

Laporan UAS Sistem Web dengan Laravel 11

Nama: Nabila Winanda Meirani

NIM: H1D022108

Kasus: Sistem Reservasi Hotel

GitHub: https://github.com/meirani/UAS_PemWebII

Mengimplementasikan Laravel Breeze

```
composer require laravel/breeze --dev
php artisan breeze:install
npm install
npm run build
```

File Migration

1. Hotel

```
<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('hotels', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('alamat');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
}
```

```

        */
    public function down(): void
    {
        Schema::dropIfExists('hotels');
    }
};

```

2. Kamar

```

<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('kamars', function (Blueprint $table) {
            $table->id();
            $table->string('nomor');
            $table->foreignId('hotel_id')->constrained(
                table: 'hotels',
                indexName: 'kamar_hotel_id'
            );
            $table->timestamps();
        });
    }

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

```

3. Reservasi

```

<?php

```

```

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

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('reservasis', function (Blueprint $table) {
            $table->id();
            $table->string('tamu');
            $table->foreignId('hotel_id')->constrained(
                table: 'hotels',
                indexName: 'reservasi_hotel_id'
            );
            $table->foreignId('kamar_id')->constrained(
                table: 'kamars',
                indexName: 'reservasi_kamar_id'
            );
            $table->foreignId('user_id')->constrained(
                table: 'users',
                indexName: 'reservasi_user_id'
            );
            $table->date('tanggal');
            $table->timestamps();
        });
    }

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

```

4. User (Tamu)

```

<?php

use Illuminate\Database\Migrations\Migration;

```

```
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });

        Schema::create('password_reset_tokens', function (Blueprint $table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });

        Schema::create('sessions', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->foreignId('user_id')->nullable()->index();
            $table->string('ip_address', 45)->nullable();
            $table->text('user_agent')->nullable();
            $table->longText('payload');
            $table->integer('last_activity')->index();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('users');
        Schema::dropIfExists('password_reset_tokens');
        Schema::dropIfExists('sessions');
    }
}
```

```
}
```

File Model

1. Hotel

```
<?php

namespace App\Models;

use App\Models\Kamar;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Hotel extends Model
{
    use HasFactory;

    protected $guarded = ['id'];

    public function kamar(): HasMany
    {
        return $this->hasMany(Kamar::class, 'hotel_id');
    }

    public function reservasi(): HasMany
    {
        return $this->hasMany(Reservasi::class, 'hotel_id');
    }
}
```

2. Kamar

```
<?php

namespace App\Models;

use App\Models\Hotel;
use App\Models\Reservasi;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```

class Kamar extends Model
{
    use HasFactory;

    protected $guarded = ['id'];

    public function hotel(): BelongsTo
    {
        return $this->belongsTo(Hotel::class);
    }

    public function reservasi(): HasMany
    {
        return $this->hasMany(Reservasi::class, 'kamar_id');
    }
}

```

3. Reservasi

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Reservasi extends Model
{
    use HasFactory;

    public function hotel(): BelongsTo
    {
        return $this->belongsTo(Hotel::class);
    }

    public function kamar(): BelongsTo
    {
        return $this->belongsTo(Kamar::class);
    }

    public function user(): BelongsTo
    {
        return $this->belongsTo(User::class);
    }
}

```

```
}
```

4. User (Tamu)

```
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Notifications\Notifiable;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    public function reservasi(): HasMany
    {
        return $this->hasMany(Reservasi::class, 'user_id');
    }

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
```

```

        * Get the attributes that should be cast.
        *
        * @return array<string, string>
        */
protected function casts(): array
{
    return [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
}
}

```

File Controller

1. Hotel

```

<?php

namespace App\Http\Controllers;

use App\Models\Hotel;
use Illuminate\Http\Request;

class HotelController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $hotel = Hotel::all();
        return view('hotels', compact('hotel'));
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('hotel.create');
    }

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

```



```

    */
public function store(Request $request)
{
    $validateData = $request->validate([
        'nama' => 'required|max:20',
        'alamat' => 'required|max:200'
    ]);

    Hotel::create($validateData);

    return redirect('/hotels');
}

/**
 * Display the specified resource.
 */
public function show(Hotel $hotel)
{
    //
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(Hotel $hotel)
{
    return view('hotel.edit', [
        'hotel' => $hotel
    ]);
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, Hotel $hotel)
{
    $validateData = $request->validate([
        'nama' => 'required|max:20',
        'alamat' => 'required|max:200'
    ]);

    Hotel::where('id', $hotel->id)->update($validateData);

    return redirect('/hotels');
}

```

```

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(Hotel $hotel)
    {
        Hotel::destroy($hotel->id);

        return redirect('/hotels');
    }
}

```

2. Kamar

```

<?php

namespace App\Http\Controllers;

use App\Models\Hotel;
use App\Models\Kamar;
use App\Http\Requests\StoreKamarRequest;
use App\Http\Requests\UpdateKamarRequest;

class KamarController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $kamar = Kamar::all();
        return view('kamars', compact('kamar'));
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('kamar.create', [
            'hotels' => Hotel::all()
        ]);
    }

