

Nama : Salomo Polanco

NIM : M0521068

Kelas : Informatika C

1. Table t_fighter

```
home.blade.php  form.blade.php  edit-form.blade.php  2023_06_16_032903_create_t_fighter_table.php  edit-style.css  table-fighter.blade.php  edit-form-style.css  FighterController.php  web.php
database > migrations > 2023_06_16_032903_create_t_fighter_table.php > % class > up
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9
10     /**
11      * Run the migrations.
12      */
13     public function up(): void
14     {
15         Schema::create('t_fighter', function (Blueprint $table) {
16             $table->id();
17             $table->string('name',100);
18             $table->integer('age');
19             $table->string('basic',100);
20             $table->string('weightclass',100);
21             $table->string('fighter_photo',500);
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('t_fighter');
32     }
33 }
```

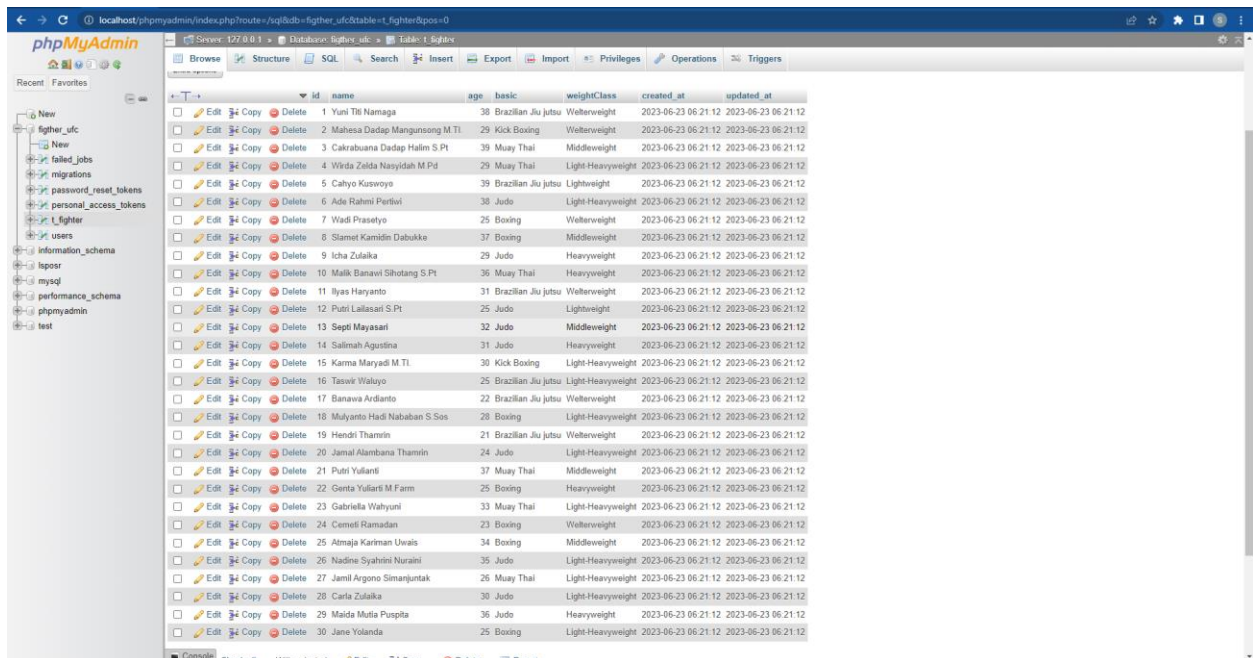
```
m.blade.php  table-fighter.blade.php  web.php  2023_06_16_032903_create_t_fighter_table.php  Fighter.php  FighterFactory.php  FighterController.php
database > factories > FighterFactory.php > FighterFactory
3  namespace Database\Factories;
4
5  use Illuminate\Database\Eloquent\Factories\Factory;
6
7  /**
8   * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Fighter>
9   */
10 class FighterFactory extends Factory
11 {
12     /**
13      * Define the model's default state.
14      *
15      * @return array<string, mixed>
16      */
17     public function definition(): array
18     {
19         return [
20             'name' => fake()->name(),
21             'age' => fake()->numberBetween(20,40),
22             'basic' => fake()->randomElement(["Brazilian Jiu jitsu", "Muay Thai", "Boxing", "Kick Boxing", "Judo"]),
23             'weightClass' => fake()->randomElement(["Lightweight", "Welterweight", "Middleweight", "Light-Heavyweight", "Heavyweight"]);
24         ];
25     }
26 }
27
```

```
rm.blade.php  table-fighter.blade.php  web.php  2023_06_16_032903_create_t_fighter_table.php  Fighter.php  FighterFactory.php  FighterSeeder.php  config

database > seeders > FighterSeeder.php > FighterSeeder > run
1  <?php
2
3  namespace Database\Seeders;
4
5  use App\Models\Fighter;
6  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
7  use Illuminate\Database\Seeder;
8
9  class FighterSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         Fighter::factory()
17             ->count(30)
18             ->create();
19     }
20 }
21
```

```
FighterSeeder.php  DatabaseSeeder.php  app.php  config > app.php
96 |
97 | /*
98 |
99 | 'fallback_locale' => 'en',
100 |
101 | /*
102 | -----
103 | | Faker Locale
104 | | -----
105 | |
106 | | This locale will be used by the Faker PHP library when generating fake
107 | | data for your database seeds. For example, this will be used to get
108 | | localized telephone numbers, street address information and more.
109 | |
110 | | /*
111 | |
112 | | 'faker_locale' => 'id_ID',
113 | |
114 | | /*
115 | | -----
116 | | | Encryption Key
117 | | | -----
118 | | |
119 | | | This key is used by the Illuminate encrypter service and should be set
120 | | | to a random, 32 character string, otherwise these encrypted strings
121 | | | will not be safe. Please do this before deploying an application!
122 | | |
123 | | | /*
124 | | |
125 | | | 'key' => env('APP_KEY'),
126 | | |
127 | | | 'cipher' => 'AES-256-CBC',
128 | | |
129 | | | /*
```

```
FighterSeeder.php DatabaseSeeder.php x
database > seeders > DatabaseSeeder.php > DatabaseSeeder > run
1 <?php
2
3 namespace Database\Seeders;
4
5 // use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7
8 class DatabaseSeeder extends Seeder
9 {
10     /**
11      * Seed the application's database.
12      */
13     public function run(): void
14     {
15         // \App\Models\User::factory(10)->create();
16
17         // \App\Models\User::factory()->create([
18         //     'name' => 'Test User',
19         //     'email' => 'test@example.com',
20         // ]);
21
22         $this->call([
23             FighterSeeder::class
24         ]);
25     }
26 }
27
28
```

