

UJIAN AKHIR SEMESTER PEMROGRAMAN WEB 2

Zia Khusnul Fauzi Akhmad

H1D022023

Database Migration :

```
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();
    });

    Schema::create('product', function (Blueprint $table) {
        $table->id();
        $table->string('name');
```

```
    $table->text('description');
    $table->string('image');
    $table->decimal('price', 10, 2);
    $table->integer('stock');
    $table->timestamps();
});
```

```
Schema::create('order', function (Blueprint $table) {
    $table->id();
    $table->string('invoice');
    $table->foreignId('user_id')->constrained();
    $table->integer('quantity');
    $table->enum('status', ['unpaid', 'paid', 'delivered', 'canceled']);
    $table->timestamps();
});
```

```
Schema::create('order_detail', function (Blueprint $table) {
    $table->id();
    $table->foreignId('order_id')->constrained('order');
    $table->foreignId('product_id')->constrained('product');
    $table->timestamps();
});
```

```
Schema::create('payment', function (Blueprint $table) {
    $table->id();
    $table->foreignId('order_id')->constrained('order');
    $table->string('payment_number');
    $table->decimal('amount', 10, 2);
    $table->timestamps();
});
```

Model :

1. Order

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Order extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table = 'order';
```

```
    protected $fillable = ['product_id', 'quantity', 'invoice', 'status', 'user_id'];
```

```
    public function products()
```

```
    {
```

```
        return $this->belongsToMany(Product::class)
```

```
            ->withTimestamps();
```

```
    }
```

```
    public function payment()
```

```
    {
```

```
        return $this->hasOne(Payment::class);
```

```
    }
```

```
    public function user()
```

```
    {
```

```
        return $this->belongsTo(User::class);
```

```
    }
```

```
}
```

Products

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Product extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table = 'product';
```

```
    protected $fillable = ['name', 'description', 'price',  
'stock','image'];
```

```
    protected $casts = [  
        'price' => 'decimal:2',
```

```
    ];
```

```
};
```

```
    public function orders()
```

```
    {
```

```
        return $this->belongsToMany(Order::class)
```

```
            ->withTimestamps();
```

```
    }
```

```
}
```

Payment

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```

class Payment extends Model
{
    use HasFactory;
    protected $table = 'payment';
    protected $fillable = ['order_id', 'payment_number', 'amount'];

    public function order()
    {
        return $this->belongsTo(Order::class);
    }
}

```

User :

```

<?php
namespace App\Models;
// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use App\Models\Pokemon;

class User extends Authenticatable
{
    use HasFactory, Notifiable;
    protected $fillable = [
        'name',
        'email',
        'password',
    ];
    protected $hidden = [

```

```

        'password',
        'remember_token',
    ];
    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }

    public function orders()
    {
        return $this->hasMany(Order::class);
    }
}

```

Controller

Order :

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Order;
```

```
use App\Models\Product;
```

```
use Illuminate\Support\Facades\Auth;
```

```
use Illuminate\View\View;
```

```
use Illuminate\Http\Request;
```

```
class OrderController extends Controller
```

```
{
```

```
    public function index() : View
```

```
    {
```

```

        if(!Auth::check()){
            return view('auth.login');
        }
        $orders = Order::where('user_id', auth()->id())->paginate(5);
        return view('order.index',[
            'orders' => $orders,
            'title' => 'Order List'
        ]);
    }
    public function create() : View
    {
        if(!Auth::check()){
            return view('auth.login');
        }
        $products = Product::all();
        return view('order.create',[
            'title' => 'Create Order',
            'products' => $products
        ]);
    }
    public function store(Request $request)
    {
        if(!Auth::check()){
            return redirect()->route('login');
        }
        $request->validate([
            'product_id' => 'required',
            'quantity' => 'required',
            'invoice' => 'required',
        ]);
        $newOrder = Order::create([
            'invoice' => $request->invoice,

```

```

        'quantity' => $request->quantity,
        'status' => 'unpaid',
        'user_id' => auth()->id()
    ]);

    $newOrder->products()->attach($request->product_id);

    return redirect()->route('order.index')->with('status', 'Order
created successfully');
}

public function show(Order $order) : View
{
    if(!Auth::check()){
        return view('auth.login');
    }
    return view('order.show',[
        'order' => $order,
        'title' => 'Order Details'
    ]);
}

public function edit(Order $order) : View
{
    if(!Auth::check()){
        return view('auth.login');
    }
    $products = Product::all();
    return view('order.edit', [
        'order' => $order,
        'title' => 'Edit Order',
        'products' => $products
    ]);
}