    /**

```

```

    * Store a newly created resource in storage.
    */
public function store(StoreKamarRequest $request, Hotel $hotel)
{
    $validateData = $request->validate([
        'nomor' => 'required|max:10'
    ]);

    $validateData['hotel_id'] = $hotel->id;

    Hotel::create($validateData);

    return redirect('/kamars');
}

/**
 * Display the specified resource.
 */
public function show(Kamar $kamar)
{
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(Kamar $kamar)
{
    return view('kamar.edit', [
        'kamar' => $kamar,
        'hotels' => Hotel::all()
    ]);
}

/**
 * Update the specified resource in storage.
 */
public function update(UpdateKamarRequest $request, Kamar $kamar, Hotel
$hotel)
{
    $validateData = $request->validate([
        'nomor' => 'required|max:10'
    ]);

    $validateData['hotel_id'] = $hotel->id;

```

```

        Kamar::where('id', $kamar->id)->update($validateData);

        return redirect('/kamars');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(Kamar $kamar)
    {
        Kamar::destroy($kamar->id);

        return redirect('/kamars');
    }
}

```

3. Reservasi

```

<?php

namespace App\Http\Controllers;

use App\Models\User;
use App\Models\Hotel;
use App\Models\Kamar;
use App\Models\Reservasi;
use App\Http\Requests\StoreReservasiRequest;
use App\Http\Requests\UpdateReservasiRequest;

class ReservasiController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $reservasi = Reservasi::all();
        return view('reservasis', compact('reservasi'));
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {

```

```

        return view('reservasi.create', [
            'hotels' => Hotel::all(),
            'kamars' => Kamar::all(),
            'users' => User::all(),
        ]);
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(StoreReservasiRequest $request, Hotel $hotel, Kamar
    $kamar, User $user)
    {
        $validateData = $request->validate([
            'tamu' => 'required|max:20',
            'tanggal' => 'required'
        ]);

        $validateData['hotel_id'] = $hotel->id;
        $validateData['kamar_id'] = $kamar->id;
        $validateData['user_id'] = $user->id;

        Reservasi::create($validateData);

        return redirect('/reservasis');
    }

    /**
     * Display the specified resource.
     */
    public function show(Reservasi $reservasi)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(Reservasi $reservasi)
    {
        return view('reservasi.edit', [
            'reservasi' => $reservasi,
            'hotels' => Hotel::all(),
            'kamars' => Kamar::all(),
            'users' => User::all()
        ]);
    }
}

```

```

    });
}

/**
 * Update the specified resource in storage.
 */
public function update(UpdateReservasiRequest $request, Reservasi $reservasi,
Hotel $hotel, Kamar $kamar, User $user)
{
    $validateData = $request->validate([
        'tamu' => 'required|max:20',
        'tanggal' => 'required'
    ]);

    $validateData['hotel_id'] = $hotel->id;
    $validateData['kamar_id'] = $kamar->id;
    $validateData['user_id'] = $user->id;

    Reservasi::where('id', $reservasi->id)->update($validateData);

    return redirect('/reservasis');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(Reservasi $reservasi)
{
    Reservasi::destroy($reservasi->id);

    return redirect('/reservasis');
}
}

```

4. User (Tamu)

```

<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;

class UserController extends Controller
{

```

```
/**
 * Display a listing of the resource.
 */
public function index()
{
    $user = User::all();
    return view('users', compact('user'));
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    return view('user.create');
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    $validateData = $request->validate([
        'name' => 'required|max:20',
        'email' => 'required|max:50',
        'password' => 'required'
    ]);

    User::create($validateData);

    return redirect('/users');
}

/**
 * Display the specified resource.
 */
public function show(User $user)
{
    //
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(User $user)
```

```
{
    return view('user.edit', [
        'user' => $user
    ]);
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, User $user)
{
    $validateData = $request->validate([
        'name' => 'required|max:20',
        'email' => 'required|max:50',
        'password' => 'required'
    ]);

    User::where('id', $user->id)->update($validateData);

