Sprawozdanie

Rodzaj zajęć i nazwa przedmiotu
Podstawy Informatyki

Rok akademicki	Miasto	Tryb	Kierunek	Semestr	Prowadzący	Grupa
2023/24	K	S	IP	1	DJ	4

Termin oddania ćwiczenia	
Data	Godzina
15.01.2024	15:30–17:45

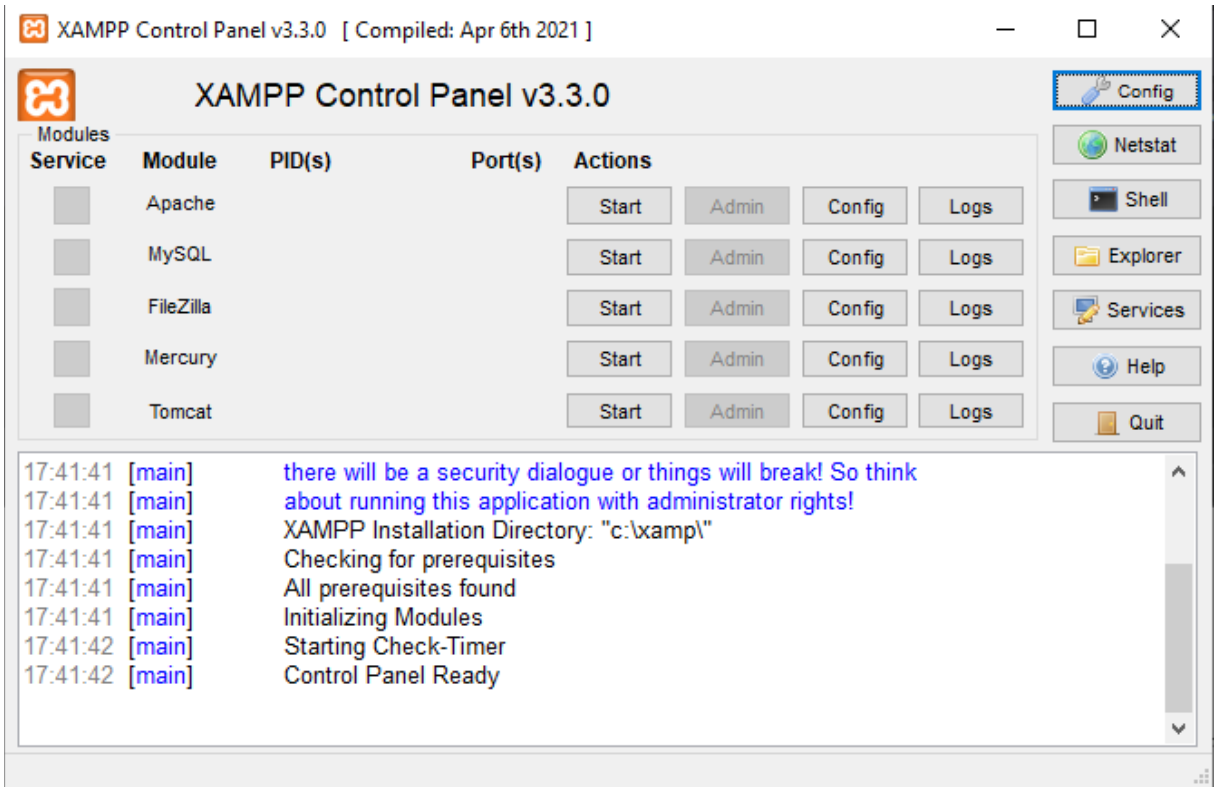
Temat ćwiczenia
Zadanie REST API oparte na frameworku Laravel (PHP)

