

Project Case

COMP6681001

Web Programming

Odd Semester Year 2022/2023



Online Shop Startup With Information Technology Based on Website Applications Using Laravel 8

Group Forming : Group 17
Group Member : 2440018854 - Gerry William Nanlohy
 2440010063 - Maria Jessica Sutjiadi
Class : BO01
Subject : Web Programming - COMP6681001

A. Introduction

Barbatos Shop

Barbatos Shop is a startup online shop in Japan made by a computer science student. For its business, Barbatos Shop's CEO wants to create a website for their shop. In that website, people can find information about the product that Barbatos sells right now and order it. As a web developer, we create a website using Laravel 8 for this online shop. There are three types of user's role in this website are Admin, Customer, and Guest (non-logged-in user). The pages in this website are as follows in B section.

B. Website Page

1. Login Page
2. Register Page
3. Home Page
4. View Product by Category Page
5. Product Detail Page
6. Search Product Page
7. Manage Product Page
8. Add Product Page
9. Update Product Page
10. Profile Page
11. Cart Page
12. History Page

C. Website Database

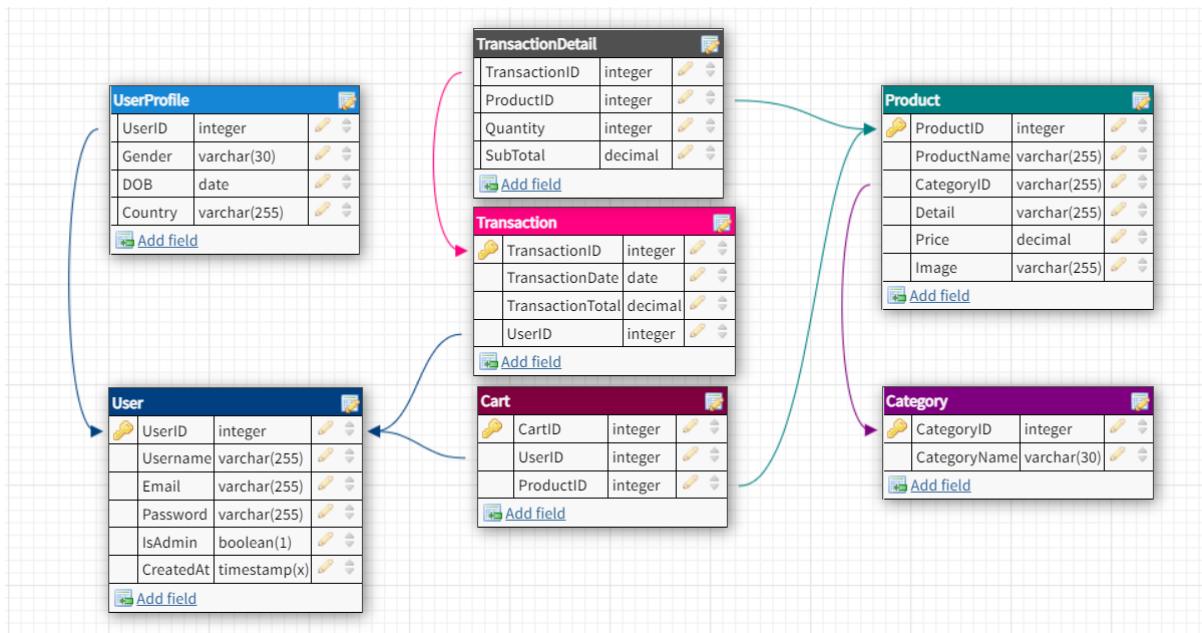
1. Admin
 - Email
 - Password
 - Name
 - Category
 - Genre
 - Detail
 - Price
 - Photo
2. Customer
 - Name
 - Email
 - Password
 - Gender
 - Date of Birth
 - Country
3. Guest (non-logged-in user)
 - Name
 - Email
 - Password
 - Gender
 - Date of Birth
 - Country

D. Website UML Diagram

UML (Unified Modeling Language) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships among objects. The following is UML Diagram of the Barbatos Shop Website:

1. Database Schema

Database schema defines how data is organized within a relational database; this is inclusive of logical constraints such as, table names, fields, data types, and the relationships between these entities. The following is a database schema of the Barbatos Shop Website:



2. Use Case Diagram

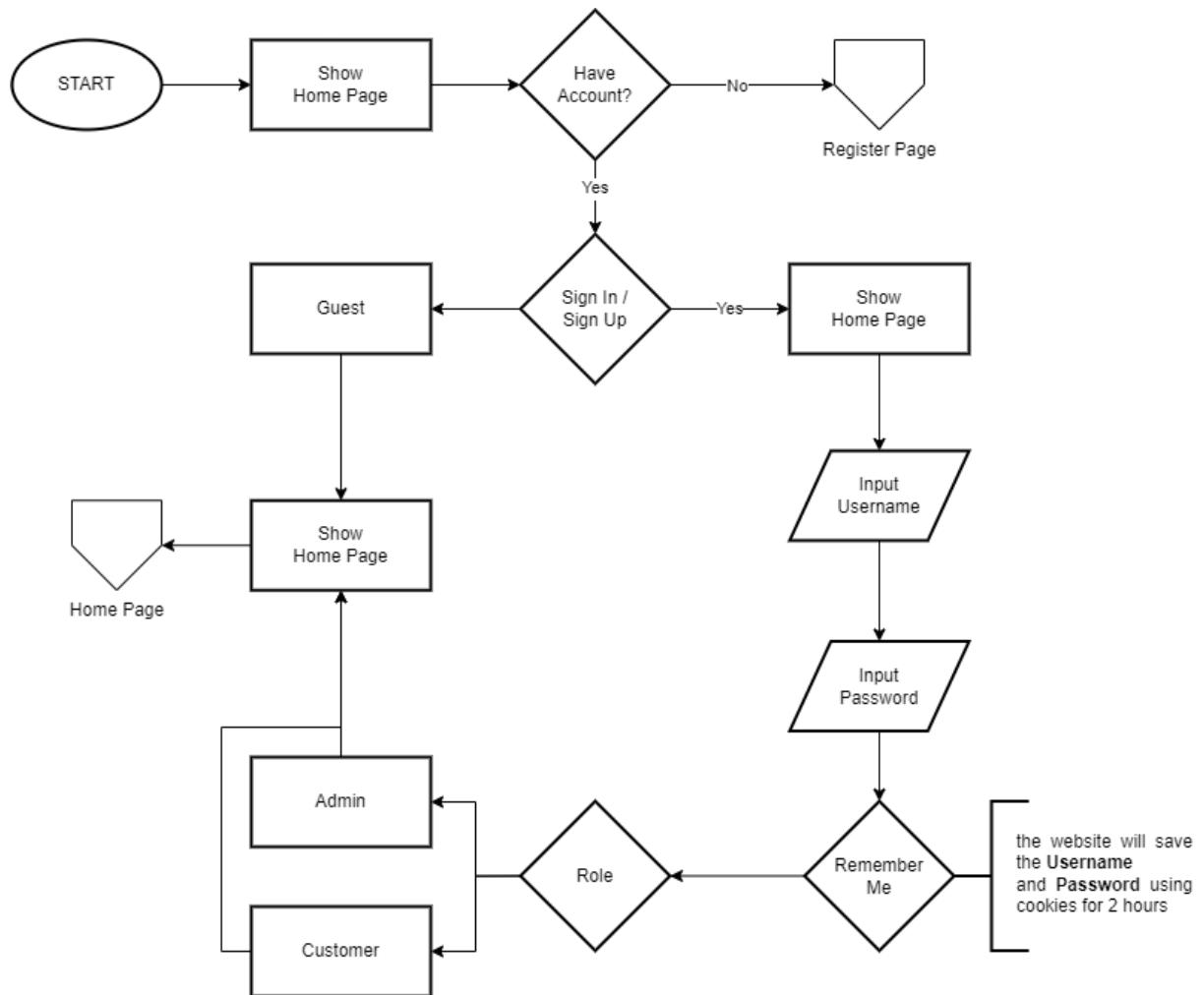
Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. The following is a use case diagram of the Barbato Shop Website:



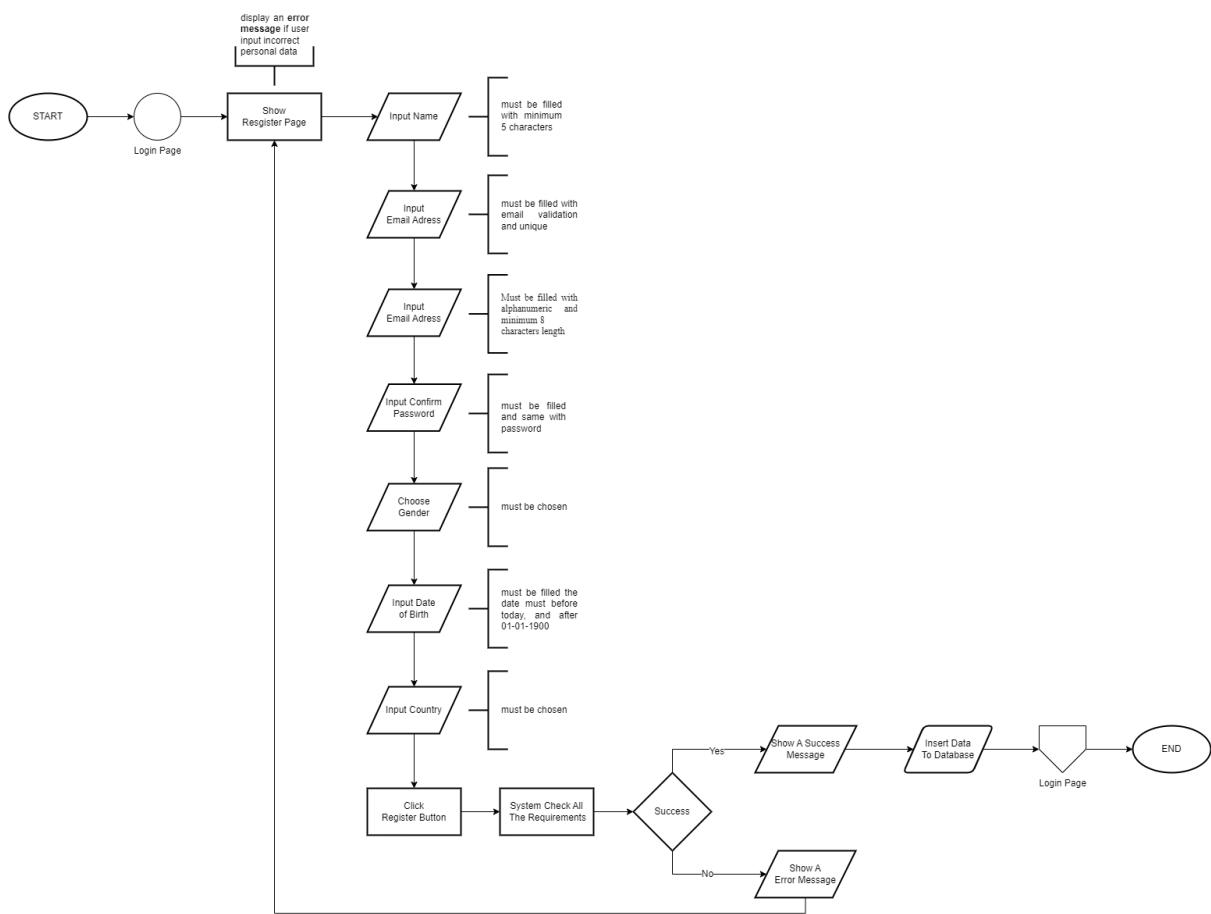
3. Website Flowchart

Flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan. The following is a flowchart of each page on the Barbatos Shop Website:

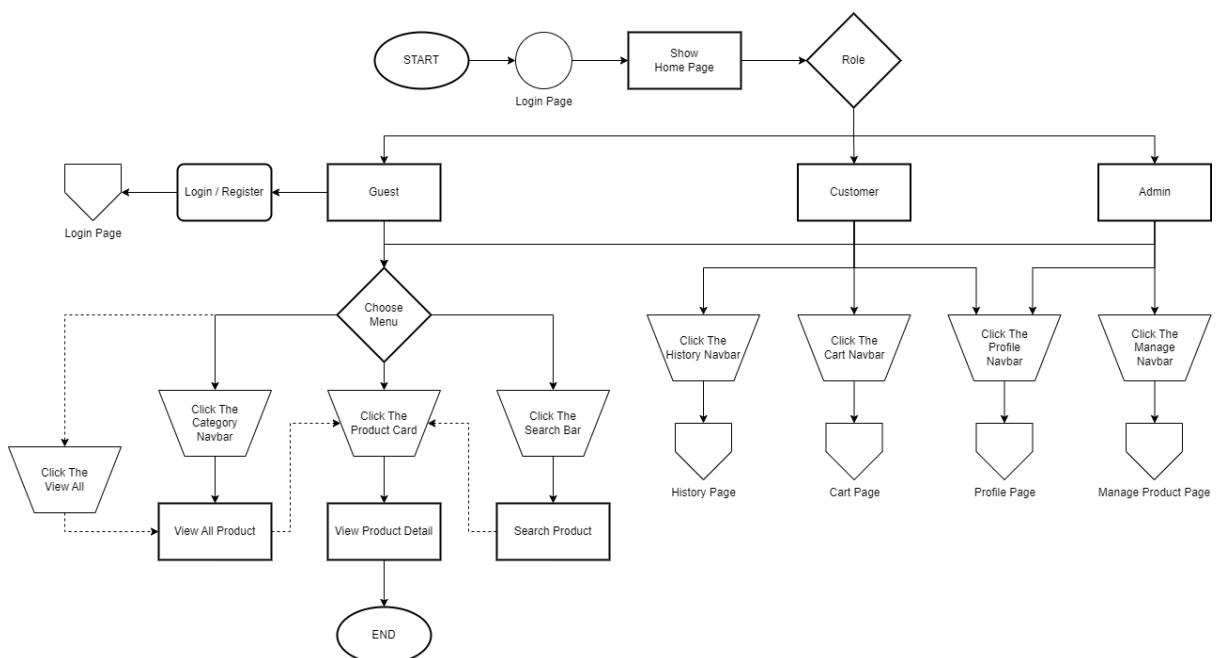
3.1 Login Page



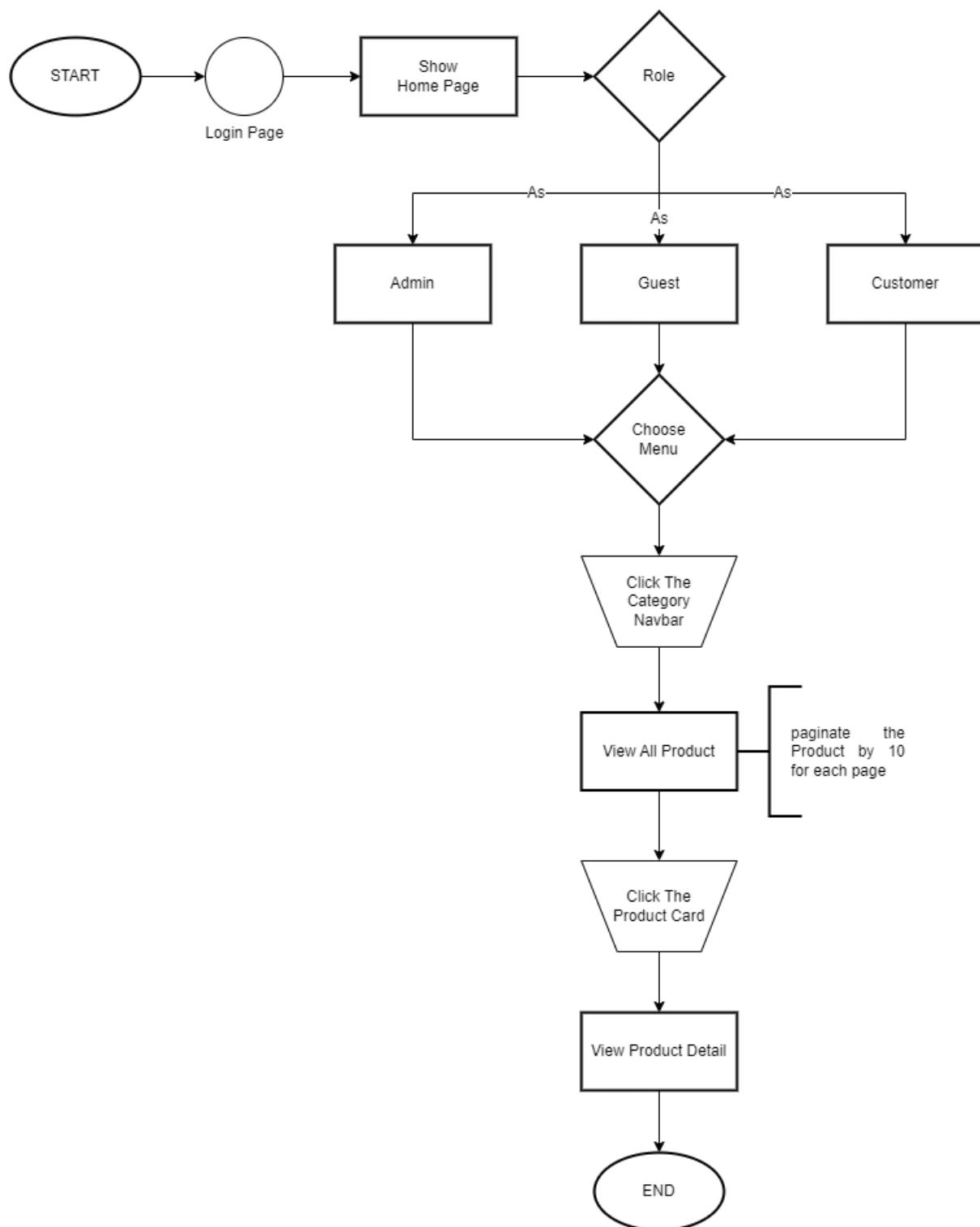