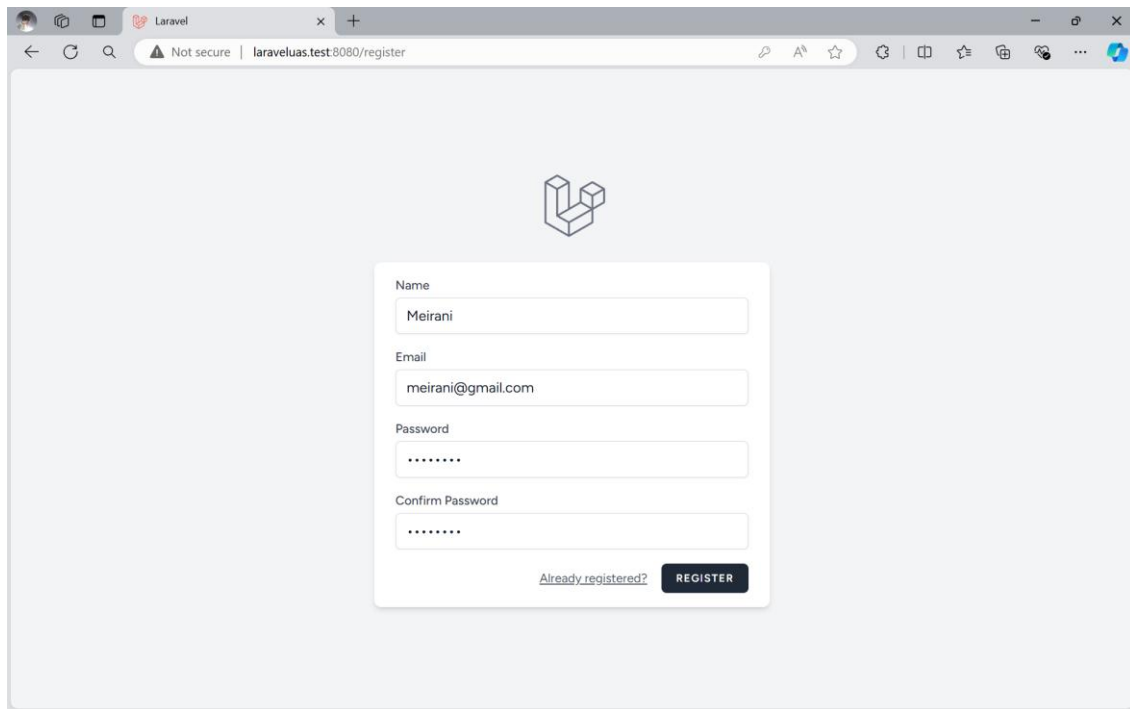
    return redirect('/users');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(User $user)
{
    User::destroy($user->id);