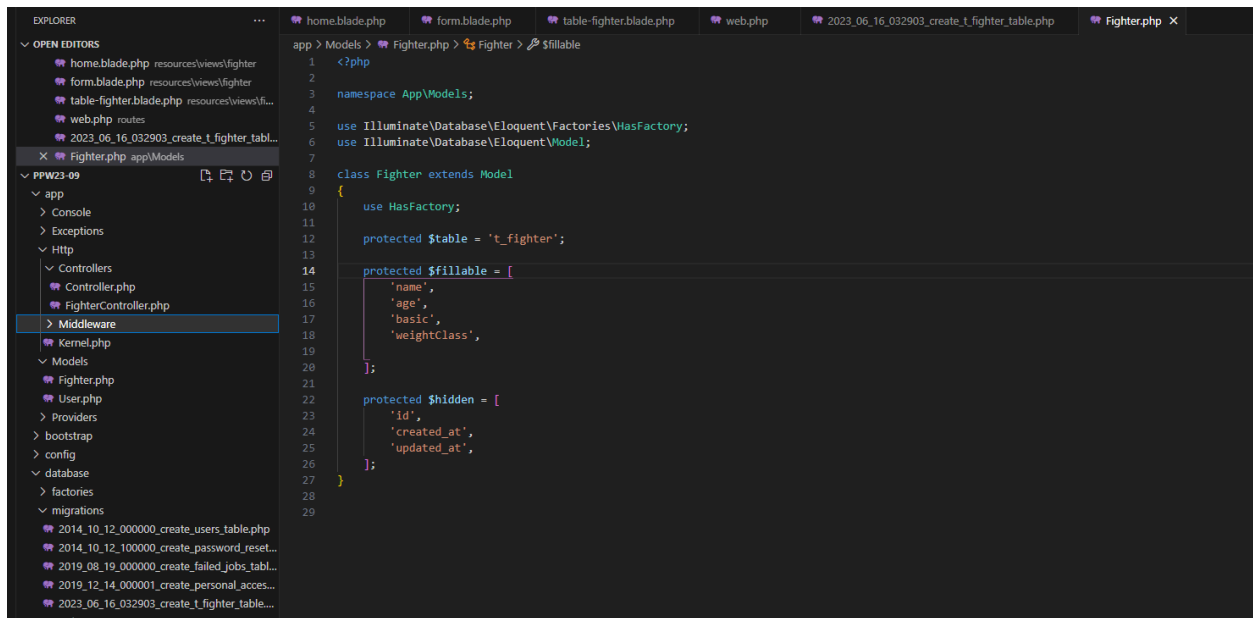


The screenshot shows the phpMyAdmin interface with the 't_fighter' table selected. The table contains 30 rows of data, each representing a fighter. The columns are: id, name, age, basic, weightClass, created_at, and updated_at. The data is as follows:

id	name	age	basic	weightClass	created_at	updated_at
1	Yuni Titi Namaga	38	Brazilian Jiu Jitsu	Welterweight	2023-06-23 06:21:12	2023-06-23 06:21:12
2	Mahesa Dadap Mangunsong M.Ti.	29	Kick Boxing	Welterweight	2023-06-23 06:21:12	2023-06-23 06:21:12
3	Cakrabuana Dadap Halim S.Pi	39	Muay Thai	Middleweight	2023-06-23 06:21:12	2023-06-23 06:21:12
4	Winda Zaida Nasyidah M.Pd	29	Muay Thai	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
5	Cahyo Kuswoyo	39	Brazilian Jiu Jitsu	Lightweight	2023-06-23 06:21:12	2023-06-23 06:21:12
6	Adi Rahmi Perliwi	38	Judo	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
7	Wadi Prasetyo	25	Boxing	Welterweight	2023-06-23 06:21:12	2023-06-23 06:21:12
8	Samet Kamidin Dabukke	37	Boxing	Middleweight	2023-06-23 06:21:12	2023-06-23 06:21:12
9	Ichha Zulaka	29	Judo	Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
10	Maik Banavi Sitotang S.Pi	36	Muay Thai	Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
11	Ilyas Haryanto	31	Brazilian Jiu Jitsu	Welterweight	2023-06-23 06:21:12	2023-06-23 06:21:12
12	Puti Lailasari S.Pi	25	Judo	Lightweight	2023-06-23 06:21:12	2023-06-23 06:21:12
13	Segit Mayasari	32	Judo	Middleweight	2023-06-23 06:21:12	2023-06-23 06:21:12
14	Salimah Agustina	31	Judo	Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
15	Karna Maryadi M.Ti.	30	Kick Boxing	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
16	Taswir Waluyo	25	Brazilian Jiu Jitsu	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
17	Banawa Ardianto	22	Brazilian Jiu Jitsu	Welterweight	2023-06-23 06:21:12	2023-06-23 06:21:12
18	Mulyanto Hadi Nababan S.Sos	28	Boxing	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
19	Handri Thamrin	21	Brazilian Jiu Jitsu	Welterweight	2023-06-23 06:21:12	2023-06-23 06:21:12
20	Jamal Alambara Thamrin	24	Judo	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
21	Puti Yulianti	37	Muay Thai	Middleweight	2023-06-23 06:21:12	2023-06-23 06:21:12
22	Genta Yulianti M.Fam	25	Boxing	Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
23	Gabriela Wahyuni	33	Muay Thai	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
24	Cemeli Ramadan	23	Boxing	Welterweight	2023-06-23 06:21:12	2023-06-23 06:21:12
25	Almaja Kartman Uwais	34	Boxing	Middleweight	2023-06-23 06:21:12	2023-06-23 06:21:12
26	Nadine Syahrini Nuraini	35	Judo	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
27	Jamil Argono Simanjuntak	26	Muay Thai	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
28	Carla Zulaka	30	Judo	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
29	Maide Mutia Puspita	36	Judo	Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12
30	Janie Yolanda	25	Boxing	Light-Heavyweight	2023-06-23 06:21:12	2023-06-23 06:21:12

Gambar di atas merupakan hasil migration table t_fighter. Pada gambar tersebut terdiri dari id, name yang bertipe string varchar(100), age yang bertipe integer, basic yang bertipe string varchar(100), weightClass yang bertipe string varchar(100), dan yang terakhir timestamps. Pada table ini saya membuat seeder dan juga factory. Dengan factory yang menggunakan fake() untuk mengisi nama racak, angka acak untuk umur dalam rentang 20-40. Lalu element acak untuk mengisi basic belad diri, dan yang terakhir element acak untuk mengisi weight class. Lalu pada seedernya saya membuat 30 saja. Saya menggunakan 'faker_locale' nya untuk Indonesia yaitu id_ID. Lalu agar factory dan seedernya dapat bekerja, pada database.php saya menambahkan panggilan untuk class FighterSeeder. Saya menambah kolom foto dalam bentuk string untuk tampilan foto

