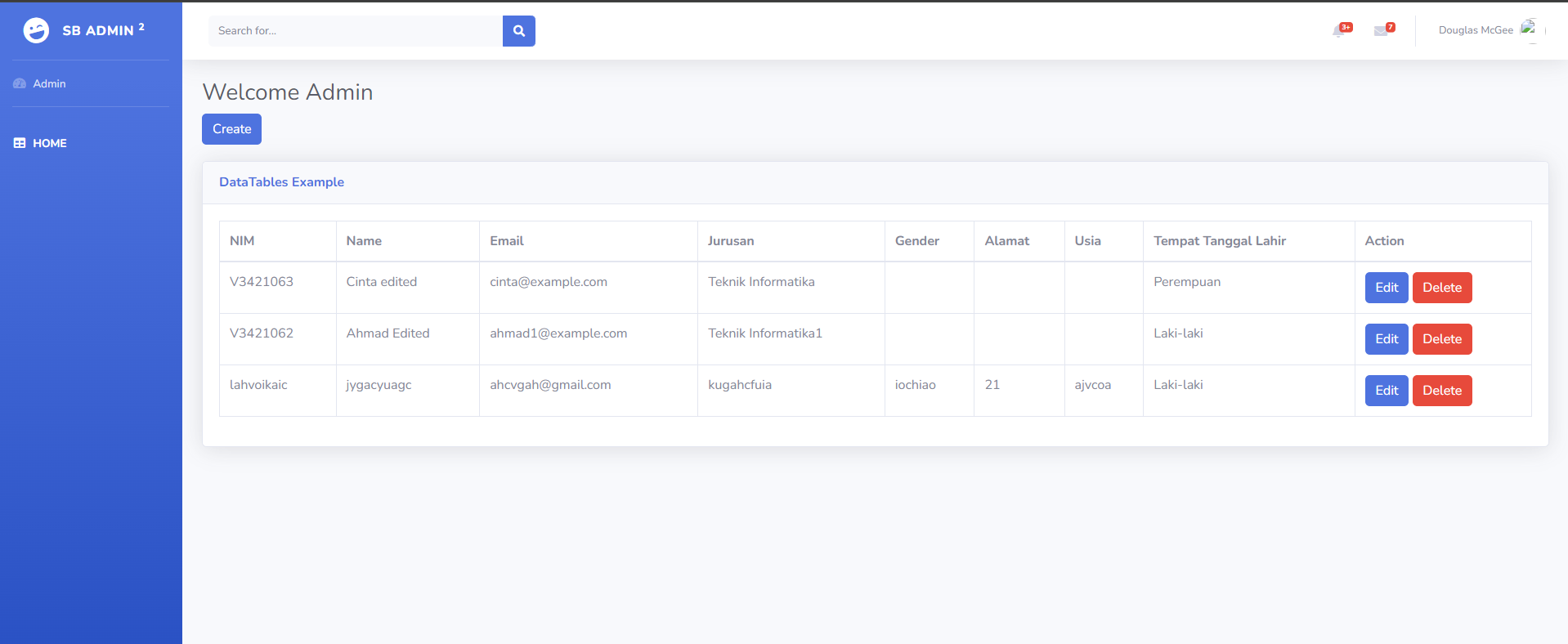
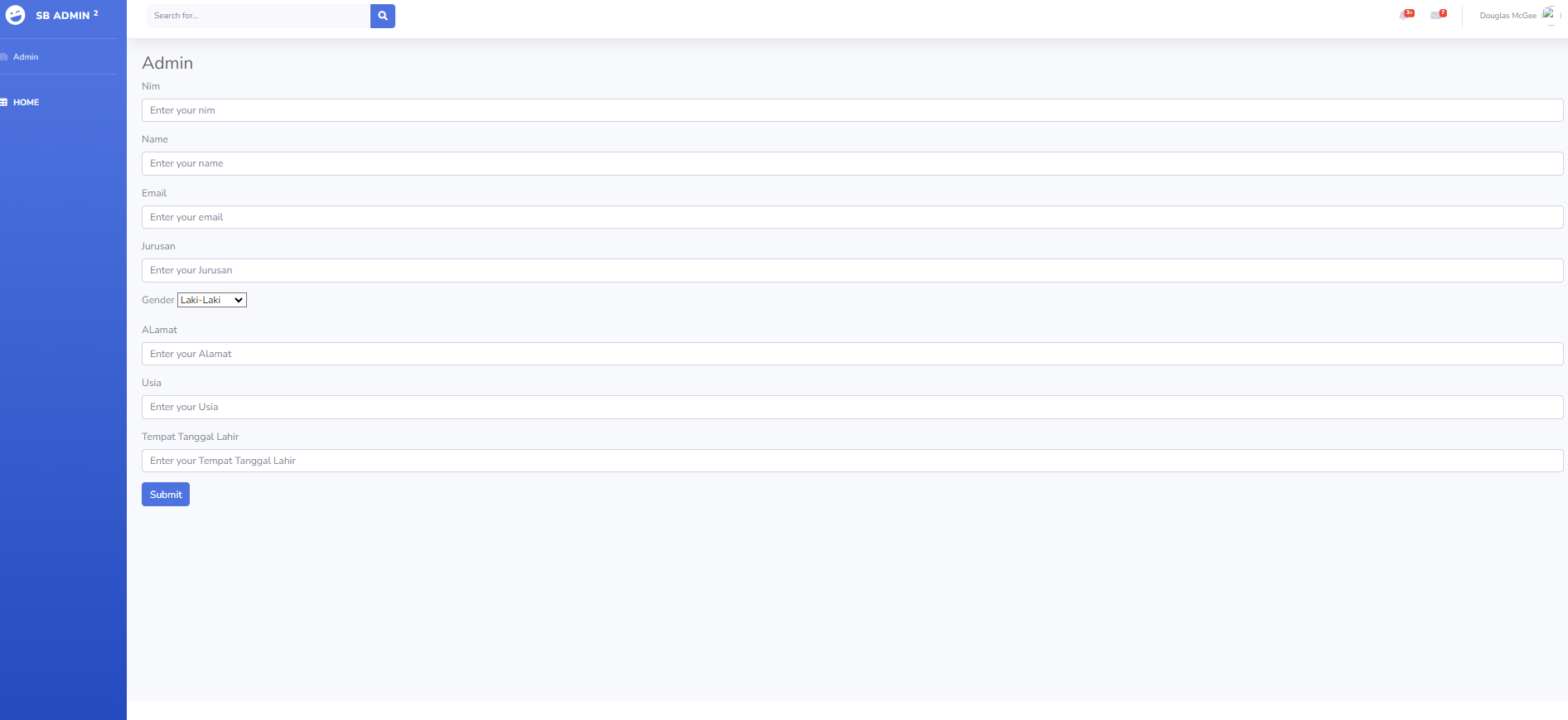
1. Untuk front end saya menggunakan Laravel blade dengan bootsrap,saya menggunakan template dari <https://startbootstrap.com/theme/sb-admin-2>