    return redirect('/users');
}
}
```

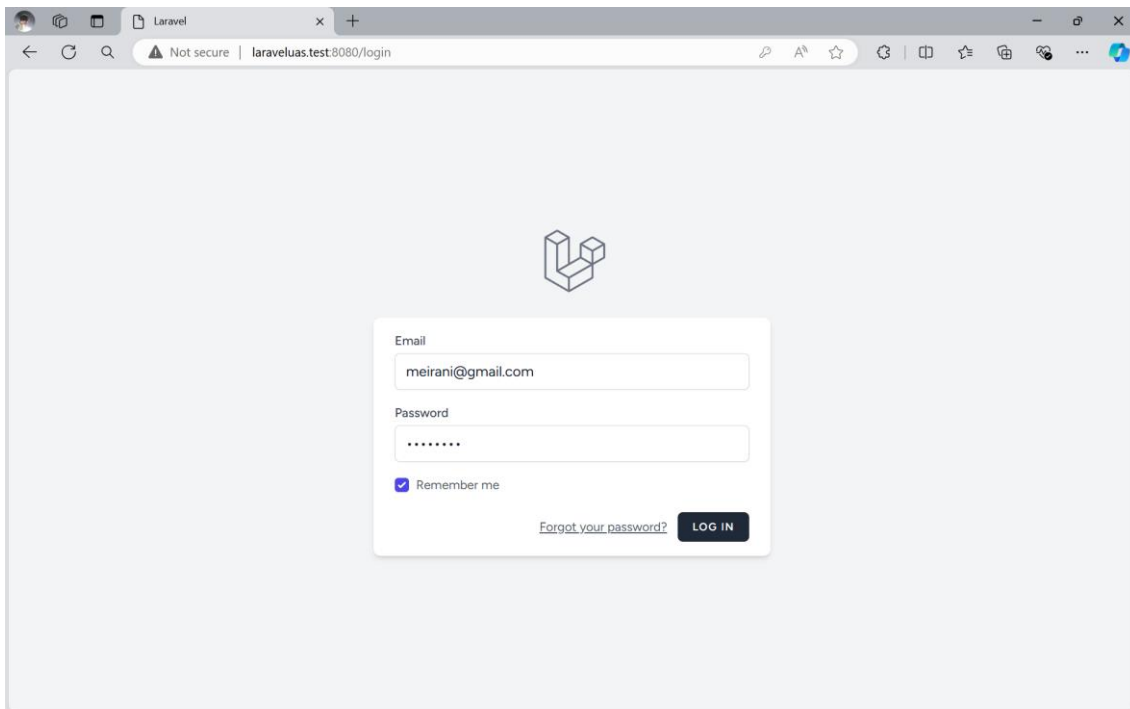

Screenshoot Tampilan

1. Register



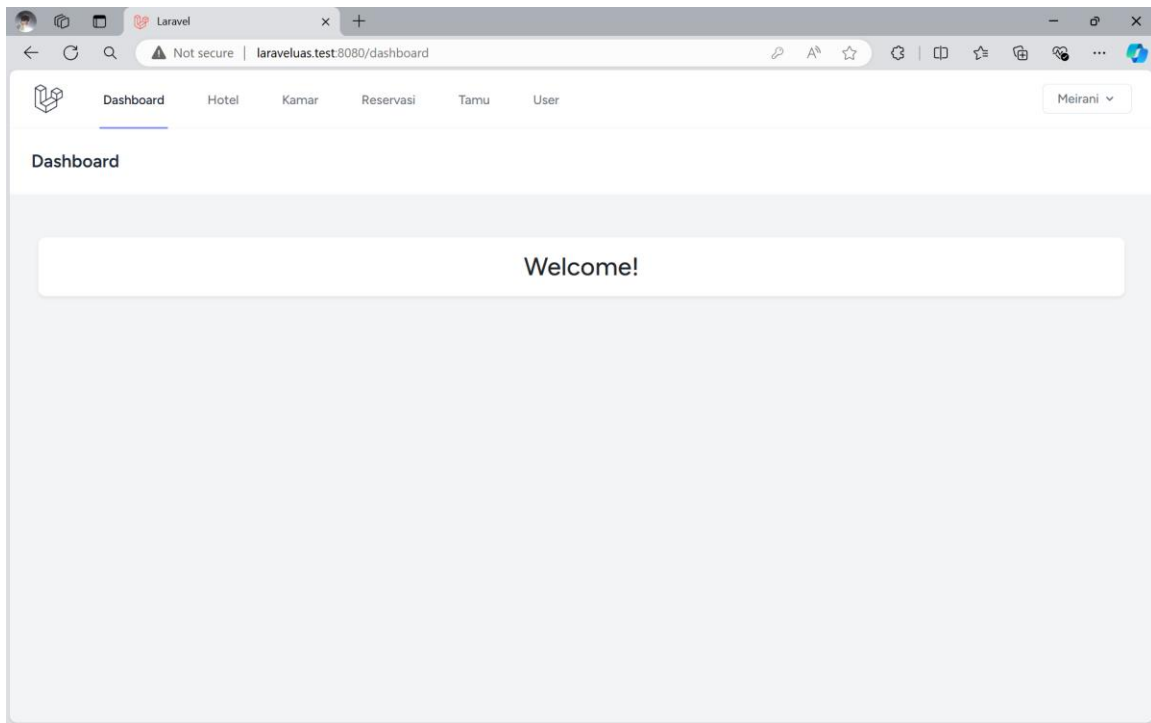
A screenshot of a web browser displaying the Laravel registration page. The browser's address bar shows the URL `laraveluas.test:8080/register`. The page features the Laravel logo at the top center. Below the logo is a registration form with the following fields: "Name" (containing "Meirani"), "Email" (containing "meirani@gmail.com"), "Password" (masked with dots), and "Confirm Password" (masked with dots). At the bottom of the form, there is a link for "[Already registered?](#)" and a dark blue "REGISTER" button.

