



Politechnika  
Śląska

Przedmiot: Podstawy informatyki

Nr. Ćwiczenia: 4

Imię i Nazwisko: Michał Witczak

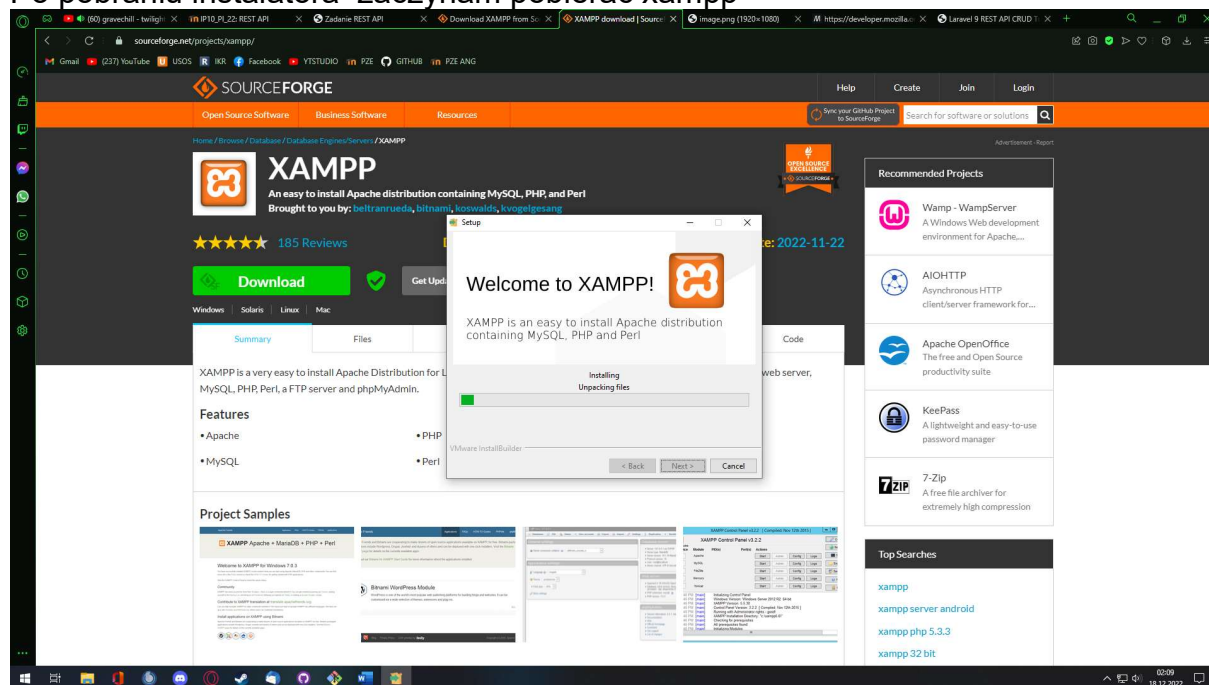
Nr. Albumu 305423

Link do repozytorium: <https://github.com/MichałWitczakmw/RESTAPI.git>

## 1. Instalacja xampp

Obeszło się bez problemów.

