**Web programozás**

**11.Laborgyakorlat**

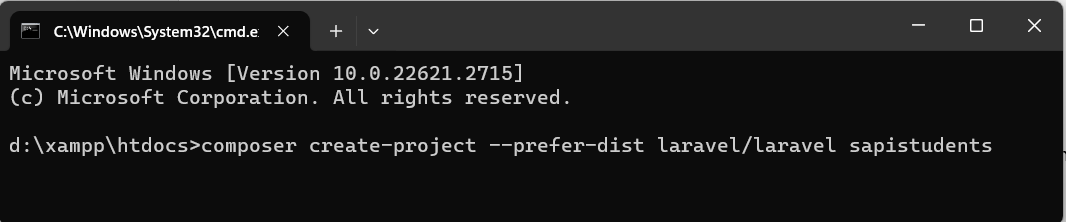
**Téma:**

* AJAX kérések és feldolgozás
* Laravel telepítés, környezet megismerése

**Feladat:**

1. Tanulmányozzuk a következő AJAX technikákat:
   1. httprequest, jquery, Fetch API, axios
2. Tanulmányozzuk az AJAX CRUD műveleteket
3. Laravel telepítés és környezet megismerése
   1. Telepítés, adatbázis beállítása
      1. Egy *sapistudents* nevű Laravel projekt telepítése
         1. Parancssorból (cmd vagy Git Bash) kell a parancsot kiadni (pld. d:\xampp\htdocs) a XAMPP htdocs mappáján

composer create-project --prefer-dist laravel/laravel sapistudents



* + 1. Projekt működésének ellenőrzése
       1. Böngésző címsorába beírni:
          1. <https://localhost/sapistudents/public>
    2. Adatbázis létrehozása (pld. sapistudents)
    3. Adatbázis beállítása az .env fájlban

DB\_DATABASE=sapistudents

* 1. Migrációs fájl, fake adatok, seed fájl létrehozása
     1. Student model, migrációs fájl létrehozása

php artisan make:model Student –mfs

* 1. Student séma elkészítése

$table->id();

$table->string('name');

$table->float('grade');

$table->timestamps();

* 1. Factory: fake adatok (StudentFactory.php)

'name' => $this->faker->name(),

'grade' => $this->faker->randomFloat(2, 1, 10),

* 1. Seeder fájl (StudentSeeder.php)

Student::factory(20)->create();

* 1. Seeder hívása (DatabaseSeeder.php)

$this->call(StudentSeeder::class);

* 1. Migráció végrehajtása

php artisan migrate –seed

1. Adatok egyszerű megjelenítése (get route a web.php-ben)
   1. Az összes Student lekérése

Route::get('students', function (){

$students = \App\Models\Student::all();

foreach ($students as $student){

echo $student->name . "<br>";

}

});

* 1. Egy adott (id alapján) Post lekérése

Route::get('students/{id}', function ($id){

echo \App\Models\Student::find($id);

});

1. Kontrollerek, view meghívás
   1. Kontroller létrehozás

php artisan make:controller StudentController

* 1. Kontroller metódus (pld. index) megírása

public function index()

{

$posts = Post::all();

return view('posts.index', compact('posts'));

}

* 1. Blade fájl létrehozása (resources\views\students\**index.blade.php**)

@foreach($students as $student)

<tr>

<td>{{$student->id}}</td>

<td>{{$student->name}}</td>

<td>{{$student->grade}}</td>

</tr>

@endforeach

* 1. Route megadása (routes\web.php)

Route::get('students', [StudentController::class, 'index'])->name('students.index');

1. Adat létrehozás (create)
2. Kontroller (create)

public function create()

{

return view('students.create');

}

1. View (students.create)

<form method="post" action="{{route('students.store')}}">

@csrf

<div class="">

<label for="name">Name:</label>

<input type="text" name="name">

</div>

<div class="">

<label for="grade">Grade:</label>

<input type="text" name="grade">

</div>

<button type="submit">Add student</button>

</form>

1. Kontroller (store)

public function store(Request $request)

{

$request->validate([

name' => 'required',

'grade' => 'required'

]);

Student::create($request->all());

return redirect()->route('students.index');

}

1. Routes

Route::get('students/create', [StudentController::class, 'create']);

Route::post('students/store', [StudentController::class, 'store'])->name('students.store');