2. Login

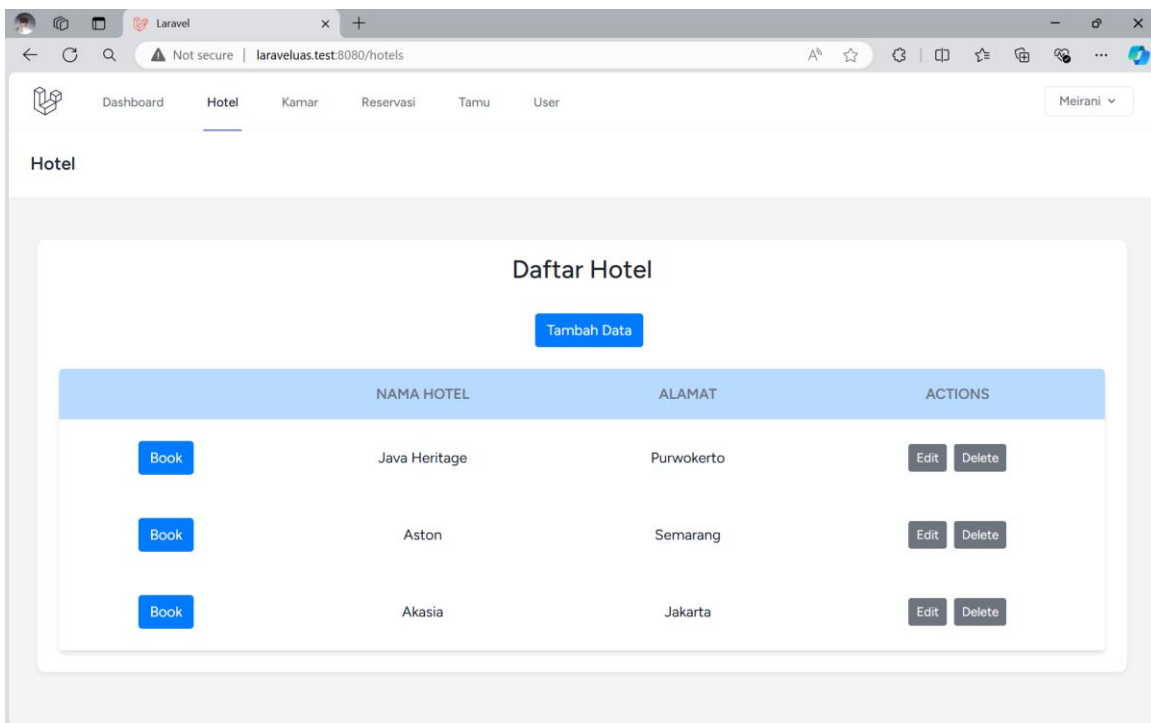


A screenshot of a web browser displaying the Laravel login page. The browser's address bar shows the URL `laraveluas.test:8080/login`. The page features the Laravel logo at the top center. Below the logo is a login form with the following fields: "Email" (containing "meirani@gmail.com") and "Password" (masked with dots). Below the password field is a checked checkbox labeled "Remember me". At the bottom of the form, there is a link for "[Forgot your password?](#)" and a dark blue "LOG IN" button.

3. Dashboard



4. Hotel



view hotel

Hotel

Create Data Hotel

Nama Hotel

Alamat

Create!

create hotel