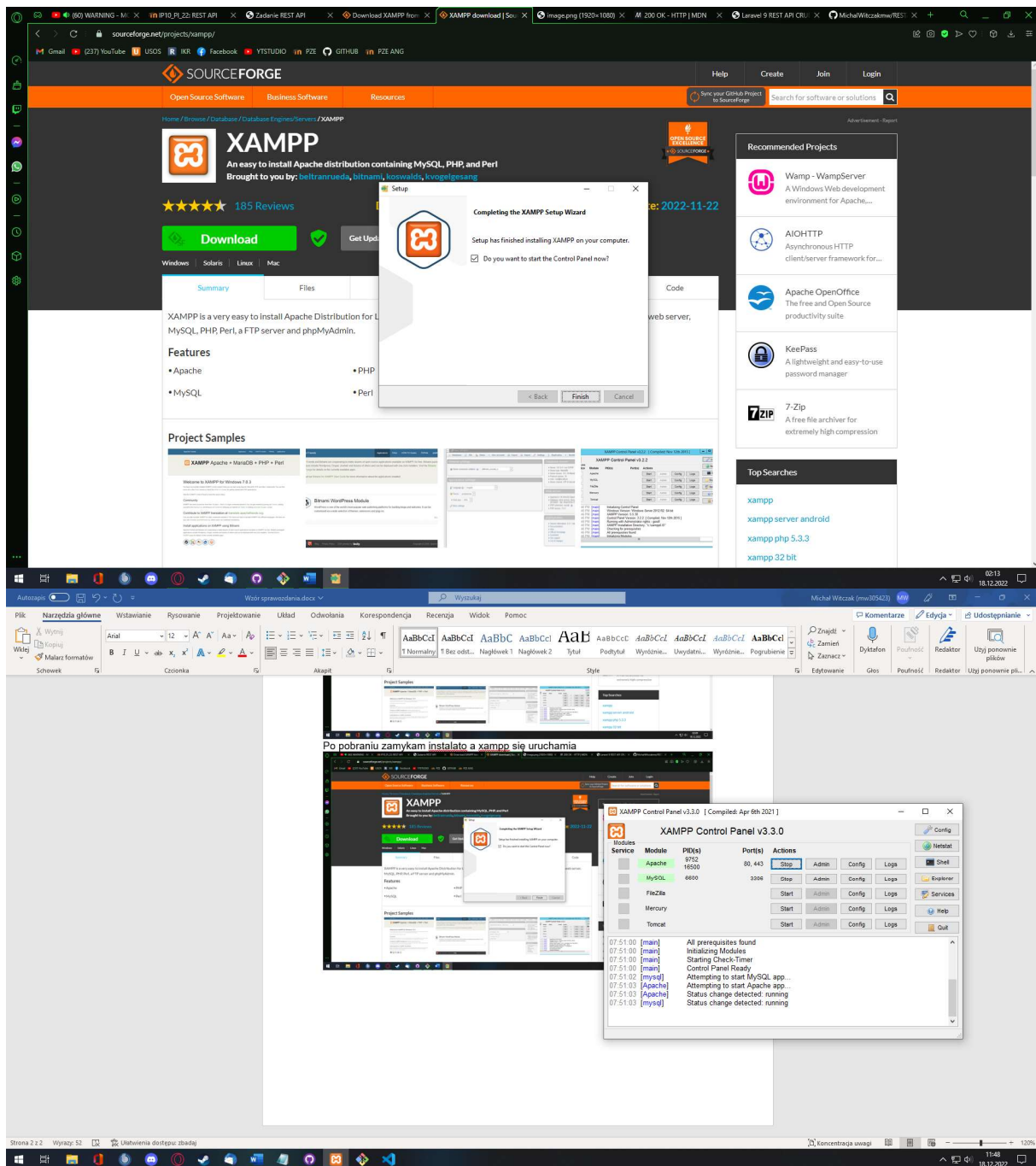
Po pobraniu instalatora zaczynam pobierać xampp



Po pobraniu zamykam instalato a  
xampp się uruchamia



Politechnika  
Śląska



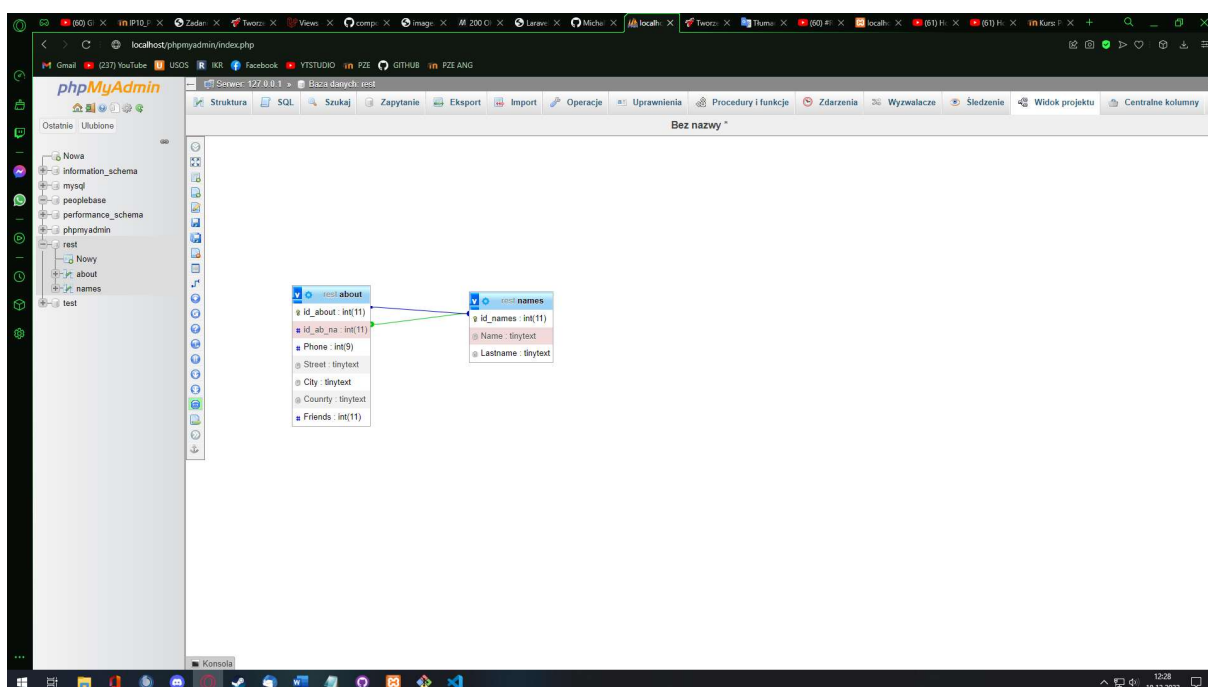
Uruchomiłem od razu serwer Apache i MySQL wbudowane w xampp.