Numer albumu	Imię i nazwisko
310794	Damian Sobczyk

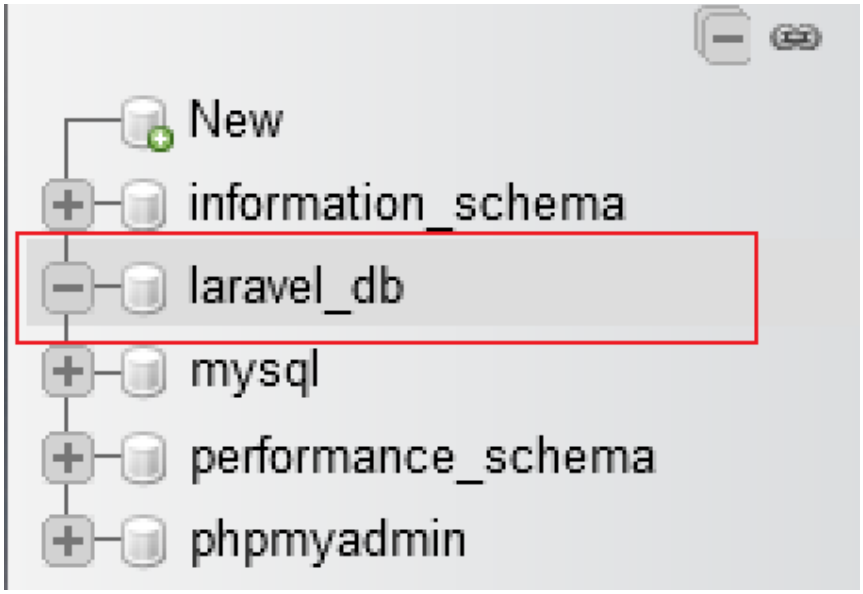
Politechnika Śląska

https://github.com/DSobczyk2003/REST_API_LARAVEL_IPPP.git

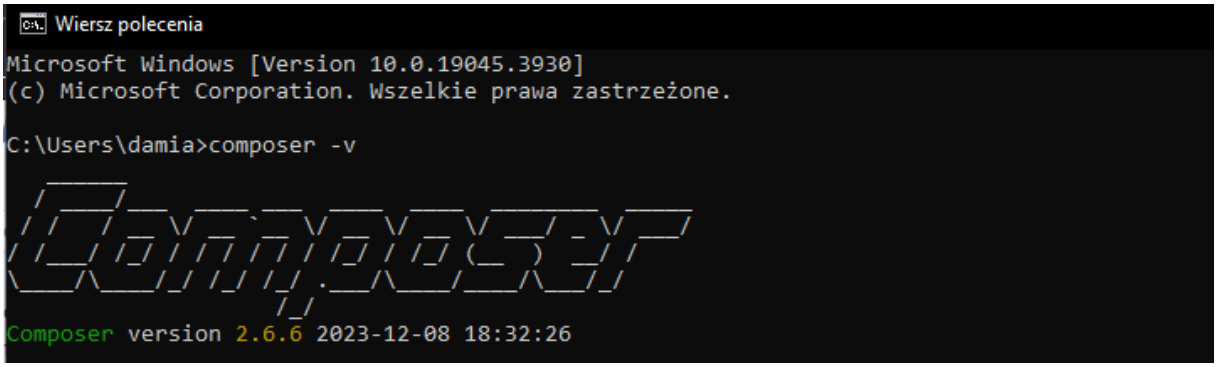
Instalacja pakietu XAMPP:



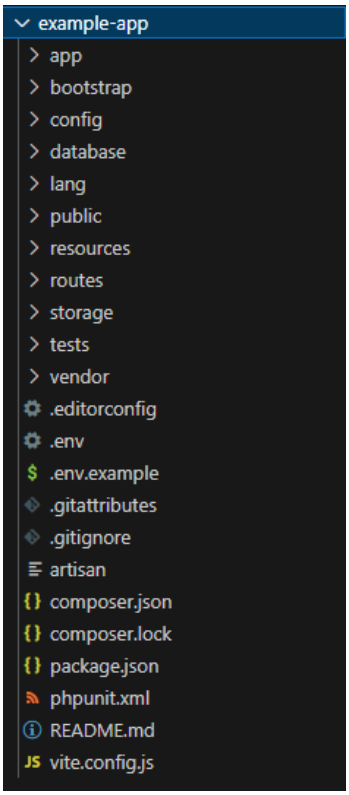
Stworzenie bazy danych w PhpMyAdmin:



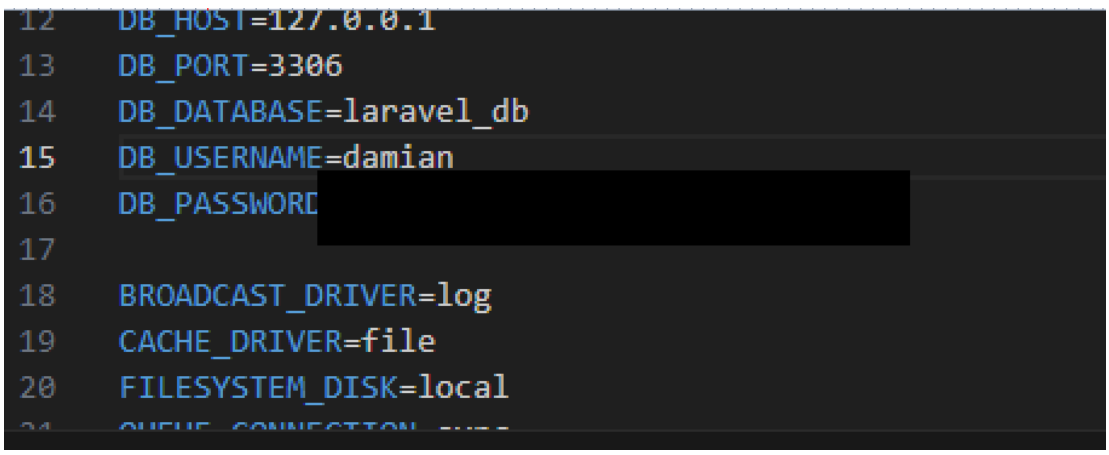
Instalacja Composera:



Pliki powstałe w wyniku instalacji:



Konfiguracja pliku .env:



Konfiguracja metod w pliku kontrolera:

Wyświetlanie wszystkich rekordów z bazy:

```php
public function index()
{
    $users = User::all();
    if (!$users) {
        return response()->json(['message' => 'No records found'], 404);
    }
    return response()->json($users);
}
```

Zapisywanie dodatkowych danych wybranych przez użytkownika:

```php
public function store(Request $request)
{
    $user = User::create($request->all());
    return response()->json($user, 201);
}
```

Metoda wyświetla wybrany rekord z tabeli: Filtracja odbywa się po ID

```php
public function show(int $id)
{
    $user = User::find($id);
    if (!$user) {
        return response()->json(['message' => 'Record does not exist'], 404);
    }
    return response()->json($user);
}
```

Metoda aktualizująca rekord:

```php
public function update(Request $request, int $id)
{
    $user = User::find($id);
    if (!$user) {
        return response()->json(['message' => 'Record does not exist'], 404);
    }

    $user = User::find($id);

    if ($request->has('name')) {
        $user->name = $request->name;
    }
    if ($request->has('surname')) {
        $user->surname = $request->surname;
    }
    if ($request->has('phone_number')) {
        $user->phone_number = $request->phone_number;
    }
    if ($request->has('street')) {
        $user->street = $request->street;
    }
    if ($request->has('country')) {
        $user->country = $request->country;
    }
    $user->save();
    return response()->json($user);
}
```

Metoda kasująca:

```php
    public function destroy(int $id)
    {
        $user = User::find($id);
        if (!$user) {
            return response()->json(['message' => 'Record does not exist'], 404);
        }
        $user->delete();
        return response()->json(["message" => "User with id[$id] deleted"], 200);
    }
}
```

Konfiguracja FAKERA, który uzupełnia tabelę losowymi danymi.