* Tampilan setelah saya slicing

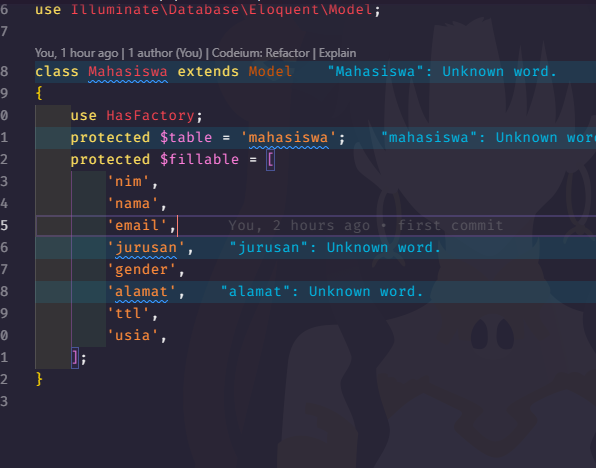


* Tampilan create data



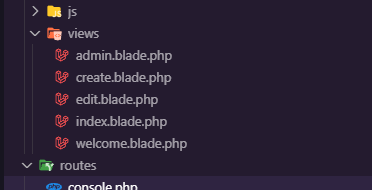
* Tampilan edit data



1. Untuk Backend saya menggunakan Laravel  
    Model  
   

Menggunakan $table untuk memprotect nama table dan $fillable untuk mengatur yang dapat terhubung ke database

