Aplikacje internetowe 1 (AI1) – laboratorium nr 5

Framework Laravel – wstęp

Początek laboratorium:

• na pulpicie utworzyć folder `lab5` oraz otworzyć go w *VSCode*.

Zadania (*Laravel*):

Zadanie 5.1: Zapoznać się z następującymi zagadnieniami: • o framework'u *Laravel*, • wykorzystanie *Laravel'a* do tworzenia aplikacji *MVC* z widokami będącymi szablonami renderowanymi przez silnik *Blade*, • wykorzystanie *Laravel'a* do tworzenia backend'owych części aplikacji – *backend API* z wykorzystaniem formatu wymiany danych: *JSON*, • zastosowanie *MVC* we framework'u *Laravel*.
https://laravel.com/docs/11.x/#meet-laravel https://laravel.com/docs/11.x/#laravel-the-fullstack-framework https://laravel.com/docs/11.x/#laravel-the-api-backend https://fkrihnif.medium.com/understanding-the-mvc-architecture-in-laravel-a-comprehensive-guide-8f620cc139b6 (w przypadku osiągnięcia limitu artykułów z tego portalu, otworzyć link w oknie prywatnym)
Zadanie 5.2: Otworzyć terminal *cmd (Command Prompt)* w *VSCode*. Za pomocą *Composer'a* utworzyć nowy projekt *Laravel'a* w bieżącym katalogu (nie można pominąć kropki na końcu polecenia!). Zapoznać się z strukturą utworzonych katalogów. Uruchomić serwer deweloperski *php* dla przy użyciu komendy `serve` *artisan'a*.

`composer create-project laravel/laravel .`
 	
`php artisan serve`
https://laravel.com/docs/11.x/structure
 	
W przeglądarce internetowej przejść pod adres: http://localhost:8000

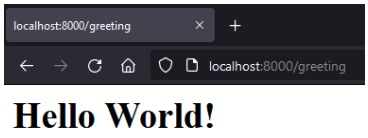
Zadanie 5.3: Zapoznać się z *trasowaniem* (*routing*). Sprawdzić, gdzie została ustawiona *trasa* (*route*) dla strony wyświetlonej w poprzednim zadaniu (`'/'`). Odnaleźć plik z kodem widoku odpowiedzialnego za przygotowanie treści tej strony.
https://laravel.com/docs/11.x/routing routes\web.php https://laravel.com/docs/11.x/views resources\views\welcome.blade.php

Zadanie 5.4:
Ustawić nową *trasę* tak, aby po przejściu pod *ścieżkę* (*path*) `/greeting` na stronie został wyświetlony napis „*Hello world!*”. Następnie zmienić go, tak aby był nagłówkiem `h1`. Otworzyć drugą kartę terminala *cmd (Command Prompt)*. Wyświetlić w niej wszystkie obecnie ustawione trasy. Jaka *metoda HTTP* jest ustawiona dla tej *trasy*?

https://laravel.com/docs/11.x/routing#basic-routing

Po wykonaniu modyfikacji w kodzie aplikacji nie ma potrzeby uruchamiać serwera ponownie.

http://localhost:8000/greeting → Hello world!



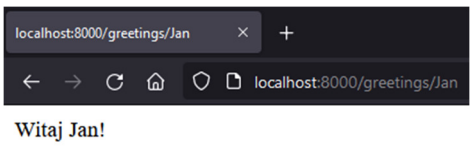
https://laravel.com/docs/11.x/routing#the-route-list

```
php artisan route:list
```

Zadanie 5.5:
Ustawić nową *trasę* tak, aby po przejściu pod *ścieżkę* `/greetings/{name}` [gdzie `{name}` występuje w roli wymaganego *parametru w ścieżce* (*path param*)] na stronie został wyświetlony napis: „*Witaj ...!*” (imię tej osoby).
Następnie dokonać zmiany tak, aby parametr był opcjonalny `{name?}`, a w przypadku nie podania imienia domyślnie było nim „*nieznajomy*”.

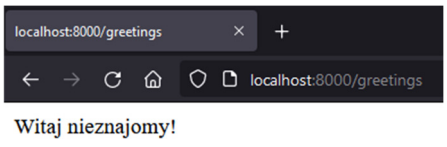
https://laravel.com/docs/11.x/routing#required-parameters

http://localhost:8000/greetings/Jan → Witaj Jan!



https://laravel.com/docs/11.x/routing#parameters-optional-parameters

http://localhost:8000/greetings → Witaj nieznajomy!