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Faker\Factory as Faker;

class PeopleTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $faker = Faker::create();

        for ($i = 0; $i < 200; $i++) {
            \DB::table('people')->insert([
                'name' => $faker->firstName,
                'surname' => $faker->lastName,
                'phone_number' => $faker->phoneNumber,
                'street' => $faker->streetAddress,
                'country' => $faker->country,
            ]);
        }
    }
}
```

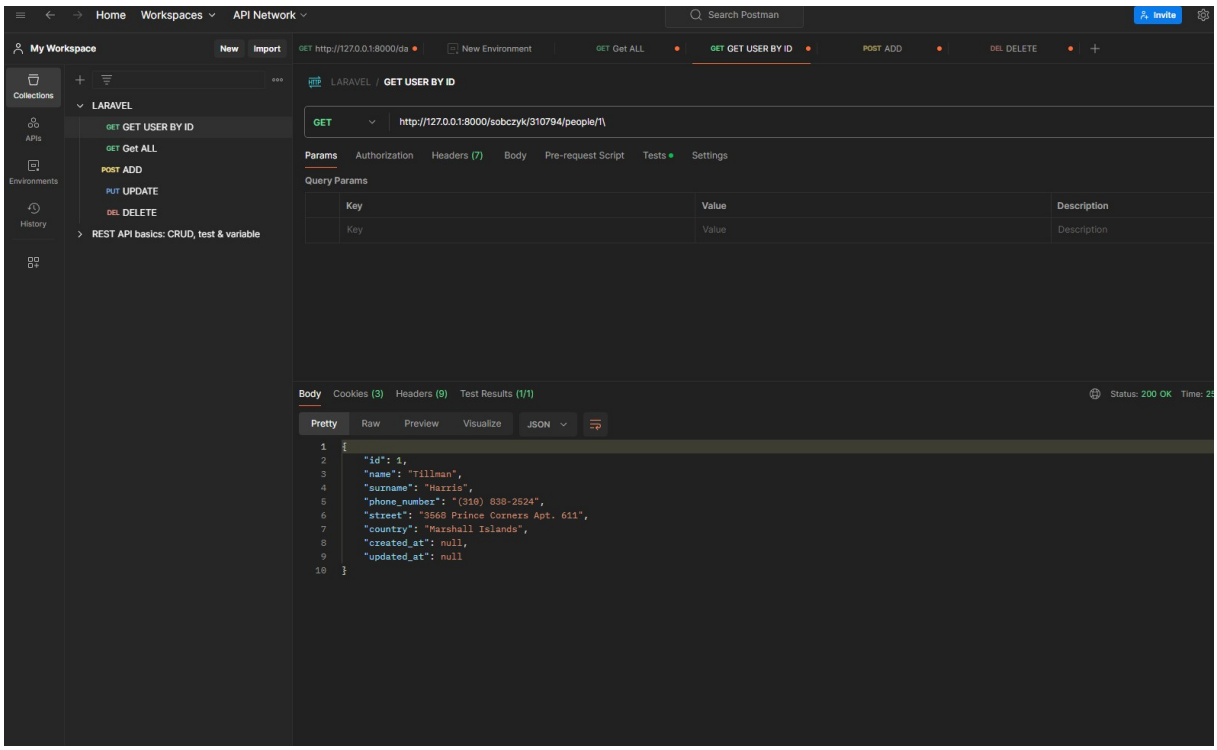
Terminal:

```
PS C:\XAMP\htdocs\damian-app\example-app> php artisan db:seed

  INFO  Seeding database.

  Database\Seeders\PeopleTableSeeder ..... RUNNING
  Database\Seeders\PeopleTableSeeder ..... 569.17 ms DONE

PS C:\XAMP\htdocs\damian-app\example-app>
```

Tabela powstała w wyniku migracji:

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

require_once 'vendor/autoload.php';

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('people', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('surname');
            $table->string('phone_number');
            $table->string('street');
            $table->string('country');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('people');
    }
};

```

GET USER BY ID



Wszystkie rekordy: