Aplikacje internetowe 1 (AI1) – laboratorium nr 6

Laravel – Eloquent ORM, migracje, seedy

Początek laboratorium:

• w *XAMPP* uruchomić *Apache* oraz *MySQL*, następnie przejść do *phpMyAdmin*,

• pobrać na pulpit archiwum `Lab006_AI1_start.zip`, w którym umieszczony jest projekt startowy do wykonania zadań oraz rozpakować to archiwum,

• przejść do rozpakowanego folderu oraz w przypadku posiadania innych ustawień niż domyślne (np. połączenia z bazą), wykonać ich zmianę w pliku `.env.example`,

• uruchomić skrypt `start.bat` (*Windows,* 2x kliknięciem) lub `start.sh` (inne systemy, przez polecenie *bash start.sh*).

Zadania (*Laravel*):

Zadanie 6.1: *   Wyjaśnić działanie poleceń zawartych w pliku *start.bat*.
https://getcomposer.org/doc/01-basic-usage.md#installing-from-composer-lock   https://laravel.com/docs/11.x/seeding#running-seeders   https://laravel.com/docs/11.x/encryption   https://laravel.com/docs/11.x/filesystem#the-public-disk
Zadanie 6.2:   Otworzyć terminal *cmd (Command Prompt)* w *VSCode*.   Uruchomić serwer deweloperski *php* dla przy użyciu komendy `serve` *artisan'a*.
`php artisan serve`   W przeglądarce internetowej przejść pod adres: http://localhost:8000
Zadanie 6.3:   Zapoznać się z następującymi zagadnieniami: • *modele (app/Models)*,   • *migracje (database/migrations)*,   • *seeder'y (database/seeders)*,   • konwencje nazewnictwa zaakceptowane przez społeczność *Laravel'a*.   Odnaleźć w projekcie pliki dotyczące powyższych zagadnień dotyczących modelu *Country*.   Otworzyć drugą kartę terminala *cmd (Command Prompt)* w *VSCode*.   Wygenerować nowy model *wycieczek* wraz wygenerowaniem dla niego: *migracji*, *seedera* oraz *kontrolera*.
https://laravel.com/docs/11.x/eloquent#generating-model-classes   https://laravel.com/docs/11.x/migrations   https://laravel.com/docs/11.x/seeding#main-content   https://github.com/alexeymezenin/laravel-best-practices?tab=readme-ov-file#follow-laravel-naming-conventions   `php artisan make:model Trip -msc`

Zadanie 6.4:

Na podstawie wymagań z ostatniego zadania poprzedniego laboratorium oraz uwzględniając jego przykładowe rozwiązanie dostępne w pliku *trips.sql* uzupełnić *migrację* (*database/migrations/..._create_trips_table.php*) tworzącą tabelę `trips` o wymagane kolumny wraz z ich typami danych.
Następnie uruchomić wykonanie *migracji* oraz sprawdzić zawartość bazy w *phpMyAdminie* (odświeżać go po każdym wykonaniu).
W przypadku zauważenia pomyłki, cofnąć ostatnią *migrację*, wykonać poprawę i końcowo wykonać znów *migracje*. Ewentualnie można użyć *fresh*.

https://laravel.com/docs/11.x/migrations#migration-structure
https://laravel.com/docs/11.x/migrations#running-migrations
https://laravel.com/docs/11.x/migrations#rolling-back-migrations
https://laravel.com/docs/11.x/migrations#drop-all-tables-migrate

https://laravel.com/docs/11.x/migrations#tables
https://laravel.com/docs/11.x/migrations#foreign-key-constraints

```
php artisan migrate
php artisan migrate:rollback
php artisan migrate:fresh
```

Zadanie 6.5:

Na podstawie przykładowego rozwiązania dostępnego w pliku *trips.sql* został przygotowany *seeder* dotyczący *krajów* (plik *database/seeders/CountrySeeder.php*).
Uzupełnić także *database/seeders/TripSeeder.php* aby do tabeli `trips` były dodawane przykładowe dane (na razie dwóch pierwszych wycieczek, później wszystkich *).

https://laravel.com/docs/11.x/seeding#writing-seeders
https://laravel.com/docs/11.x/migrations#toggling-foreign-key-constraints
https://laravel.com/docs/11.x/eloquent#deleting-models

Zadanie 6.6:

Zapoznać się z zawartością pliku *app/Models/Country.php*.
Uzupełnić *model wycieczek app/Models/Trip.php* wraz z uwzględnieniem:
• tabela *wycieczek* ma zawierać kolumny *created_at* i *updated_at*,
• domyślna wartość dni dla wycieczki to 7,
• ustalić wszystkie właściwości jako `fillable` oprócz *id* tabel,
• zrealizować powiązanie pomiędzy modelami *krajów* i *wycieczek* (obustronnie).

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

https://laravel.com/docs/11.x/eloquent#timestamps `$timestamps`
https://laravel.com/docs/11.x/eloquent#default-attribute-values `$attributes`
https://laravel.com/docs/11.x/eloquent#mass-assignment `$fillable`

https://laravel.com/docs/11.x/eloquent-relationships
https://laravel.com/docs/11.x/eloquent-relationships#one-to-many-inverse
https://laravel.com/docs/11.x/eloquent-relationships#one-to-many (uzupełnić w *Country.php*)

Zadanie 6.7:

Wykonać *seed* oraz sprawdzić zawartość bazy w *phpMyAdminie*

```
php artisan db:seed --class=CountrySeeder
php artisan db:seed --class=TripSeeder
```

Zadanie 6.8: *
Uzupełnić plik *database/seeders/DatabaseSeeder.php*, tak aby *seeder'y krajów* i *wycieczek* były wykonywane w odpowiedniej kolejności, przy wpisaniu jednej komendy.

```
php artisan db:seed
```

Zadania (*Blade, Eloquent ORM*):

Zadanie 6.9:
Uzupełnić *app/Http/Controllers/TripController.php* o dwie publiczne funkcje:
• funkcja `index`, zwracająca widok z pliku *resources/views/trips/index.blade.php* wraz z przekazaniem do niego zmiennej `trips` zawierającej kolekcję wszystkich wycieczek pobranych z bazy danych,
• funkcja `show`, przyjmująca jeden parametr *id*, zwracająca widok z pliku *resources/views/trips/show.blade.php* z przekazaniem do niego jednego obiektu wycieczki pobranego z bazy danych (wybór na podstawie wartości parametru `id`) lub status *404* w przypadku, gdy wycieczka nie istnieje.

https://laravel.com/docs/11.x/views
https://laravel.com/docs/11.x/eloquent#retrieving-models
https://laravel.com/docs/11.x/eloquent#not-found-exceptions

```
    public function index()
    {
        return /* … */;
    }

    public function show($id)
    {
        return /* … */;
    }
```

Zadanie 6.10:
Ustawić *trasy* dla obu funkcji *kontrolera* z poprzedniego zadania (ścieżka `/trips`).
Dla drugiej funkcji należy uwzględnić `id` (*parametr w ścieżce*). Nazwać tą *trasę* (*named route*, czyli „*trasa nazwana*") jako: `trips.show`.

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

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

Zadanie 6.11:
Korzystając z zawartości kolekcji `trips` (przekazywanej do widoku) wykonać z wykorzystaniem pętli `@forelse` … `@empty` … `@endforelse` dynamiczne generowanie *kart* i *tabelki* dotyczących wycieczek.
W linku „*Więcej szczegółów…*" ustalić `href` jako odniesienie do *trasy nazwanej* poprzednim zadaniu, wraz z użyciem *id* poszczególnej wycieczki.

```
{{route('trips.show', ['id' => $trip->id ])}}
```

Zadanie 6.12: *

Korzystając z obiektu wycieczki przekazywanego do widoku *show.blade.php*:
• umieścić na stronie kartę z danymi tej wycieczki,
• usunąć link „*Więcej szczegółów…*”,
• wyśrodkować kartę,
• wyśrodkować nagłówek „*Wycieczka*”,
• zmienić tytuł strony na „*Wycieczka …*” (nazwa danej wycieczki).

https://getbootstrap.com/docs/5.3/utilities/flex/#justify-content
https://getbootstrap.com/docs/5.3/utilities/text/#text-alignment

Przejść pod następujące adresy:
http://localhost:8000/trips/1
http://localhost:8000/trips/10

Zadanie 6.13: *

Sprawdzić działanie linków „*Więcej szczegółów…*” na kartach na stronie:

http://localhost:8000/trips

Zadanie 6.14: *

Zastąpić *trasy* dla obu funkcji *kontrolera* z zadania 6.10 jedną grupą tras dla tego kontrolera. Nie pominąć nazwy „*trasy nazwanej*” `trips.show`.

https://laravel.com/docs/11.x/routing#route-group-controllers

Zadanie 6.15: *

Zaproponować rozwiązanie problemu nadmiarowej ilości kart na stronie, np.:
• wyświetlać 4 najtańsze wycieczki,
• wyświetlać 4 losowe wycieczki,
• zmienić układ wyświetlania kart.

–

Zadanie 6.16: *

Wykorzystać skrypt `archiwizacja.bat` do spakowania projektu oraz innych plików wykluczonych w *.gitignore* (głównie z pominięciem folderu *vendor*).
Następnie, po wypakowaniu tego archiwum można analogicznie użyć *start.bat*.

–

Zadanie 6.17: *

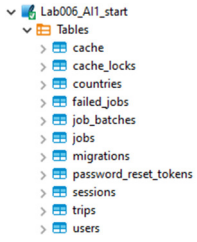
Wypróbować zastosowanie *SQLite* zamiast *MySQL*:
• w *.env* zmienić `DB_CONNECTION` na `sqlite`,
• wykonać *migracje* i *seeder'y*,
• sprawdzić zawartość nowo utworzonej bazy w pliku `ai1_lab6` (w katalogu głównym projektu) poprzez np. *DBeaver'a*.

https://laravel.com/docs/11.x/database#sqlite-configuration

```
php artisan migrate:fresh --seed
```

https://dbeaver.io/download/

Database -> *New Database Connection* -> *SQLite* -> ścieżka do pliku -> *Finish*



* – 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: v1.01