2. Wykorzystać należy relacyjną bazę danych.

Tu stworzyłem bazę danych za pomocą MySQL poprzez wchodząc na za pomocą przycisku admin w xampp. I stworzyłem bazę danych z dwoma tabelami O nazwie baza: rest table: names, about. Potem połączyłem je kluczem obcym czyli zrobiłem „relację”.



Politechnika  
Śląska



Obyło się bez większych komplikacji. Jedynie to użyłem strony. Z taką komendą na jej podstawie stworzyłem tabele. A Klucz obcy zrobiłem w sposób okienkowy na MySQL.

[webmaster.helion.pl/index.php/kursmysql-tworzenie-modyfikowanie-i-usuwanie-tabel](http://webmaster.helion.pl/index.php/kursmysql-tworzenie-modyfikowanie-i-usuwanie-tabel)

IKR Facebook YTSTUDIO PZE GITHUB PZE ANG

Wykonanie instrukcji spowoduje dodanie nowej tabeli do b:

W pierwszej kolejności należy utworzyć tabele słownikowe

**Listing 10.2.** Tworzymy pierwszą tabelę bazy firma. Tabeli:

```
CREATE TABLE seria_wydawnicza
(id_serii int AUTO_INCREMENT PRIMARY KEY,
seria varchar (50) NOT NULL,
uwagi varchar (200))
```

3. Zainstalować program composer.



**Wzrost i zdrowie dziecka.docx**

Wyszukiwanie

Plik Narzędzia główne Wstawianie Rysowanie Projektowanie Układ Odwołania Korrespondencja Recenzja Widok Pomoc

Arial 12 I B U Z K x<sub>1</sub> x<sub>2</sub> Aa - Ap Włkn Malarz formatów Schowek Czcionka Akapit

Styl

T Normalny T Bez odst. Nagłówek 1 Nagłówek 2 Tytuł Podtytuł Wyświeżone... Wyświeżone... Wyświeżone... Poprawienie

Znajdź Zamień Zastąp Edytowanie Głos Północni Redaktor Użyj ponownie pliku

wykonanie instrukcji spowoduje utworzenie nowego tabeli w b.d.

W pierwszej kolejności należy utworzyć tabele słownikowe

**Listing 10.2. Tworzymy pierwszą tabelę bazy firm. Tabeli**

```
CREATE TABLE seria_wydawnicza
(id_serii int AUTO_INCREMENT PRIMARY KEY,
seria varchar(50) NOT NULL,
uwagi varchar(200))
```

3. Zainstalować program **composer**.

4. Utworzyć nowe repozytorium na github.com znajdując się na głównym **branchu main/master** znajdując się w **Merge Requestach**.

5. Utworzyć nowy projekt oparty na **frameworku laravel**.

6. Poprawnie skonfigurować projekt korzystając tutaj m.in. poprawnie skonfigurować bazę danych w projekcie wykorzystując pliki konfiguracyjne **.env** instrukcje do starszej wersji **Laravela**.

7. Utworzyć kontroler wraz z metodami będzie obsługiwać wszystkie przychodzące zapytania.

8. Dodać potrzebne wpisy obsługujące plik **routes/api.php**.

9. Przygotować odpowiednio migracje instancji obiektu People. Do zadania można wykorzystać bibliotekę.. Obiekt klasy People ma posiadać oprócz obowiązkowych właściwości (np. id) minimum: imię, nazwisko, numer telefonu, ulica, miasto/kraj. Można dodać również inne swoje.

10. Kontroler ma posiadać metody które umożliwią obsługę następujących funkcjonalności (wszystkie metody CRUD – **create**, **read**, **update**, **delete** + pokaż wszystkich)

Composer Setup

Information  
Please read the following information before continuing.

Important  
You must open a new command window to use Composer for the first time, because your environment has changed and running programs may not be aware of this.  
  
If this does not work, you will have to do one of the following:  
- Close all File Explorer windows, then open a new command window. OR  
- Logout and Login again, then open a new command window.  
  
As a last resort, you may need to restart your computer.