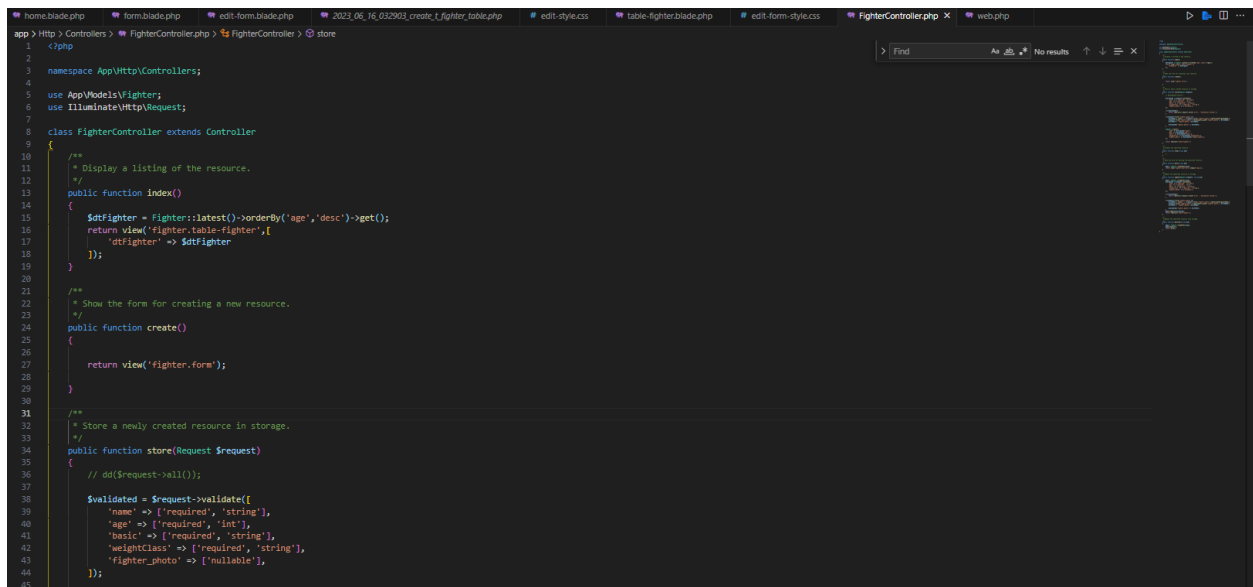
2.Fighter.php dalam app/Models



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Fighter extends Model
9 {
10     use HasFactory;
11
12     protected $table = 't_fighter';
13
14     protected $fillable = [
15         'name',
16         'age',
17         'basic',
18         'weightClass',
19     ];
20
21     protected $hidden = [
22         'id',
23         'created_at',
24         'updated_at',
25     ];
26 }
27
28
29
```

pada model ini terhubung dengan table t_fighter untuk manajemen inputan yang dapat diisi yaitu ada name, age, basic, weightClass dan yang tidak dapat diisi yaitu id, created_at, updated_at.

3. FighterController.php



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Fighter;
6 use Illuminate\Http\Request;
7
8 class FighterController extends Controller
9 {
10     /**
11      * Display a listing of the resource.
12      */
13     public function index()
14     {
15         $fighters = Fighter::latest()->orderBy('age','desc')->get();
16         return view('fighter.table-fighter', [
17             'fighters' => $fighters
18         ]);
19     }
20
21     /**
22      * Show the form for creating a new resource.
23      */
24     public function create()
25     {
26         return view('fighter.form');
27     }
28
29     /**
30      * Store a newly created resource in storage.
31      */
32     public function store(Request $request)
33     {
34         // dd($request->all());
35
36         $validated = $request->validate([
37             'name' => ['required', 'string'],
38             'age' => ['required', 'int'],
39             'basic' => ['required', 'string'],
40             'weightClass' => ['required', 'string'],
41             'fighter_photo' => ['nullable'],
42         ]);
43
44     }
45 }
```

```
home.blade.php | form.blade.php | edit-form.blade.php | 2023_06_16_012903_create_fighter_table.php | edit-style.css | table-fighter.blade.php | edit-form-style.css | FighterController.php | web.php

app > Http > Controllers > FighterController.php > FighterController > store
'fighter_photo' => ['nullable'],
});
44
45
46 if(!$validated){
47     return redirect()->back()->with('error', 'Validation failed!');
48 }
49
50 if($request->file('fighter_photo')){
51     $fileName = time().'.'.$request->file('fighter_photo')->getClientOriginalName();
52     $request->file('fighter_photo')->move(public_path('fighter_photo'), $fileName);
53     $filePath = 'fighter_photo/'.$fileName;
54     $validated['fighter_photo'] = $filePath;
55 }
56
57 $fighter::create([
58     'name' => $validated['name'],
59     'age' => $validated['age'],
60     'basic' => $validated['basic'],
61     'weightClass' => $validated['weightClass'],
62     'fighter_photo' => $validated['fighter_photo'],
63 ]);
64
65 return redirect('table-fighter');
66
67
68 /**
69  * Display the specified resource.
70  */
71 public function show(string $id)
72 {
73     //
74 }
75
76
77 /**
78  * Show the form for editing the specified resource.
79  */
80 public function edit(string $id)
81 {
82     $peg = $fighter::findOrFail($id);
83     return view('fighter.edit-form', compact('peg'));
84 }
85
86 /**
87  * Update the specified resource in storage.
```

```
home.blade.php | form.blade.php | edit-form.blade.php | 2023_06_16_012903_create_fighter_table.php | edit-style.css | table-fighter.blade.php | edit-form-style.css | FighterController.php | web.php

app > Http > Controllers > FighterController.php > FighterController > store
public function edit(string $id)
{
    $peg = $fighter::findOrFail($id);
    return view('fighter.edit-form', compact('peg'));
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, string $id)
{
    $peg = $fighter::findOrFail($id);
    $validated = $request->validate([
        'name' => ['required', 'string'],
        'age' => ['required', 'int'],
        'basic' => ['required', 'string'],
        'weightClass' => ['required', 'string'],
        'fighter_photo' => ['nullable'],
    ]);

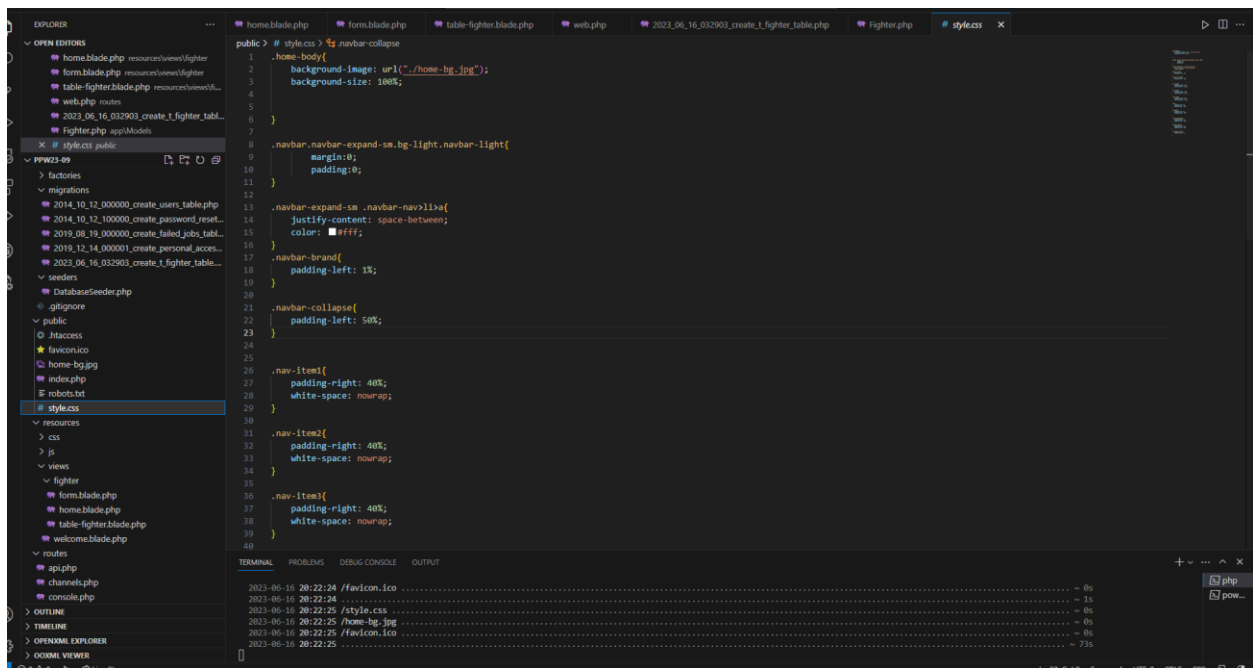
    if(!$validated){
        return redirect()->back()->with('error', 'Validation failed!');
    }

    if($request->file('fighter_photo')){
        $fileName = time().'.'.$request->file('fighter_photo')->getClientOriginalName();
        $request->file('fighter_photo')->move(public_path('fighter_photo'), $fileName);
        $filePath = 'fighter_photo/'.$fileName;
        $validated['fighter_photo'] = $filePath;
    }
    $peg->update($validated);
    return redirect('table-fighter');
}

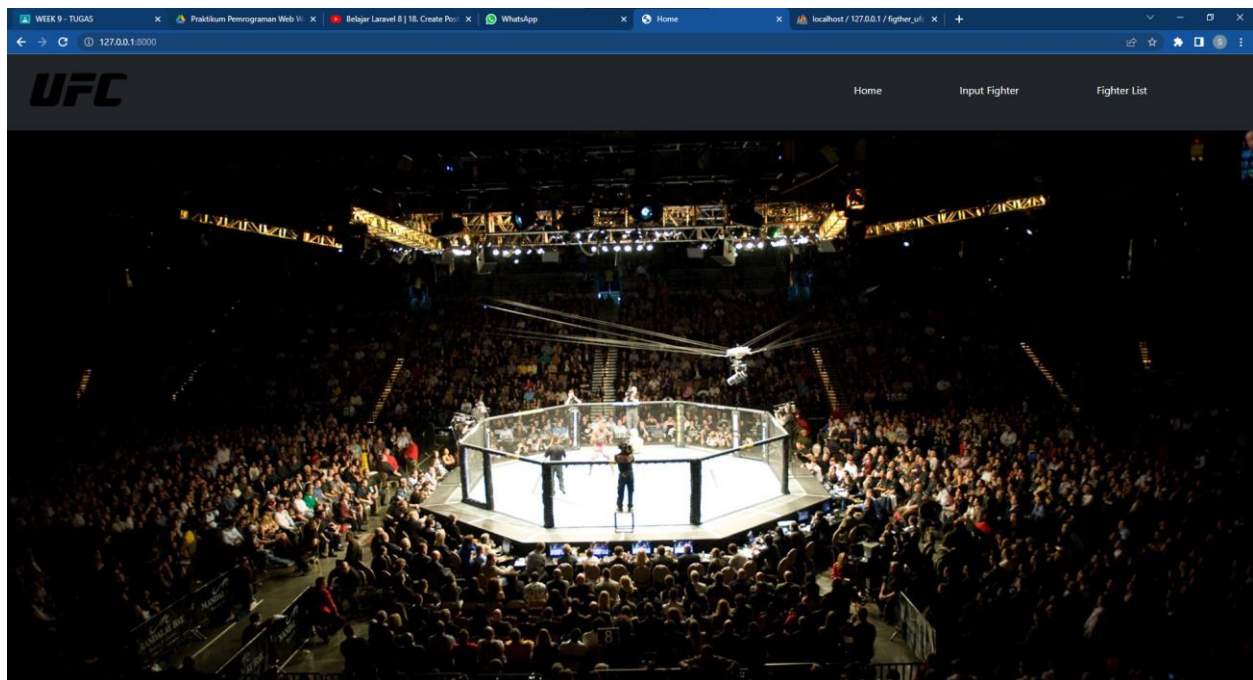
/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    $peg = $fighter::findOrFail($id);
    $peg->delete();
    return back();
}
```

FighterController ini berfungsi untuk manajemen table seperti create untuk membuat row table yang baru, store untuk menyimpan hasil inputan oleh user, index untuk menampilkan halaman. Pada index saya membuat build query dengan order by age secara descending. Lalu hasilnya disimpan pada variable dtFighter. Pada bagian store saya mengimplementasikan validate dan juga file upload photo. Lalu pada edit saya dapat edit fighter tertentu dan juga pada update ketika kita megedit fighter tertentu maka akan menggunakan function update. Dan destroy untuk menghapus fighter.

4.style.css



5. home-blade.php



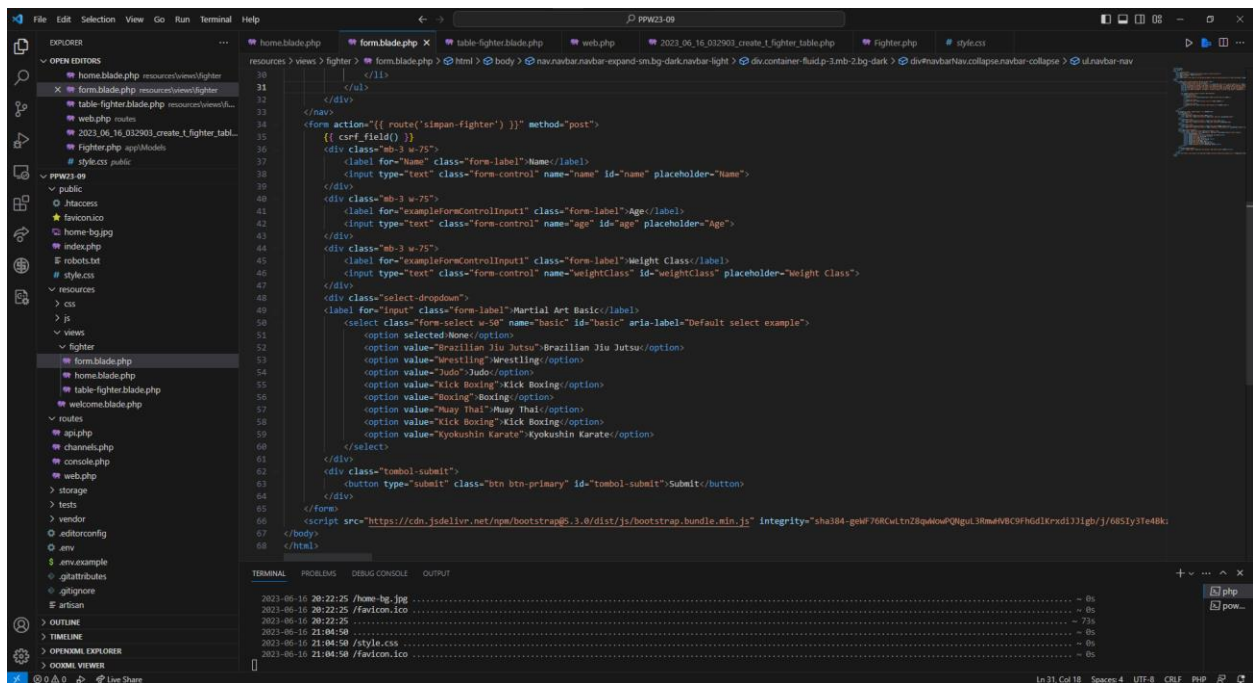
Pada [home.blade.php](#) ini saya menggunakan bantuan bootstrap dalam mendesain. Pada navbar dapat terlihat ada 3 link yaitu Home untuk merujuk ke home, Input Fighter ke ke form-blade.php dan Fighter List ke table.blade.php.

6. form-blade.php

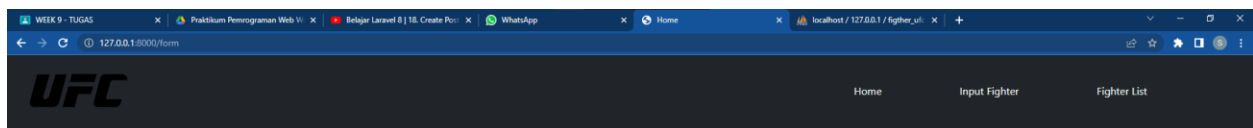
```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <link rel="stylesheet" href="style.css">
7 <title>Home</title>
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyliaIb0a12fVX3J8CjKc5p5M7SpNjef486ghlmu22cdeRth"
9 </head>
10 <body>
11 <div class="nav-bar navbar-expand-sm bg-dark navbar-light">
12 <div class="container-fluid p-3 mb-2 bg-dark">
13 <a class="nav-bar-brand text-white" href="#"><svg xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/
14 (path d="M 48.090296 2.2219693 H 188.79669 L 71.165836 122.02373 C -2.58898,8.4843 2.80518,15.33594 11.54688,15.33594 1 0 0 c 8.74463,0 17.746094,-6.93164 28.744144,
15 (polygon points="226,222,07,6616 226,222,07,6616 346,040,07,6616 333,383,189,144 261,632,189,144 244,269,179,931 244,269,179,931 128,217,179,931 210,77,67,6616 " id
16 (path d="M 411.08988,0,42362933 h 88.47676 c 12.97168,0 34.99021,18.39386967 38.48438,26.92889967 l -1.77539,6.58732 h -79.18888 c -4.86445,0 -8.29395,1.25938 -9.39
17 (path d="M 353.0467,43.453489 h -69.58398 l -0.8352,0.11963 H 217.8758 L 238.48761,-2.8866665e-4 c 45.83857,0 98.87764,0 135.11279,0 l 353.0467,43.453489 l 0 0 z" id
18 </svg></a>
19 <div class="collapse navbar-collapse" id="navbarNav">
20 <ul class="nav navbar-nav">
21 <li class="nav-item">
22 <a class="nav-link" aria-current="page" href="{{route('home')}}">Home</a>
23 </li>
24 <li class="nav-item">
25 <a class="nav-link" href="{{route('form')}}">Input Fighter</a>
26 </li>
27 <li class="nav-item">
28 <a class="nav-link" href="{{route('table-fighter')}}">Fighter List</a>
29 </li>
30 </ul>
31 </div>
32 </div>
33 <div>
34 <form action="{{ route('simpan-fighter') }}" method="post">
35 <div class="mb-3 w-75">
36 <input type="text" class="form-control" name="name" id="name" placeholder="Name">
37 </div>
38 <div class="mb-3 w-75">
39 </div>
40 </div>