* View saya menggunakan blade



* Controller

Fungsi index untuk menampilkan list data

* public function index(Request $request)
* {
* $search = $request->query('search');
* $query = Mahasiswa::query();
* if ($search) {
* $query->where('nama', 'like', "%$search%");
* }
* $mahasiswa = $query->orderBy('nim', 'desc')->get();
* return view('index', compact('mahasiswa', 'search'));
* }

Fungsi create(frontend) dan store(backend) untuk menyimpan data baru

public function create()

    {

        return view('create');

    }

*/\*\**

*\* Store a newly created resource in storage.*

*\*/*

    public function store(Request $request)

    {

        $validated = $request->validate([

            'nim' => 'required',

            'nama' => 'required',

            'email' => 'required|email',

            'jurusan' => 'required',

            'gender' => 'required',

            'alamat' => 'required',

            'ttl' => 'required',

            'usia' => 'required',

        ]);

        Mahasiswa::create($validated);

        return redirect('/');

    }

Fungsi edit(frontend) dan update(backend) untuk mengubah data yang ada

 public function edit(string $id)

    {

        $edit = Mahasiswa::find($id);

        return view('edit')->with('edit', $edit);

    }

*/\*\**

*\* Update the specified resource in storage.*

*\*/*

    public function update(Request $request, string $id)

    {

        $value = [

            'nim' => $request->nim,

            'nama' => $request->nama,

            'email' => $request->email,

            'jurusan' => $request->jurusan,

            'gender' => $request->gender,

            'alamat' => $request->alamat,

            'ttl' => $request->ttl,

            'usia' => $request->usia,

        ];

        Mahasiswa::where('id', $id)->update($value);

        return redirect('/admin');

    }

Fungsi destroy untuk menghapus data

public function destroy(string $id)

    {

        Mahasiswa::where('id', $id)->first()->delete();

        return redirect('/admin');

    }

* Route
* Route::get('/', [MahasiswaController::class, 'index'])->name('mahasiswa.index');
* Route::get('/admin', [MahasiswaController::class, 'admin'])->name('mahasiswa.admin');
* Route::get('/mahasiswa/create', [MahasiswaController::class, 'create'])->name('mahasiswa.create');
* Route::post('/mahasiswa/store', [MahasiswaController::class, 'store'])->name('mahasiswa.store');
* Route::get('/mahasiswa/{id}', [MahasiswaController::class, 'edit'])->name('mahasiswa.edit');
* Route::patch('/mahasiswa/{id}', [MahasiswaController::class, 'update'])->name('mahasiswa.update');
* Route::delete('/mahasiswa/{id}', [MahasiswaController::class, 'destroy'])->name('mahasiswa.destroy');

Untuk alur nya

Page Admin

1. Terdapat tombol create

* Jika ditekan akan teredirect ke page create
* Mengisi data form
* Submit
* Proses penyimpana data dan diredirect ke page index

1. Terdapat table dimana ada list data mahasiswa dengan urutan descending berdasarkan nim ada tombol update dan delete

* Jika dipencet tombol update akan teredirect ke page edit yang sudah menyimpan data berdasarkan id
* Sudah ada nilai yang ada(khusus nim tidak bisa diubah)
* Mengganti data
* Submit
* Proses pengupdatan data dan terdirect ke indes

1. Jika diklik tombol delete

* Data langsung terhapus dan tetap di page yang sama

1. Fitur search merupakan sebuah form]

* Menuliskan keyword berdasarkan nama
* Submit
* Muncul query (<http://127.0.0.1:8000/admin?search=yang> dinputkan)
* Data yang tertampil berubah hanya data berdasarkan query

Untuk Page Home

* mirip dengan Admin,bedanya tidak ada fungsionalitas untuk create,update,dan delete
* terdapat fungsionalitas untuk menunjukan statistic jumlah mahasiswa,dan berdasarkan gender tertentu

1. Untuk database saya menggunakan mysql
2. Code editor menggunakan VSC
3. Framework menggunakan Laravel
4. Dan Laragon(mysql,apache,database management)