Next

Strona 4 z 5 Wyrzucić 35% Uaktualnienia dostępu: zbłądził

---

**Wzrost i zdrowie dziecka.docx**

Wyszukiwanie

Plik Narzędzia główne Wstawianie Rysowanie Projektowanie Układ Odwołania Korrespondencja Recenzja Widok Pomoc

Arial 12 I B U Z K x<sub>1</sub> x<sub>2</sub> Aa - Ap Włkn Malarz formatów Schowek Czcionka Akapit

Styl

T Normalny T Bez odst. Nagłówek 1 Nagłówek 2 Tytuł Podtytuł Wyświeżone... Wyświeżone... Wyświeżone... Poprawienie

Znajdź Zamień Zastęp Edytowanie Głos Północni Redaktor Użyj ponownie pliku

(id\_serii int AUTO\_INCREMENT PRIMARY KEY,  
seria varchar(50) NOT NULL,  
uwagi varchar(200))

3. Zainstalować program **composer**.

4. Utworzyć nowe repozytorium na github.com znajdując się na głównym **branchu main/master**. Obecnie znajdując się w **Merge Requestach**.

5. Utworzyć nowy projekt oparty na **frameworku laravel**.

6. Poprawnie skonfigurować projekt korzystając tutaj m.in. poprawnie skonfigurować bazę danych w projekcie - tak aby połączenie działało i dobrze wykorzystywać pliki konfiguracyjne **.env**. Tutaj jako pomoc można skorzystać instrukcje do starszej wersji **Laravela**.

Completing Composer Setup

Setup has installed Composer on your computer.  
Usage: Open a command window and type "composer".

Click Finish to exit Setup.