```

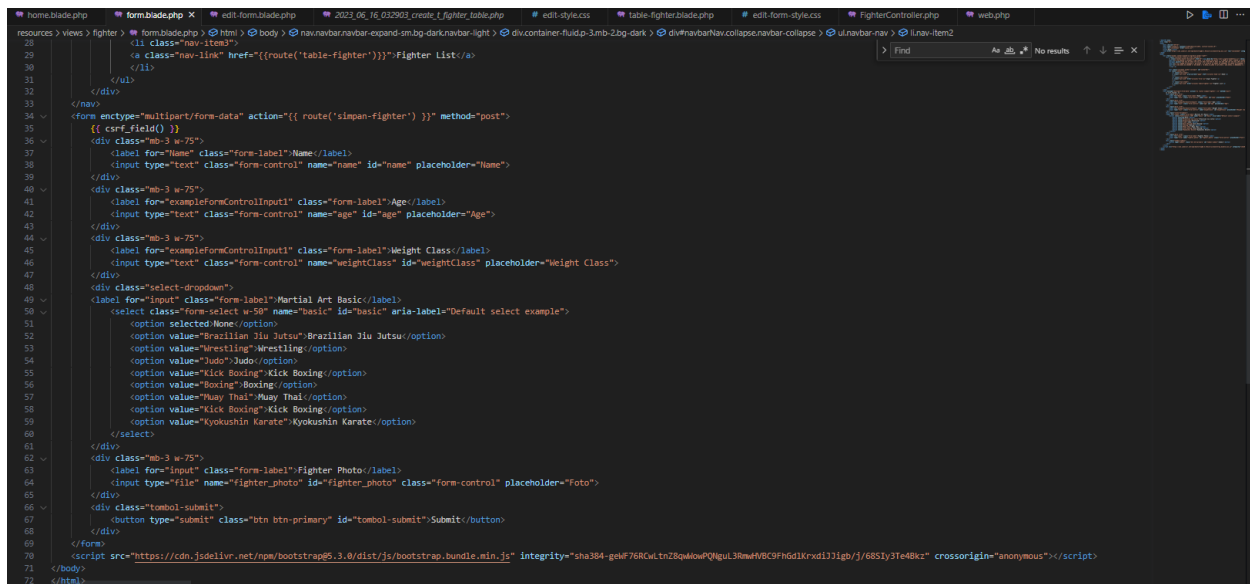
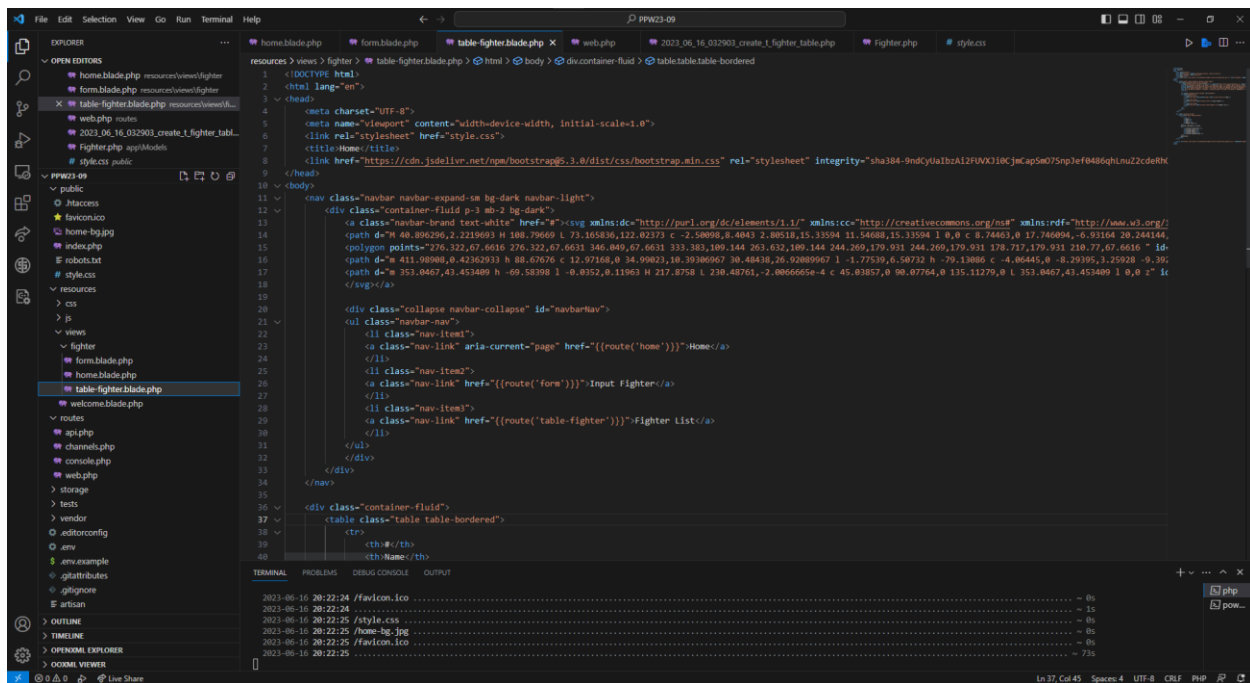



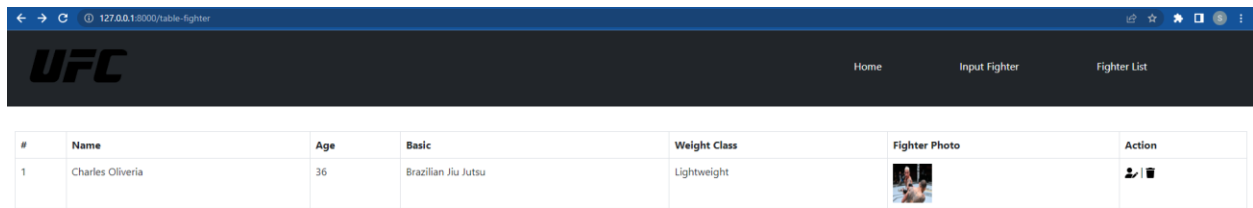
```
30 </div>
31 </div>
32 </div>
33 </div>
34 <form action="{{ route('simpan-fighter') }}" method="post">
35 <!-->
36 <div class="mb-3 w-75">
37 <label for="name" class="form-label">Name</label>
38 <input type="text" class="form-control" name="name" id="name" placeholder="Name">
39 </div>
40 <div class="mb-3 w-75">
41 <label for="exampleFormControlInput1" class="form-label">Age</label>
42 <input type="text" class="form-control" name="age" id="age" placeholder="Age">
43 </div>
44 <div class="mb-3 w-75">
45 <label for="exampleFormControlInput1" class="form-label">Weight Class</label>
46 <input type="text" class="form-control" name="weightClass" id="weightClass" placeholder="Weight Class">
47 </div>
48 <div class="select-dropdown">
49 <label for="input" class="form-label">Martial Art Basic</label>
50 <select class="form-select w-50" name="basic" id="basic" aria-label="Default select example">
51 <option selected="selected">None</option>
52 <option value="Brazilian Jiu Jitsu">Brazilian Jiu Jitsu</option>
53 <option value="Brazilian Wrestling">Brazilian Wrestling</option>
54 <option value="Judo">Judo</option>
55 <option value="Kick Boxing">Kick Boxing</option>
56 <option value="Boxing">Boxing</option>
57 <option value="Muay Thai">Muay Thai</option>
58 <option value="Kick Boxing">Kick Boxing</option>
59 <option value="Kyokushin Karate">Kyokushin Karate</option>
60 </select>
61 </div>
62 <div class="text-center">
63 <button type="submit" class="btn btn-primary" id="tombol-submit">Submit</button>
64 </div>
65 </form>
66 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-gwM9JctHpkr4...>
67 </body>
68 </html>
```






Pada bagaian form.blade.php, saya dibantu dengan desain dari bootstrap untuk navbar dan juga formnya. Formnya terdiri dari inputan Name, Age, Weight Class, dan Martial Art Basic. Lalu ada tombol submit. Jika tombol submit ditekan nnti akan berinteraksi dengan route('simpan-fighter')

7. table-blade.php

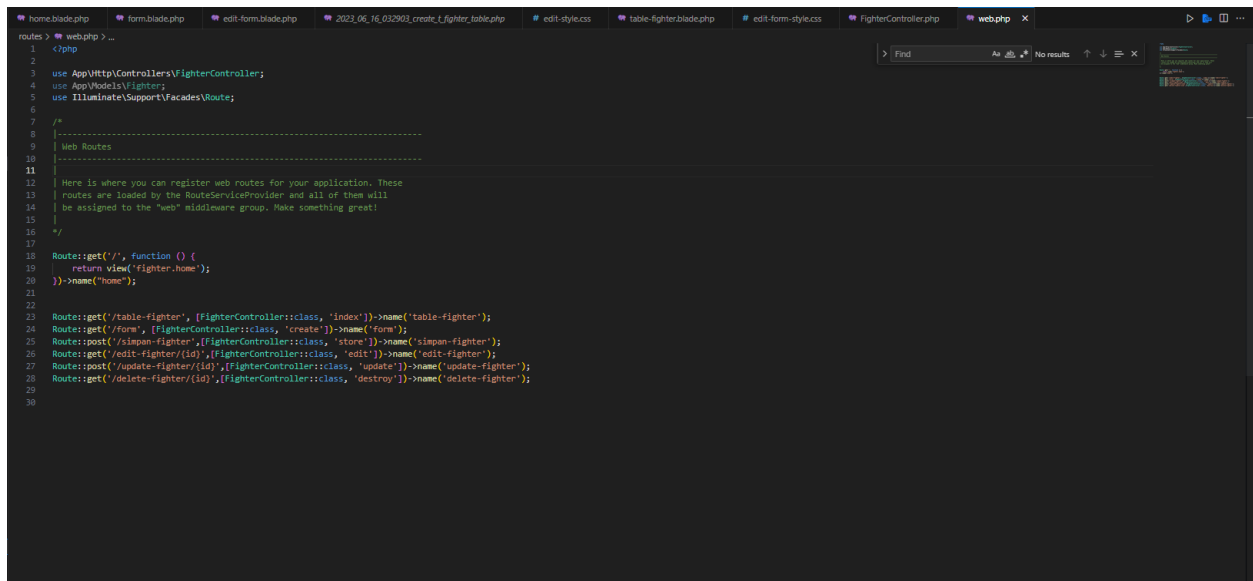




#	Name	Age	Basic	Weight Class	Fighter Photo	Action
1	Charles Oliveira	36	Brazilian Jiu Jitsu	Lightweight		 

Pada bagian ini terdapat table yang merupakan hasil dari inputan dari page form.blade.php atau input fighter. Yang kolomnya terdiri dari #, Name, Age, Basic, Weight Class dan Fighter Photo. Saya menggunakan foreach looping untuk menampilkan hasil dari table nya.

8. web.php



```

1  <?php
2
3  use App\Http\Controllers\FighterController;
4  use App\Models\Fighter;
5  use Illuminate\Support\Facades\Route;
6
7  /*
8  |-----
9  | Web Routes
10 |-----
11 |
12 | Here is where you can register web routes for your application. These
13 | routes are loaded by the RouteServiceProvider and all of them will
14 | be assigned to the "web" middleware group. Make something great!
15 |
16 */
17
18 Route::get('/', function () {
19     return view('fighter.home');
20 })->name('home');
21
22
23 Route::get('/table-fighter', [FighterController::class, 'index'])->name('table-fighter');
24 Route::get('/form', [FighterController::class, 'create'])->name('form');
25 Route::post('/simpan-fighter', [FighterController::class, 'store'])->name('simpan-fighter');
26 Route::get('/edit-fighter/{id}', [FighterController::class, 'edit'])->name('edit-fighter');
27 Route::post('/update-fighter/{id}', [FighterController::class, 'update'])->name('update-fighter');
28 Route::get('/delete-fighter/{id}', [FighterController::class, 'destroy'])->name('delete-fighter');
29
30

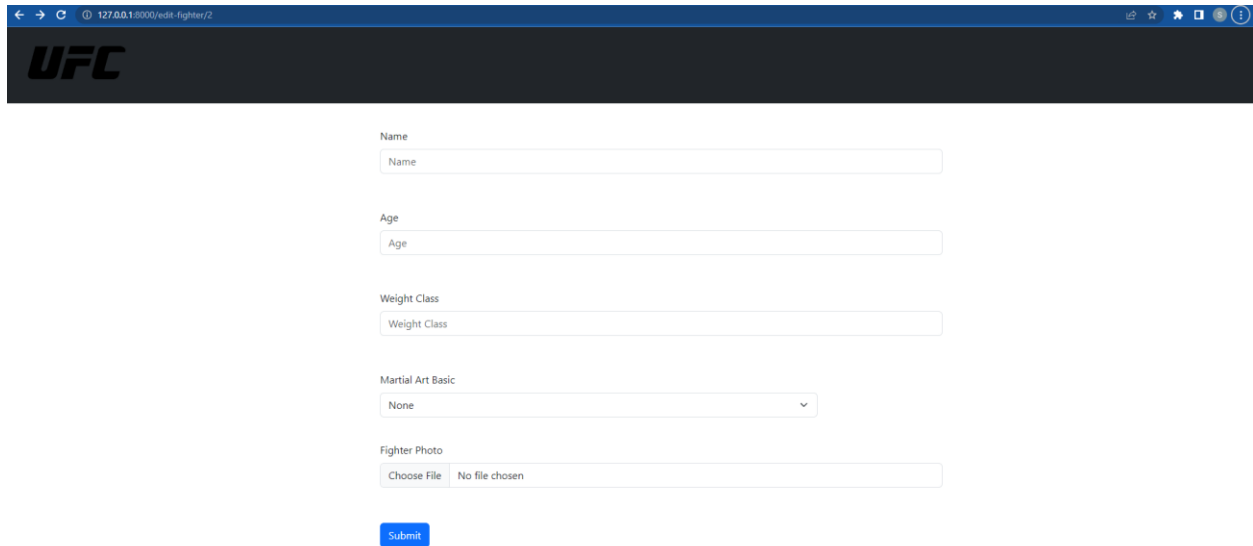
```

gambar diatas terdapat route route yang saya buat yaitu ada 4 metode get dan 2 metode post.

