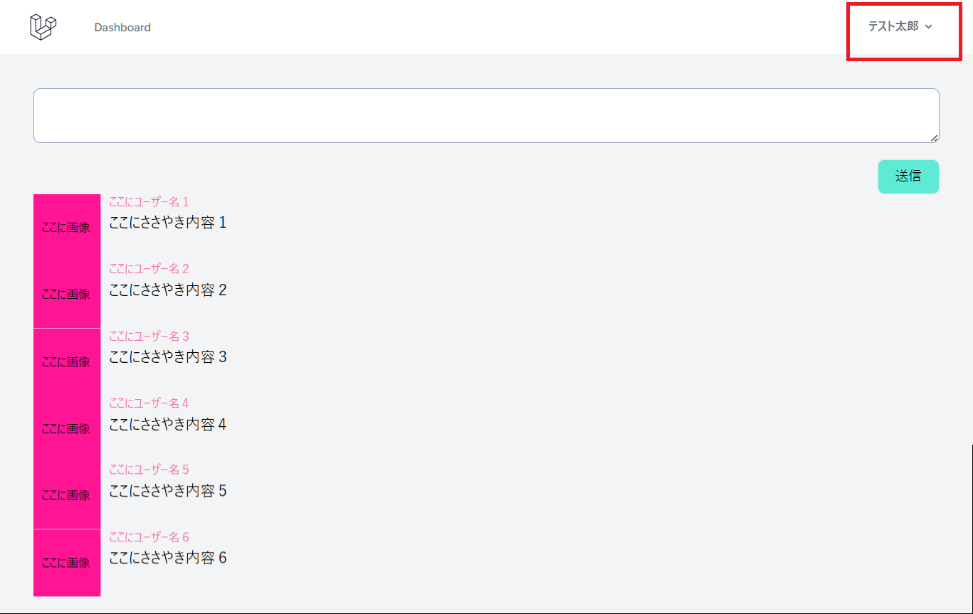


1. 「Register」から新規ユーザー登録を行う。登録が完了したらダッシュボード画面へ遷移するので、再度「<http://localhost:8000>」にアクセスする。
2. ログイン後のナビゲーションメニューは下記のように表示されていることを確認。

また、ユーザー名をクリックすると、ポップアップメニューが表示されること。

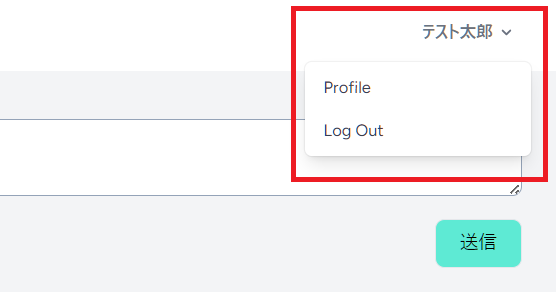
■ログイン後

　＊作成したユーザー名が表示されていること。（赤枠部分）



　　■ユーザー名押下時

　　　　＊ポップアップメニューが出ること。



1. ログアウト押下時、ナビゲーションもログアウト時の表示に切り替わること。

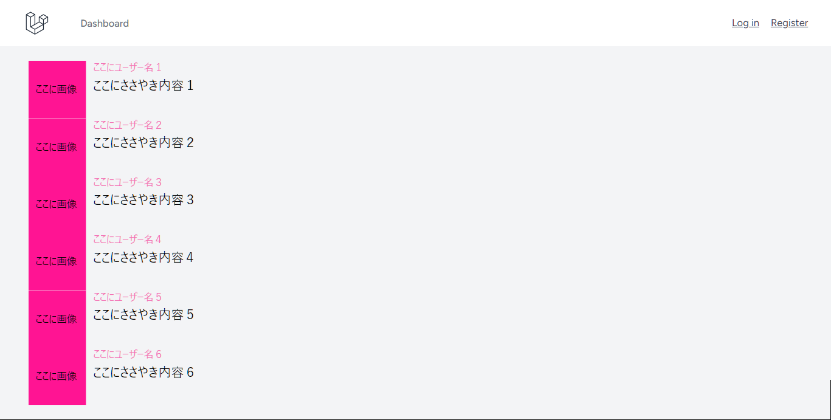
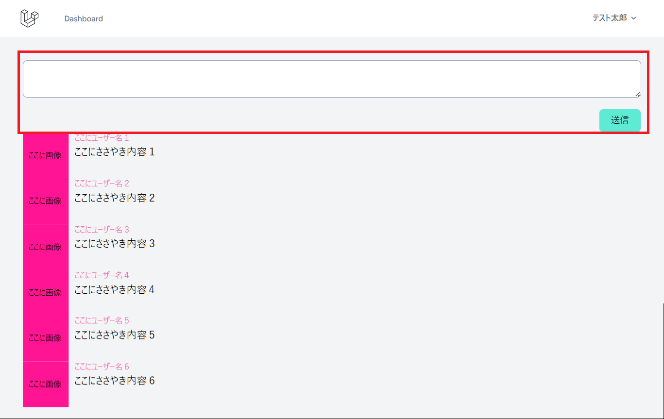
（ログイン時の画面からログアウト時の画面に切り替わる）

課題１２－４

ログアウト時、コメントの送信ができないようにすること。

kadai12\_1.blade.phpを編集。

■ログアウト時　　　　　　　　　　　　　　　　　　　　　　　　　　　　■ログイン時　※赤枠部分が表示される

課題１２－５

コメントを登録できるようにする。

■Kadai12Controller.php「送信」ボタン押下時メソッド仕様

|  |  |
| --- | --- |
| アクセス修飾子 | public |
| メソッド名 | store |
| 引数 | Request $request |
| 処理 | リクエストデータをsnspostsテーブルに登録する。  ＊useridはログインしているユーザーのidを取得して格納すること。 |

　■ルーティング設定

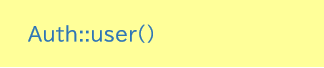
|  |  |
| --- | --- |
| リスエスト形式 | post |
| URI | http://localhost:8000**/kadai12\_1** |
| 呼び出しクラス・メソッド | Kadai12Controller の　storeメソッド |
| ルーティング名 | kadai12\_1.store |

課題１２－６

Kadai12\_1.blade.php表示時、固定値を表示しているので、snspostsテーブルのデータが表示できるように修正すること。

■Auth::user

ログインユーザー情報取得を取得



■サンプルコード

　＊Auth::userで取得できる情報を表示して確認。

public function store(Request $request)

{

    //TODO:処理未実装

    //TODO:必要に応じて下記

    $userData = Auth::user(); //ログインユーザー情報取得（連想配列）

    //TODO:debug

    dd($userData);

}

第12回目は以上です。