Hotel

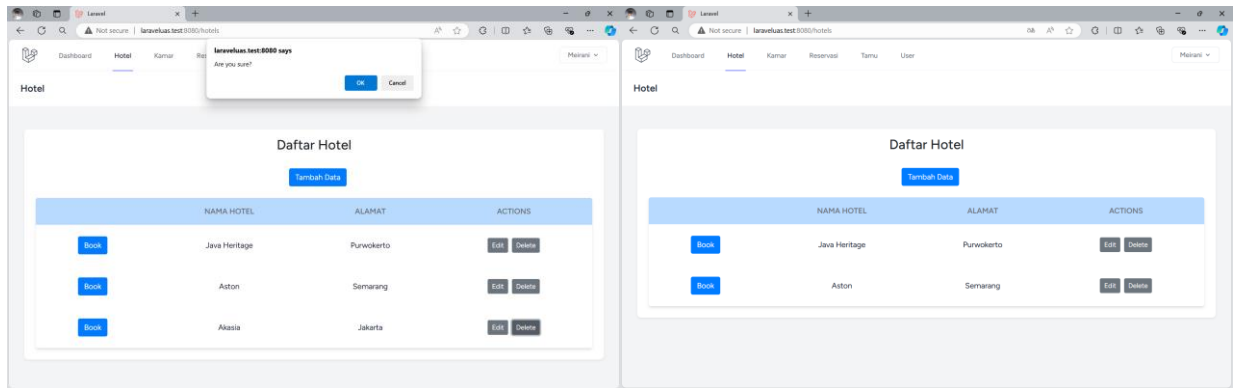
Update Data Hotel

Nama Hotel

Alamat

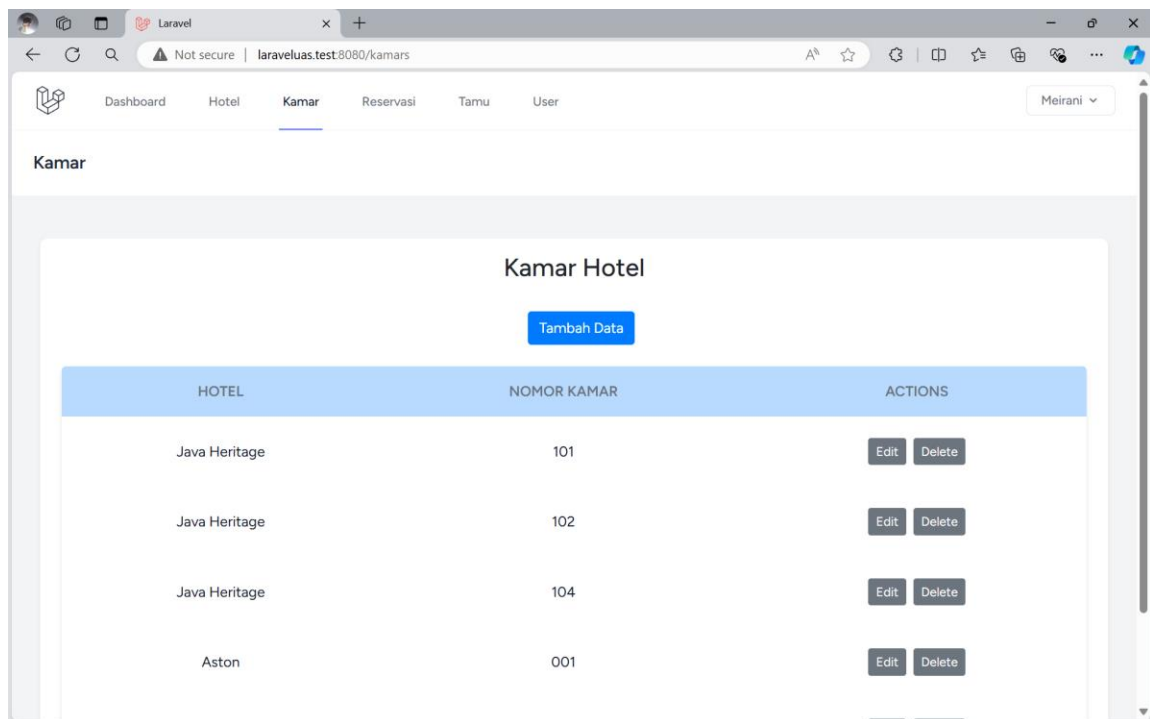
Update!

update hotel



delete hotel

5. Kamar



view kamar

Laravel x +

Not secure | laraveluas.test:8080/kamar/create

Dashboard Hotel Kamar Reservasi Tamu User Meirani v

Kamar

Create Data Kamar

Pilih Hotel Java Heritage v

Nomor

Java Heritage
Aston

Create!

create kamar

Laravel x +

Not secure | laraveluas.test:8080/kamar/1/edit

Dashboard Hotel Kamar Reservasi Tamu User Meirani v

Kamar

Update Data Kamar

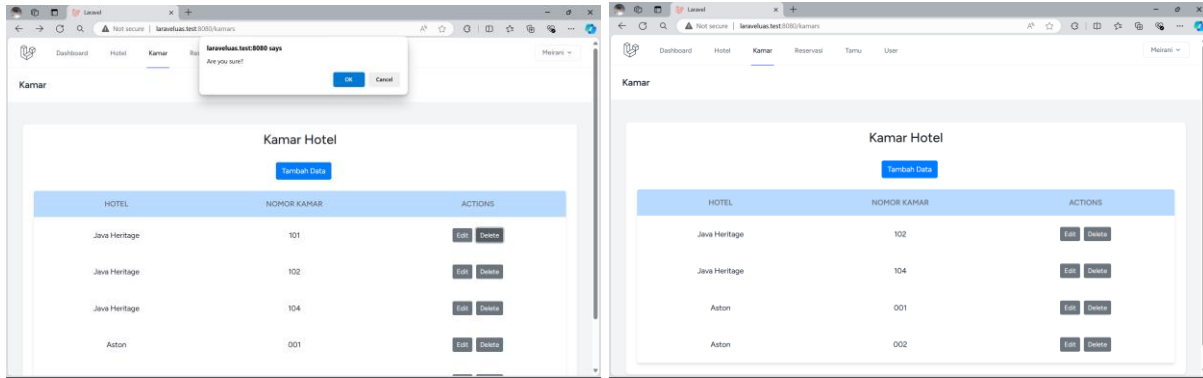
Pilih Hotel Java Heritage v

Nomor

Java Heritage
Aston

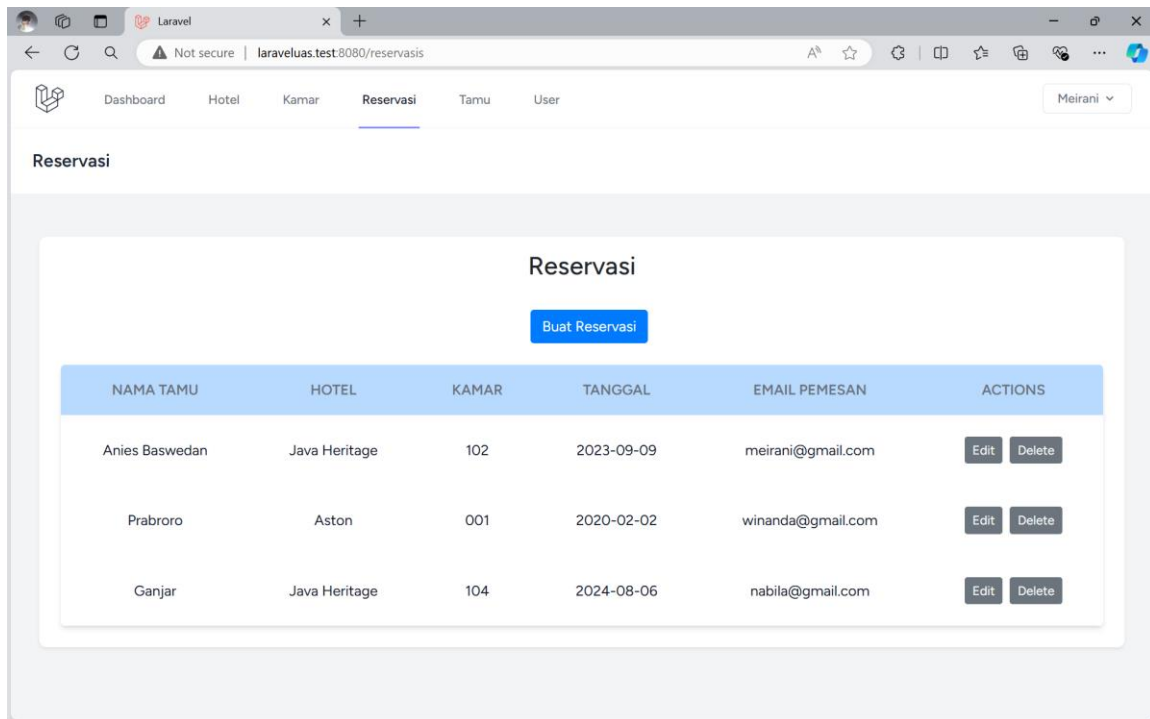