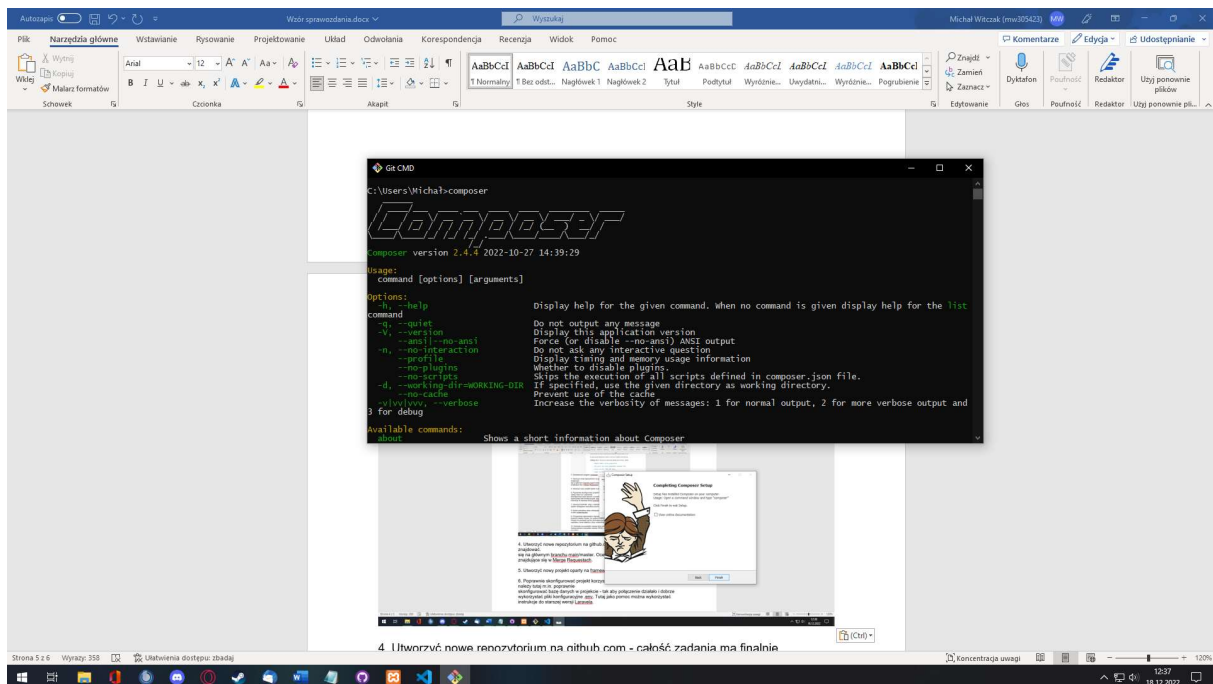
☐ View online documentation

Back Finish

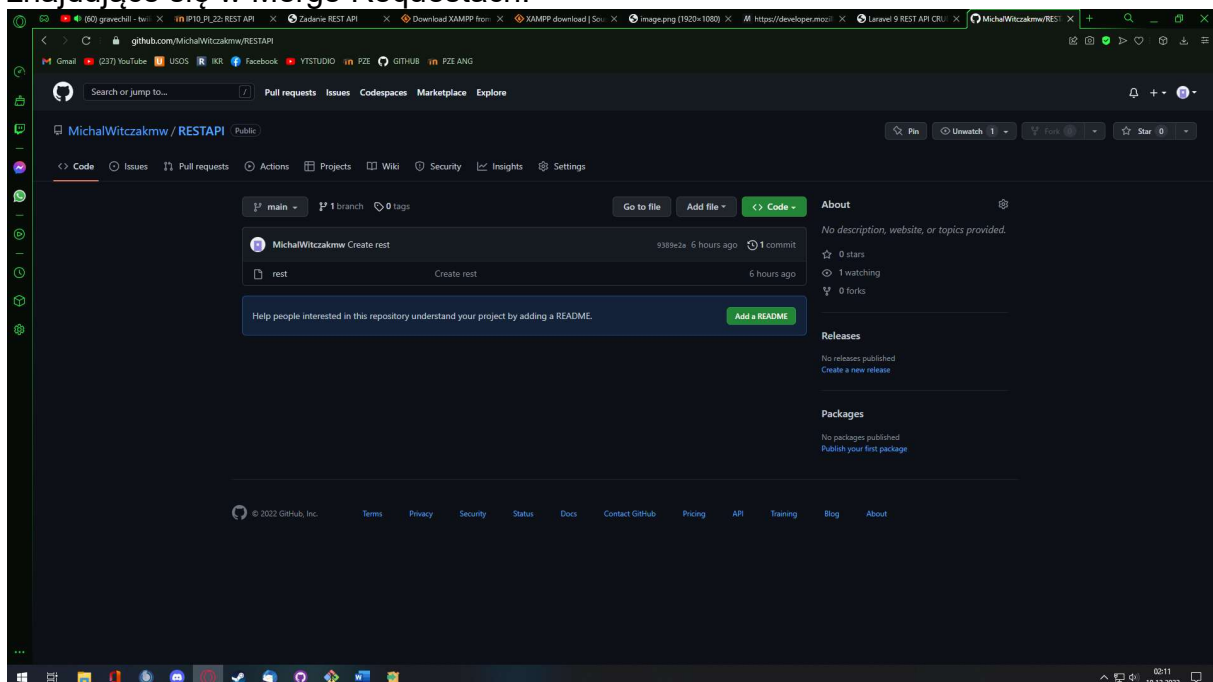
Strona 4 z 5 Wyrzucić 35% Uaktualnienia dostępu: zbłądził



Politechnika  
Śląska



4. Utworzyć nowe repozytorium na github.com - całość zadania ma finalnie znajdować się na głównym branchu main/master. Ocenie nie podlegają inne branche/zadania znajdujące się w Merge Requestach.



5. Utworzyć nowy projekt oparty na frameworku Laravel.





```
Git CMD
C:\Users\Michał\example-app>cd..
C:\Users\Michał>composer create-project laravel/laravel RestApi
Creating a "laravel/laravel" project at "./RestApi"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v9.4.0)
- Installing laravel/laravel (v9.4.0): Extracting archive
  Failed to extract laravel/laravel: (7) "C:\Program Files\7-Zip\7z.EXE" x -bb0 -y C:\Users\Michał\vendor\composer\tmp
  -e28898953a50b9d4d77e7e76e4972efb -oC:\Users\Michał\vendor\composer\30acec61

The archive may contain identical file names with different capitalization (which fails on case insensitive filesystems)
Unzip with 7z command failed, falling back to ZipArchive class
Created project in C:\Users\Michał\RestApi
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 107 installs, 0 updates, 0 removals
- Locking brick/math (0.10.2)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/instantiator (1.4.1)
- Locking doctrine/lexer (1.2.3)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (3.2.1)
- Locking fakerphp/faker (v1.21.0)
- Locking filp/whoops (2.14.6)
- Locking fruitcake/php-cors (v1.2.0)
```

```
Git CMD
Generating optimized autoload files
> illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

INFO Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

81 packages you are using are looking for funding.
Use the composer fund command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

INFO No publishable resources for tag [laravel-assets].

No security vulnerability advisories found
> @php artisan key:generate --ansi

INFO Application key set successfully.

C:\Users\Michał>
```

```
Git CMD - php artisan serve

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

81 packages you are using are looking for funding.
Use the composer fund command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

INFO No publishable resources for tag [laravel-assets].

No security vulnerability advisories found
> @php artisan key:generate --ansi

INFO Application key set successfully.

C:\Users\Michał> cd RestApi
C:\Users\Michał\RestApi>php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```



```
Git CMD - php C:\Users\Michał\RestApi\artisan serve
operable program or batch file.
C:\Users\Michał\RestApi>mkdir witzczak
C:\Users\Michał\RestApi>cd witzczak
C:\Users\Michał\RestApi\witzczak>mkdir 305423
C:\Users\Michał\RestApi\witzczak>cd 305423
C:\Users\Michał\RestApi\witzczak\305423>mkdir people
C:\Users\Michał\RestApi\witzczak\305423>cd people
C:\Users\Michał\RestApi\witzczak\305423\people>php artisan serve
Could not open input file: artisan
C:\Users\Michał\RestApi\witzczak\305423\people>php C:\Users\Michał\RestApi\artisan serve
Could not open input file: C:\Users\Michał\RestApi\artisan serve
C:\Users\Michał\RestApi\witzczak\305423\people>php C:\Users\Michał\RestApi\artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

6. Poprawnie skonfigurować projekt korzystając z dokumentacji:  
należy tutaj m.in. poprawnie  
skonfigurować bazę danych w projekcie - tak aby połączenie działało i dobrze  
wykorzystać pliki konfiguracyjne .env. Tutaj jako pomoc można wykorzystać  
instrukcje do starszej wersji Laravela.

```
File Edit Selection View Go Run ... Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
index.php .env .env.example
C:\Users\Michał\RestApi> .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:mpy5X1wetcm/uHnYFfNqATYDL/xBtRnG3uUN1xxd24c=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=rest
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailhog
33 MAIL_PORT=1025
The Marketplace has extensions that can help with '.example' files
Search Marketplace Don't Show Again for '.example' files
Ln 8, Col 30 Spaces: 4 UTF-8 LF Properties
```



7. Utworzyć kontroler wraz z metodami, który będzie obsługiwać wszystkie przychodzące requesty. Zrobiłem tyle co znalazłem i zrozumiałem w internecie tyle zrobiłem. Mam problem z poszczególnymi linijkami kodu. Same tworzenie plików nie było problemem. Nie zdążę już nic zrobić/ poprawić. Niestety muszę się zająć innymi przedmiotami.

```
peoplecontroller.php 7 x
C: > xampp > htdocs > RestApi > app > Http > Controllers > peoplecontroller.php > ...

9      /**
10     * Display a listing of the resource.
11     *
12     * @return \Illuminate\Http\Response
13     */
14     public function index()
15     {
16         //
17     }
18
19     /**
20     * Show the form for creating a new resource.
21     *
22     * @return \Illuminate\Http\Response
23     */
24     public function create()
25     {
26         //
27     }
28
29     /**
30     * Display the specified resource.
31     *
32     * @param int $id
33     * @return \Illuminate\Http\Response
34     */
35     public function show($id)
36     {
37         //
38     }
39
40     /**
41     * Update the specified resource in storage.
42     *
43     * @param \Illuminate\Http\Request $request
44     * @param int $id
45     * @return \Illuminate\Http\Response
46     */
47     public function update(Request $request, $id)
48     {
49         //
50     }
51
52     /**
53     * Remove the specified resource from storage.
54     *
55     * @param int $id
56     * @return \Illuminate\Http\Response
57     */
58     public function destroy($id)
```





```
WybierzGit CMD

INFO Migration [C:\xampp\htdocs\RestApi\database\Migrations\2022_12_18_135313_create_people_table.php] created successfully.

C:\xampp\htdocs\RestApi>php artisan make:request StorePostRequest

INFO Request [C:\xampp\htdocs\RestApi\app\Http\Requests\StorePostRequest.php] created successfully.

C:\xampp\htdocs\RestApi>php artisan make:resource PostResource

INFO Resource [C:\xampp\htdocs\RestApi\app\Http\Resources\PostResource.php] created successfully.

C:\xampp\htdocs\RestApi>create_posts_table.php
'create_posts_table.php' is not recognized as an internal or external command,
operable program or batch file.

C:\xampp\htdocs\RestApi>php artisan make:controller Api/PostController --model=Post

A App\Models\Post model does not exist. Do you want to generate it? (yes/no) [yes]
yes

INFO Model [C:\xampp\htdocs\RestApi\app\Models\Post.php] created successfully.

INFO Controller [C:\xampp\htdocs\RestApi\app\Http\Controllers\Api\PostController.php] created successfully.

C:\xampp\htdocs\RestApi>
```

```
File Edit Selection View Go Run ... RestApi

EXPLORER 2022_12_18_135313_create_people_table.php X DatabaseSeeder.php
RESTAPI database > migrations > 2022_12_18_135313_create_people_table.php > PHP > class@anonymous[]file:///c:/xampp/htdocs/RestApi
  > Resources
  > Kernel.php
  > Models
  > Providers
  > bootstrap
  > config
  > database
    > factories
    > migrations
      2014_10_12_000000...
      2014_10_12_100000...
      2019_08_19_000000...
      2019_12_14_000000...
      2022_12_18_135313...
    > seeders
  > .gitignore
  > lang
  > public
  > resources
  > routes
  > storage
  > tests
  > vendor
  > Witczak
  > .editorconfig
  > .env
  > .env.example
  > .gitattributes
  > OUTLINE
  > TIMELINE

8 {
9
10 /**
11  * Run the migrations.
12  *
13  * @return void
14  */
15 public function up()
16 {
17     Schema::create('people', function (Blueprint $table) {
18         $table->id();
19         $table->string('name');
20         $table->string('Lastname');
21         $table->integer('Phone');
22         $table->string('Street');
23         $table->string('City');
24         $table->string('Country');
25         $table->integer('Friends');
26         $table->timestamps();
27     });
28
29 /**
30  * Reverse the migrations.
31  *
32  * @return void
33  */
34 public function down()
35 {
36     Schema::dropIfExists('people');
37 }
38 }
39
```