Zadanie 5.6:
Wykonać poniższą komendę w celu utworzenia nowego *kontrolera*.
Przejść do nowo utworzonego pliku.

https://laravel.com/docs/11.x/controllers

```
php artisan make:controller TemperatureController
```

app\Http\Controllers\TemperatureController.php

Zadanie 5.7:

Do kontrolera utworzonego w poprzednim zadaniu dodać nową funkcję o nazwie `ctf`, przyjmującą jeden opcjonalny parametr typu *float* o nazwie `c`, której zadaniem będzie przeliczać stopnie *Celsjusza* na *Fahrenheita*.
W przypadku wywołania funkcji:
• bez parametru ma zostać zwrócony komunikat „*Nie podano wartości*”,
• z podaniem ma być zwracany komunikat z informacją np. „*4.0°C to 39.2°F*”.
W pliku *web.php* ustawić *trasę* na *ścieżkę* `temperature/ctf` z parametrem `c` dla tej funkcji kontrolera oraz na górze tego pliku dodać import do pliku kontrolera.

http://localhost:8000/ctf → Nie podano wartości
http://localhost:8000/ctf/4.0 → 4.0°C to 39.2°F

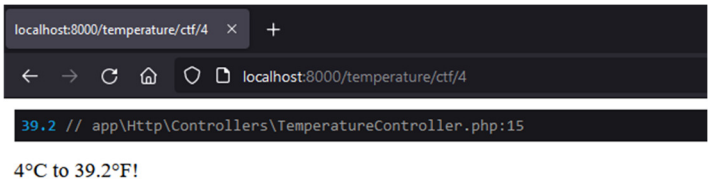
https://laravel.com/docs/11.x/controllers#basic-controllers

```
use App\Http\Controllers\TemperatureController;
```

Zadanie 5.8:

Zapoznać się z *helper'ami* dotyczącymi debugowania aplikacji: `dd()` i `dump()`.
Sprawdzić ich działanie, poprzez ich użycie wewnątrz funkcji `ctf` z poprzedniego zadania.

https://laravel.com/docs/11.x/helpers#method-dd
https://laravel.com/docs/11.x/helpers#method-dump



https://laraveldaily.com/post/7-tricks-with-dd-in-laravel

Zadanie 5.9: *

Wkleić poniższy fragment do pliku *web.php*, zapoznać się z wybranymi elementami zawartymi w `request`.

https://laravel.com/docs/11.x/requests

```
use Illuminate\Http\Request;

//...

Route::get('/zad9', function (Request $request) {
    $br = "<br>";
    $r = $request->path() . $br .
        $request->url() . $br .
        $request->fullUrl() . $br .
        $request->method(). $br .
        $request->isMethod('post'). $br .
        $request->header('User-Agent'). $br .
        $request->ip();
    return $r;
});
```

Zadanie 5.10:
Wkleić poniższy fragment do pliku *web.php*.
Zapoznać się z parametrami (oraz ich wartościami) w *ciągu zapytania* (ang. *query string* lub *query params*).
Wyjaśnić różnice pomiędzy *parametrami ścieżki* (*path params*) a parametrami przekazywanymi w *ciągu zapytania* (*query params*). *

https://laravel.com/docs/11.x/requests

https://en.wikipedia.org/wiki/Query_string

```
use Illuminate\Http\Request;

//...

Route::get('/zad10', function (Request $request) {
    $br = "<br>";
    $r = print_r($request->all(), true) . $br .
        $request->query('a') . $br .
        $request->query('b', 'brak b') . $br .
        print_r($request->query(), true) . $br .
        $request->a . $br .
        $request->has(['a', 'b']) . $br .
        $request->filled(['a']) . $br;
    return $r;
});
```

http://localhost:8000/zad9?a=45
http://localhost:8000/zad9?a=45&b=pies
http://localhost:8000/zad9?a=45&b=pies&c=3.5

Zadania (*Laravel*):

Zadanie 5.11:
Wykonać następujące komendy:

```
mkdir public\img
mkdir storage\app\public\img
mkdir resources\views\shared

php -r "touch('resources\views\zad13.blade.php');"
php -r "touch('resources\views\index.blade.php');"
php -r "touch('resources\views\shared\head.blade.php');"
php -r "touch('resources\views\shared\navbar.blade.php');"
php -r "touch('resources\views\shared\footer.blade.php');"
```

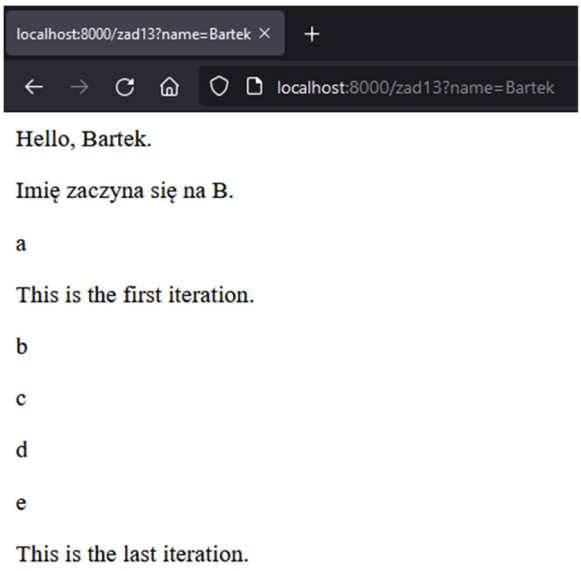
Zadanie 5.12:
Wkleić poniższy fragment do pliku *web.php*:

```
Route::get('/zad13', function (Request $request) {
    $name = $request->name;
    $arr = ['a', 'b', 'c', 'd', 'e'];
    return view('zad13', ['name' => $name, 'arr' => $arr]);
});
```