Update!

update kamar



delete kamar

6. Reservasi



view reservasi

Laravel x +

Not secure | laraveluas.test:8080/reservasi/create

Dashboard Hotel Kamar Reservasi Tamu User Meirani

Reservasi

Reservasi Kamar Hotel

Tamu

Pilih Hotel Java Heritage

Pilih Kamar 102

Tanggal

102
104
001
002

create reservasi

Laravel x +

Not secure | laraveluas.test:8080/reservasi/3/edit

Dashboard Hotel Kamar Reservasi Tamu User Meirani

Reservasi

Update Data Reservasi

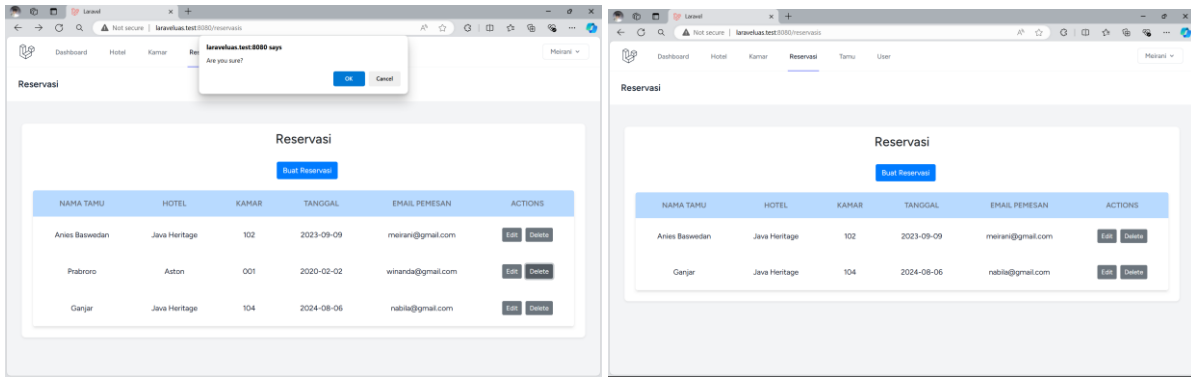
Tamu

Pilih Hotel Java Heritage

Pilih Kamar 104

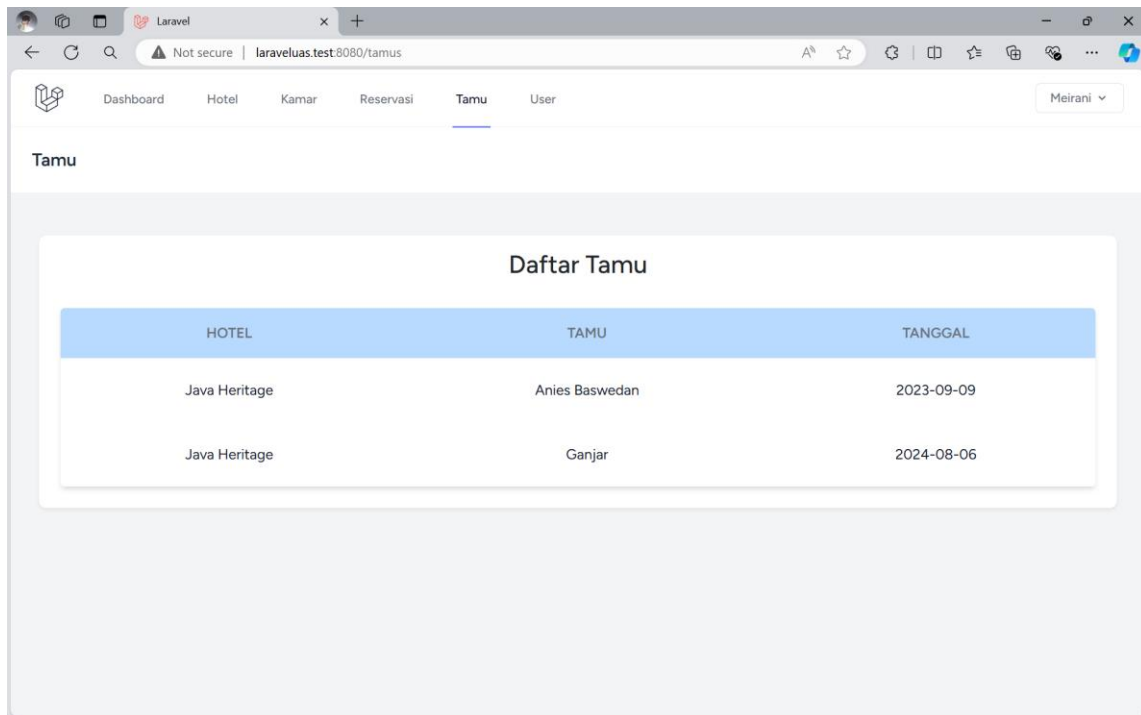
Tanggal

update reservasi

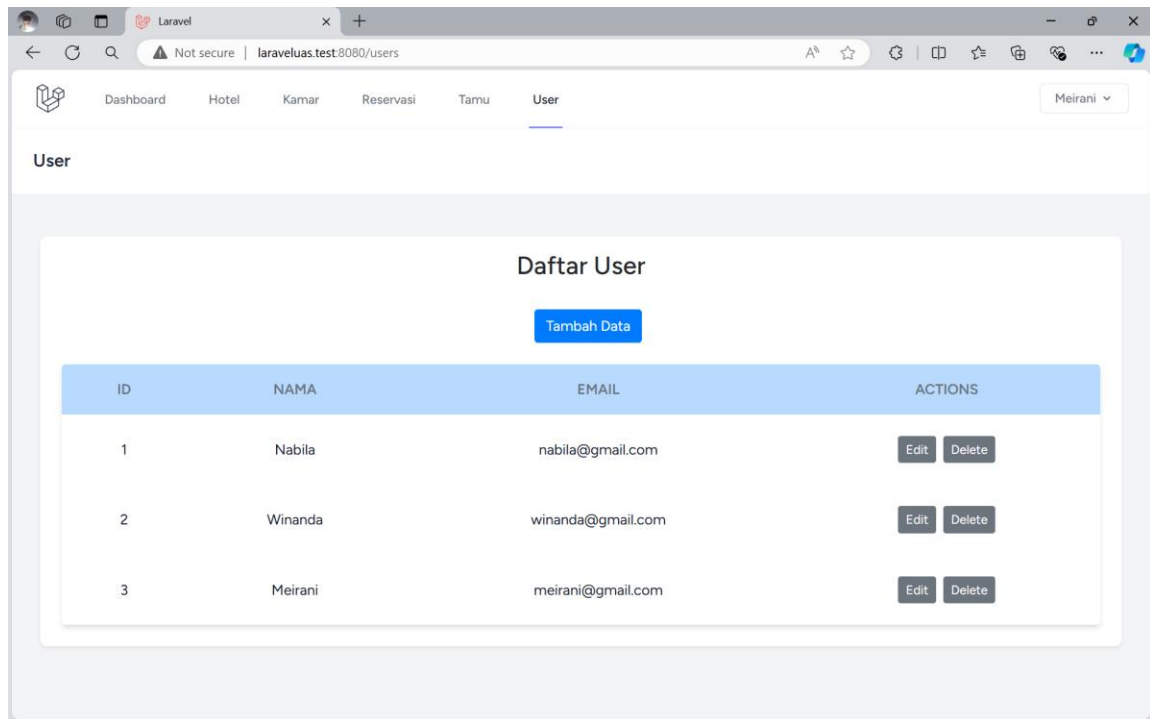