```



```

    }

    public function update(Request $request, Order $order)
    {
        if(!Auth::check()){
            return redirect()->route('login');
        }
        $request->validate([
            'product_id' => 'required',
            'quantity' => 'required',
            'invoice' => 'required'
        ]);
        $order->update([
            'invoice' => $request->invoice,
            'quantity' => $request->quantity,
            'status' => 'unpaid',
            'user_id' => auth()->id()
        ]);
        $order->products()->sync($request->product_id);
        return redirect()->route('order.index')->with('status', 'Order
updated successfully');
    }

    public function destroy(Order $order)
    {
        if(!Auth::check()){
            return redirect()->route('login');
        }
        $order->delete();
        return redirect()->route('order.index')->with('status', 'Order
deleted successfully');
    }
}

```

Product :
<?php

```
namespace App\Http\Controllers;
```

```
use App\Models\Ability;  
use Illuminate\Http\RedirectResponse;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Auth;  
Use Illuminate\View\View;  
use App\Models\Product;
```

```
class ProductController extends Controller
```

```
{  
    public function index(): View  
    {  
        if(!Auth::check()){  
            return view('auth.login');  
        }  
        $products = Product::paginate(5);  
        return view('product.index', [  
            'products' => $products,  
            'title' => 'Product List'  
        ]);  
    }  
}
```

```
    public function show(Product $product): View  
    {  
        if(!Auth::check()){  
            return view('auth.login');  
        }  
        return view('product.show', [  
            'product' => $product,  
            'title' => 'Product Details'    }  
}
```

```

        'product' => $product,
        'title' => 'Product Details'
    ]));
}

```

```

public function create(): View
{
    if(!Auth::check()){
        return view('auth.login');
    }
    return view('product.create',
        [
            'title' => 'Create Product'
        ]
    );
}

```

```

public function store(Request $request): RedirectResponse
{
    if(!Auth::check()){
        return redirect()->route('login');
    }
    $request->validate([
        'name' => 'required',
        'description' => 'required',
        'price' => 'required',
        'stock' => 'required',
        'image' => 'required|image'
    ]);

    $image = $request->file('image');
    $filename = $image->getClientOriginalName();
}

```

```

$image->move(public_path('images'), $filename);

Product::create([
    'name' => $request->name,
    'description' => $request->description,
    'price' => $request->price,
    'stock' => $request->stock,
    'image' => $filename
]);

return redirect()->route('product.index')->with('status', 'Product
created successfully');
}

public function edit(Product $product): View
{
    if(!Auth::check()){
        return view('auth.login');
    }
    return view('product.edit', [
        'product' => $product,
        'title' => 'Edit Product'
    ]);
}

public function update(Request $request, Product $product):
RedirectResponse
{
    if(!Auth::check()){
        return redirect()->route('login');
    }
    if ($request->hasFile('image')) {
        $oldImage = public_path('images/' . $product->image);

```

```

        if (file_exists($oldImage)) {
            unlink($oldImage);
        }
        $image = $request->file('image');
        $filename = $image->getClientOriginalName();
        $image->move(public_path('images'), $filename);
    } else {
        $filename = $product->image;
    }
    $product->update([
        'name' => $request->name,
        'description' => $request->description,
        'price' => $request->price,
        'stock' => $request->stock,
        'image' => $filename
    ]);

    return redirect()->route('product.show',$product->id)-
>with('status', 'Product updated successfully');
}

public function destroy(Product $product): RedirectResponse
{
    if(!Auth::check()){
        return redirect()->route('login');
    }
    $image = public_path('images/' . $product->image);
    if (file_exists($image)) {
        unlink($image);
    }
    $product->delete();
}

```

```
        return redirect()->route('product.index')->with('status', 'Product
deleted successfully');
    }
}
```

User:

<?php

```
namespace App\Http\Controllers;
```

```
use App\Models\User;
```

```
use Illuminate\Http\RedirectResponse;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\View\View;
```

```
class UserController extends Controller
```

```
{
```

```
    public function index(): View
```

```
    {
```

```
        return view('user.index', [
            'users' => User::all(),
            'title' => 'User List'
        ]);
    }
```

```
    public function show(User $user): View
```

```
    {
```

```
        return view('user.show', [
            'user' => $user,
            'title' => 'User Details'
        ]);
    }
```

```

public function create(): View
{
    return view('user.create');
}

public function store(Request $request): RedirectResponse
{
    $request->validate([
        'name' => 'required',
        'email' => 'required',
        'password' => 'required',
    ]);

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

    return redirect()->route('users.index')->with('success', 'User
created successfully');
}

public function edit(User $user): View
{
    return view('user.edit', [
        'user' => $user,
        'title' => 'Edit User'
    ]);
}

public function update(Request $request, User $user): RedirectResponse
{
    $user->update($request->all());
}