Zadanie 5.13:

W szablonie *resources\views\zad13.blade.php* napisać kod, który:
• jest *Blade'owym* komentarzem np. „*To jest komentarz.*" (w pierwszej linijce),
• jeśli zmienna **name**:
– zawiera wartość to wyświetlane jest: „*Hello, imię*" (z tej zmiennej),
– jeśli nie to: „*Brak imienia*",
• jeśli wartość zmiennej **name** zaczyna się od „*B*", to:
– wyświetlane jest: „*Imię zaczyna się na B*",
– jeśli nie to: „*Nie zaczyna się na B*", *
• wyświetlana jest zawartość tablicy **arr**, każdy element w osobnym **<p>**, z informacją:
– „*To jest pierwsza iteracja przy pierwszym elemencie*" oraz analogiczną informację przy ostatnim elemencie;
– w razie, gdyby tablica **arr** nie zawierała żadnego elementu wyświetlić stosowną informację. *
Następnie zmieniać zawartość tablicy **arr** i podawać różne imiona w parametrze *name* w ciągu zapytania.

https://laravel.com/docs/11.x/blade#comments
https://laravel.com/docs/11.x/blade#if-statements
https://laravel.com/docs/11.x/blade#blade-directives
https://laravel.com/docs/11.x/blade#loops



http://localhost:8000/zad13
http://localhost:8000/zad13?name=Jan
http://localhost:8000/zad13?name=Bartek

Zadanie 5.14:

Pobrać archiwum **pliki.zip**.
Do folderu *public\img* skopiować obrazki dotyczące karuzeli.
Do folderu *storage\app\public\img* skopiować obrazki dotyczące kart.
Wykonać poniższą komendę.

https://laravel.com/docs/11.x/filesystem#the-public-disk

```
php artisan storage:link
```

Zadanie 5.15:
Ustawić nową *trasę* tak, aby po przejściu pod poniższy adres wyświetlała się strona z szablonu *index.blade.php*.
Skopiować zawartość pliku *lab.html* do *index.blade.php*.
Poprawić ścieżki do zdjęć w kartach, tak aby były ładowane z folderu *img* ze *storage*.

http://localhost:8000/trips

```
<img src="{{asset('storage/…')}}">
```

Zadanie 5.16:
Zapoznać się z `@include`.
Wydzielić poszczególne sekcje strony do osobnych szablonów (w folderze *shared*), a następnie „*załączyć*" je do *index.blade.php* (tak żeby zawartość i układ strony był jak na wcześniejszych laboratoriach).

https://laravel.com/docs/11.x/blade#including-subviews

```
@include('shared.…')
```

Zadanie 5.17:
Uzupełnić załączanie *head* o dodatkową informację o tytule bieżącej strony.
Ustawić zawartość *tagu* tytułu na podstawie tej zmiennej.

```
@include('shared.head', ['pageTitle' => 'Wycieczki górskie'])

<title>{{ $pageTitle }}</title>
```

Zadanie 5.18: *

Na podstawie opisu słownego zawierającego wymagania postawione dla nowej bazy danych:

Należy skonstruować bazę danych, pozwalającą na przechowywanie danych dotyczących ofert tych wycieczek. Baza ma przechowywać dane na temat krajów: *nazwa*, *kod* (ISO 3166), *waluta*, *powierzchnia całkowita* (w km²), *język urzędowy*. Baza ma przechowywać dane na temat wycieczek: *nazwa*, *kontynent na którym się odbywa*, *okres trwania wycieczki* (w dniach), *opis miejsca w którym odbywa się wycieczka*, *cena wycieczki* (w złotych z groszami), nazwa obrazka danej wycieczki, oraz *kraj w którym się odbywa*.

a) napisać skrypt *SQL* tworzący tabele (uwzględniając powiązania pomiędzy nimi), posiadające *id* z *autoinkrementacją*, nie korzystać z kreatora w *GUI*,

b) utworzyć nową bazę `trips` w *phpMyAdminie* oraz uruchomić wcześniej napisany skrypt *SQL*, następnie wygenerować diagram *ERD*,

c) wstawić do bazy dane wycieczek i krajów (w oparciu o dane z poprzednich laboratoriów), także pisząc skrypt, nie korzystać z kreatora.

---

* – zadania/podpunkty do samodzielnego dokończenia/wykonania,

* – zadania/podpunkty dla zainteresowanych.

Po zakończonym laboratorium należy skasować wszystkie pobrane oraz utworzone przez siebie pliki z komputera w sali laboratoryjnej.

Wersja pliku: v1.0