9. edit-form.blade.php

```
resources > views > fighter > edit-form.blade.php > html > head > style > edit
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <link rel="stylesheet" type="text/css" href="edit-style.css">
7 <title>Home</title>
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCylbq9bYse9a5km" crossorigin="anonymous">
9 <style type="text/css">
10 .navbar-brand{
11 padding-left: 10px;
12 }
13
14 .edit{
15 padding-top: 20px;
16 padding-left: 30px;
17 }
18 </style>
19 </head>
20 <body>
21 <div class="container-fluid p-3 mb-2 bg-dark">
22 <a class="navbar-brand text-white" href="#">svg xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:svg="http://www.w3.org/2000/svg"
23 <path d="M 48.896296 2.221693 H 108.7969 L 73.165836 122.02373 C -2.50898,8.4043 2.80518,15.33594 11.54688,15.33594 1 0 0 8.74463 0 17.746894 -6.93164 20.244144 -15.33594 140.5882,2.221693 200.79865,2.221693 276.322,67.6616 276.322,67.6616 346.049,67.6616 333.383,109.144 263.632,109.144 244.289,179.931 244.289,179.931 178.717,179.931 210.77,67.6616 " id="polygon1" transform="translate(-1.0445481e-4,4.4118998e-6) rotate(93.3) h 60.67076 c 12.97188,0 34.99023,10.393806 30.48438,26.920896 1 -1.77239,6.50732 h -79.13886 c -4.06445,0 -8.29395,3.25928 -9.39255,7.24072 1 -24.50277,88.05561 c -1.04983,3.1" id="path1" inkscape:connector-curvature="0"/>
24 </a>
25 </div>
26 <div class="form" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">
27 <div class="mb-3 w-75 edit">
28 <label for="name" class="form-label">Name</label>
29 <input type="text" class="form-control" name="name" id="name" placeholder="Name">
30 </div>
31 <div class="mb-3 w-75 edit">
32 <label for="exampleFormControlInput1" class="form-label">Age</label>
33 <input type="text" class="form-control" name="age" id="age" placeholder="Age">
34 </div>
35 <div class="mb-3 w-75 edit">
36 <label for="exampleFormControlInput1" class="form-label">Weight</label>
37 <input type="text" class="form-control" name="weight" id="weight" placeholder="Weight">
38 </div>
39 <div class="mb-3 w-75 edit">
40 <label for="exampleFormControlInput1" class="form-label">Fighter Photo</label>
41 <input type="file" class="form-control" name="fighter_photo" id="fighter_photo" placeholder="Foto">
42 </div>
43 <div class="text-center">
44 <button type="submit" class="btn btn-primary">Submit</button>
45 </div>
46 </div>
47 </body>
48 </html>
```

```
resources > views > fighter > edit-form.blade.php > html > head > style > edit
31 <div class="form" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">
32 <div class="mb-3 w-75 edit">
33 <label for="name" class="form-label">Name</label>
34 <input type="text" class="form-control" name="name" id="name" placeholder="Name">
35 </div>
36 <div class="mb-3 w-75 edit">
37 <label for="exampleFormControlInput1" class="form-label">Age</label>
38 <input type="text" class="form-control" name="age" id="age" placeholder="Age">
39 </div>
40 <div class="mb-3 w-75 edit">
41 <label for="exampleFormControlInput1" class="form-label">Weight</label>
42 <input type="text" class="form-control" name="weight" id="weight" placeholder="Weight">
43 </div>
44 <div class="mb-3 w-75 edit">
45 <label for="exampleFormControlInput1" class="form-label">Fighter Photo</label>
46 <input type="file" class="form-control" name="fighter_photo" id="fighter_photo" placeholder="Foto">
47 </div>
48 <div class="text-center">
49 <button type="submit" class="btn btn-primary">Submit</button>
50 </div>
51 </div>
52 </body>
53 </html>
```



127.0.0.1:8000/edit-fighter/2

UFC

Name

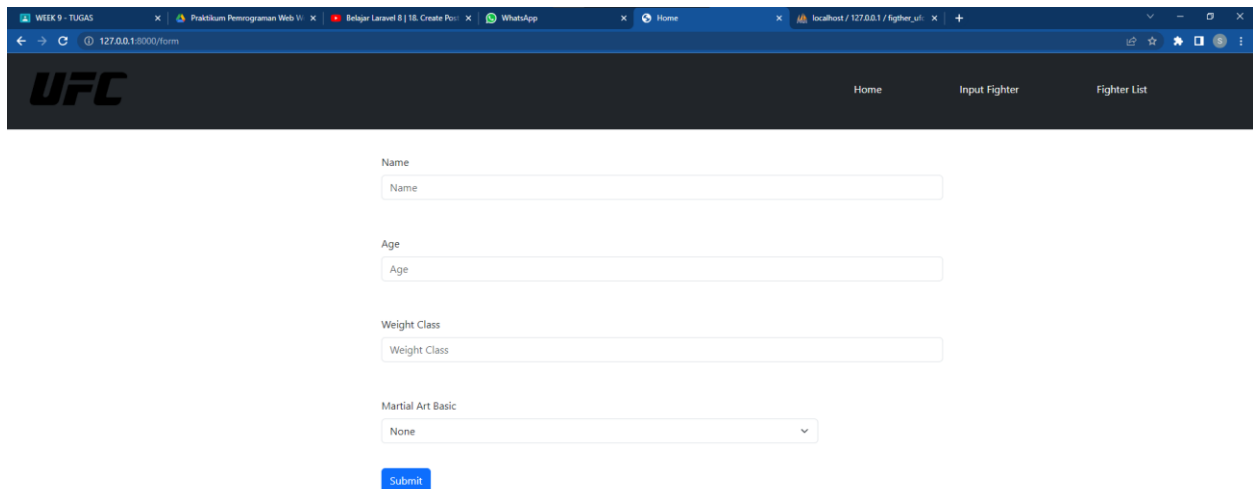
Age

Weight Class

Martial Art Basic

Fighter Photo
 No file chosen

Pada edit-form ini kurang lebih sama sepeerti form.blade.php



WEEK 9 - TUGAS | Praktikum Pemrograman Web | Belajar Laravel 8 | 18. Create Po... | WhatsApp | Home | localhost / 127.0.0.1 / fighter_u... | +

127.0.0.1:8000/form

UFC

Home Input Fighter Fighter List

Name

Age

Weight Class

Martial Art Basic

Pada bagaian form.blade.php, saya dibantu dengan desain dari bootstrap untuk navbar dan juga formnya. Formnya terdiri dari inputan Name, Age, Weight Class, dan Martial Art Basic. Lalu ada tombol submit. Jika tombol submit ditekan nnti akan berinteraksi dengan route('simpan-fighter')