```

```

        return redirect()->route('users.index')->with('success', 'User
updated successfully');
    }

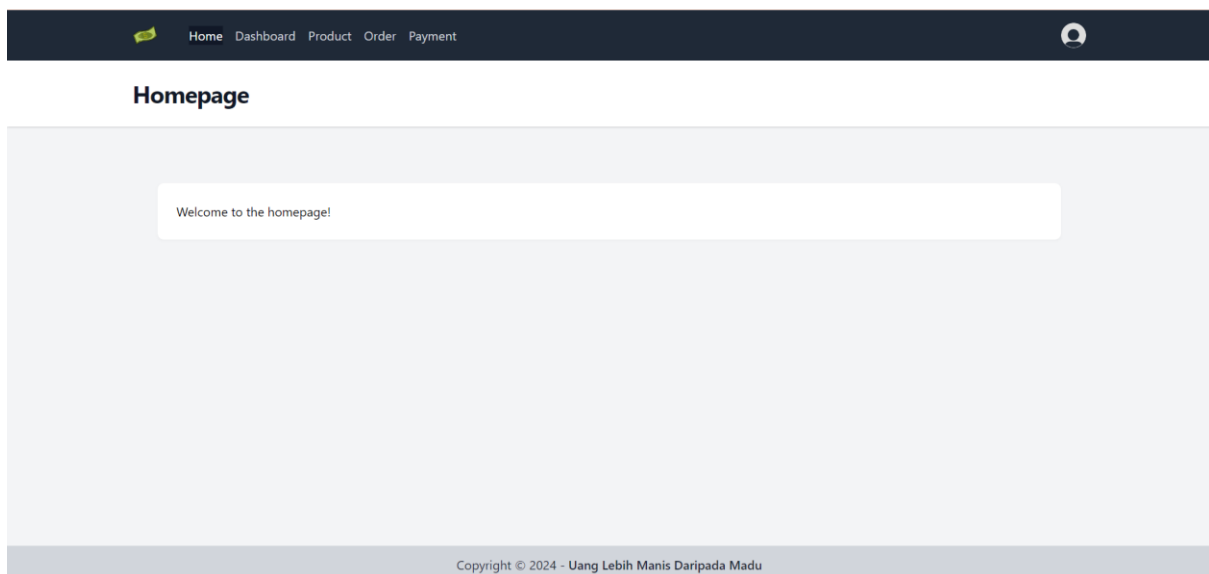
    public function destroy(User $user): RedirectResponse
    {
        $user->delete();

        return redirect()->route('users.index')->with('success', 'User
deleted successfully');
    }
}


```

Tampilan :

Homepage



Login :




Email

Password

☐ Remember me

[Forgot your password?](#) **LOG IN**

Register :



Name

Email


Password


Confirm Password


[Already registered?](#) **REGISTER**


Halaman Produk


Index :


 Home

 Dashboard

 Product

 Order

 Payment



Product

Add new product



Krabby Patty Coy

Rp.100000.00

More...

Last updated 1 hour ago



Pikachu Plushie


Rp.50000.00


More...


Last updated 1 hour ago


Copyright © 2024 - Uang Lebih Manis Daripada Madu


Show :


 Home

 Dashboard

 Product

 Order

 Payment



 Krabby Patty Coy

| | |
|-------------|-----------------------------------|
| Name | Krabby Patty Coy |
| Description | A delicious Burger Made From Fish |
| Price | 100000.00 |
| Stock | 4 |
| Action | <div>EditDeleteBack</div> |

Copyright © 2024 - Uang Lebih Manis Daripada Madu

Create :

Create New Product

Name

Image

Choose File

No file chosen

Description

Price

Stock

Create

Copyright © 2024 - Uang Lebih Manis Daripada Madu

Edit :

Edit Product

Name

Krabby Patty Coy

Image

Choose File

No file chosen

Description

A delicious Burger Made From Fish

Price

100000,00


Stock


100000,00

Update

Copyright © 2024 - Uang Lebih Manis Daripada Madu

Order :

HomeDashboardProductOrderPayment




Order


Add new order

| | |
|-------------------|--|
| Order1
unpaid | More...
Last updated 6 hours from now |
| Order2
unpaid | More...
Last updated 27 minutes ago |
| Order3
unpaid | More...
Last updated 22 minutes ago |
| Order 4
unpaid | More...
Last updated 21 minutes ago |

Copyright © 2024 - Uang Lebih Manis Daripada Madu

Show :

HomeDashboardProductOrderPayment





Order1

| | |
|----------|---------------------------------|
| Invoice | Order1 |
| Order By | Zia Khusnul Fauzi Akhmad |
| Quantity | 1 |
| Status | unpaid |
| Products | *Krabby Patty Coy, Quantity : 1 |
| Action | <div>EditBack</div> |

Copyright © 2024 - Uang Lebih Manis Daripada Madu

Payment :

HomeDashboardProductOrderPayment



Payment

Add new payment

Payment 01


12000.00


More...

Last updated 6 hours from now

Copyright © 2024 - Uang Lebih Manis Daripada Madu

Show :

HomeDashboardProductOrderPayment



Payment 01

| | |
|----------------|---------------------|
| Payment Number | Payment 01 |
| Order Invoice | Order1 |
| Amount | Rp.12000.00 |
| Action | <div>EditBack</div> |

Copyright © 2024 - Uang Lebih Manis Daripada Madu