Ln 26, Col 12 Spaces: 4 UTF-8 LF PHP 8.0



```
Wiersz polecenia
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Michał>cd C:\xampp\htdocs\RestApi^B
Nazwa pliku, nazwa katalogu lub składnia etykiety woluminu jest niepoprawna.

C:\Users\Michał>cd C:\xampp\htdocs\RestApi

C:\xampp\htdocs\RestApi>php php artisan make:seeder PeopSeeder
Could not open input file: php

C:\xampp\htdocs\RestApi>php artisan make:seeder PeopSeeder

INFO Seeder [C:\xampp\htdocs\RestApi\database\seeders\PeopSeeder.php] created successfully.

C:\xampp\htdocs\RestApi>

WybierzWiersz polecenia
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Michał>cd C:\xampp\htdocs\RestApi

C:\xampp\htdocs\RestApi>php artisan make:request requestpeople

INFO Request [C:\xampp\htdocs\RestApi\app\Http\Requests\Requestpeople.php] created successfully.

C:\xampp\htdocs\RestApi>php artisan make:resource peopresource

INFO Resource [C:\xampp\htdocs\RestApi\app\Http/Resources/peopresource.php] created successfully.

C:\xampp\htdocs\RestApi>php artisan make:controllerk peopcontroller --model=people

ERROR Command "make:controllerk" is not defined. Did you mean one of these?

make:cast
make:channel
make:command
make:component
make:controller
make:event
make:exception
make:factory
make:job
make:listener
make:mail
```



```
Wiersz polecenia

make:notification
make:observer
make:policy
make:provider
make:request
make:resource
make:rule
make:scope
make:seeder
make:test

C:\xampp\htdocs\RestApi>php artisan make:controller peopcontroller --model-people

The "--model-people" option does not exist.

C:\xampp\htdocs\RestApi>php artisan make:controller peopcontroller --model -people

A App\Models\eople model does not exist. Do you want to generate it? (yes/no) [yes]
yes

INFO: Model [C:\xampp\htdocs\RestApi\app\Models\eople.php] created successfully.
INFO: Controller [C:\xampp\htdocs\RestApi\app\Http\Controllers\peopcontroller.php] created successfully.

C:\xampp\htdocs\RestApi>
```

The screenshot shows the VS Code editor with the file `peopcontroller.php` open. The code is as follows:

```
1 <?php
2 namespace App\Http\Controllers;
3 use Illuminate\Http\Request;
4 class peopcontroller.php extends Controller
5 {
6     /**
7      * Display a listing of the resource.
8      *
9      * @return \Illuminate\Http\Response
10     */
11     public function index()
12     {
13         $people=people::all();
14         return response([peopresource::collection($people),200]);
15     }
16     /**
17      * Show the form for creating a new resource.
18      *
19      * @return \Illuminate\Http\Response
20     */
21     public function create()
22     {
23         //
24     }
25     /**
26      * Display the specified resource.
27      *
28      * @param int $id
29      * @return \Illuminate\Http\Response
30     */
31     public function show($id)
32     {
33         //
34     }
35     /**
36      * Update the specified resource in storage.
37      *
38      * @param int $id
39      * @param string $update
40      * @return \Illuminate\Http\Response
41     */
42     public function update($id, $update)
43     {
44         //
45     }
46 }
```

The Problems panel shows the following errors:

- Syntax error: unexpected token ']' PHP(PHP2014) [Ln 4, Col 23]
- Unexpected ']' Expected '[' intelephense(1001) [Ln 4, Col 23]
- Unexpected 'public:' intelephense(1001) [Ln 11, Col 5]
- Unexpected 'public:' intelephense(1001) [Ln 21, Col 5]
- Unexpected 'public:' intelephense(1001) [Ln 31, Col 5]
- Unexpected 'public:' intelephense(1001) [Ln 42, Col 5]
- Unexpected 'public:' intelephense(1001) [Ln 52, Col 5]



```
<?php
namespace Database\Seeders;

use Illuminate\Support\Str;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class PeopSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        for($i = 0; $i < 50; $i++)
        {
            DB::table('people')->insert([
                'Name' => Str::random(10),
                'Surname' => Str::random(10),
                'Phone' => rand(100000000, 999999999),
                'Street' => Str::random(7),
                'City' => Str::random(10),
                'Country' => Str::random(10),
                'Friends' => rand(1, 99)
            ]);
        }
    }
}
```

8. Dodać potrzebne wpisy obsługujące routing przychodzących requestów w pliku routes/api.php

udało mi się tylko zrobić jeden request na więcej nie mam czasu.

```
<?php
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class requestpeople extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return false;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array(string, mixed)
     */
    public function rules()
    {
        return [
            //
        ];
    }
}
```

9. Przygotować odpowiednio migrację zawierającą utworzenie 50fake-owych instancji obiektu People. Do zadania można wykorzystać bibliotekę:. Obiekt klasy People ma posiadać oprócz obowiązkowych właściwości (np. id) minimum: imię, nazwisko, numer telefonu, ulica, miasto/kraj. Można dodać również inne swoje.



Zrobiłem sam plik speeder nic dalej ponieważ brak czasu.

```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Support\Str;
6 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
7 use Illuminate\Database\Seeder;
8 use Illuminate\Support\Facades\DB;
9
10 class PeopSeeder extends Seeder
11 {
12     /**
13      * Run the database seeds.
14      *
15      * @return void
16      */
17     public function run()
18     {
19         for($i = 0; $i < 100; $i++)
20         {
21             DB::table('people')->insert([
22                 'name' => Str::random(10),
23                 'lastname' => Str::random(10),
24                 'street' => Str::random(7),
25                 'city' => Str::random(10),
26                 'country' => Str::random(10),
27                 'friends' => rand(1, 99)
28             ]);
29         }
30     }
31 }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

10. Kontroler ma posiadać metody które umożliwią obsługę następujących funkcjonalności (wszystkie metody CRUD - create, read, update, delete + pokaż wszystko)

11. Pokazać działanie aplikacji poprzez wykorzystanie programu Postman każda operacja ma posiadać swój własny screen z requestem postmana oraz otrzymaną odpowiedź.

12. Należy w programie szczególnie uwzględnić obsługę metod HTTP, które wykorzystywane są w operacjach CRUD. Inne metody nie są obowiązkowe.

13. Należy w kontrolerze oraz w postmanie szczególnie zwrócić uwagę na kody zwracanych requestów.

14. Adresy URL (endpointy) mają zawierać nazwisko oraz numer studenta w swojej pracy, np. 127.0.0.1/kowalski/12345/people/5

Co to jest migracja?

Migration tool to narzędzie w Laravelu, które pozwoli Ci stworzyć nową tabelę w bazie bazując na definicji w kodzie PHP. Wewnątrz metody PHP wskazujesz, jakie kolumny ma zawierać tabela, jakiego typu i jakiej nazwy, po czym narzędzie do migracji stworzy je w bazie danych za Ciebie.

Co to są kody HTTP?

Kod HTTP to informacja zwrócona przez serwer do użytkownika. Każdy posiada odrębne oznaczenie numeryczne. Niektóre z nich są czysto informacyjne, inne potwierdzają powodzenie, lub przekierowanie. Wyróżniamy również błędy HTTP, które strony internetowej bądź serwera.

Co to jest programowanie obiektowe?

Programowanie obiektowe to programowania, w którym programy definiuje się za pomocą obiektów – elementów łączących stan i zachowanie.





Politechnika  
Śląska

Dlaczego wykorzystuje się pliki .env ?

Do gromadzenia danych o środowisków użytkownika

Czym charakteryzuje się relacyjna baza danych?

-mniej ważyć tabelą

-ma co najmniej jedną relację pomiędzy co najmniej dwoma tabelami

-łatwiej znaleźć daną kolumnę do zmiany dla danego id

-zapobiega powtarzaniu się informacji w tabelach

Np.: kupiłeś 3 bułki to w tabeli pisze twoje imię a w drugiej ile sztuk a w trzeciej czego

Wykorzystanie plików w formacie json?

JSON może być używany w programach bez potrzeby analizy składniowej i serializacji. Jest to tekstowy sposób reprezentowania literałów, tablic i danych skalarnych obiektów