delete reservasi

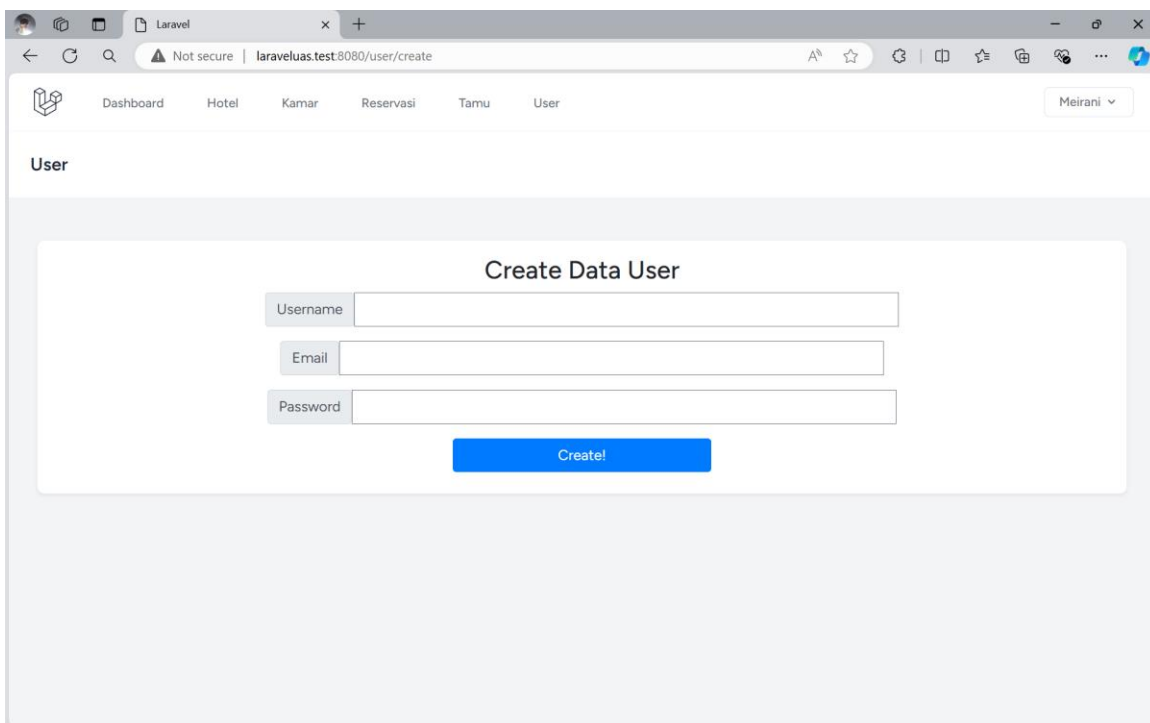
7. Tamu (User)



view tamu



view user



create user

Laravel

Not secure | laravelus.test:8080/user/3/edit

Dashboard Hotel Kamar Reservasi Tamu User Meirani

User

Update Data User

Username Meirani

Email meirani@gmail.com

Password

Create!

update user

Laravel

Not secure | laravelus.test:8080/users

Dashboard Hotel Kamar Reservasi Tamu User Meirani

User

laravelus.test:8080 says
Are you sure?
OK Cancel

Daftar User

Tampilkan Data

ID	NAMA	EMAIL	ACTIONS
1	Nabila	nabila@gmail.com	Edit Delete
2	Winanda	winanda@gmail.com	Edit Delete
3	Meirani	meirani@gmail.com	Edit Delete

delete user