3.2 Register Page



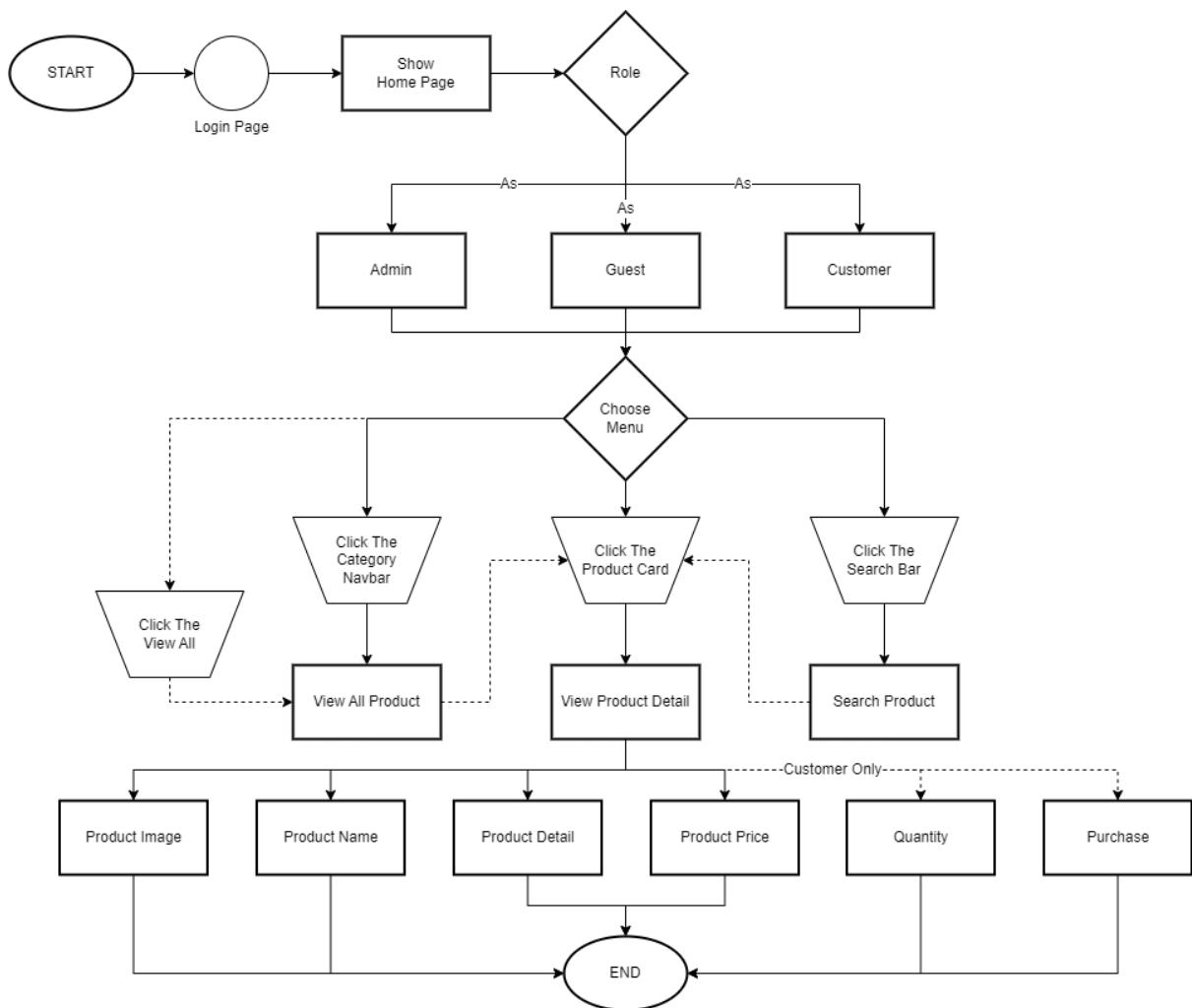
3.3 Home Page



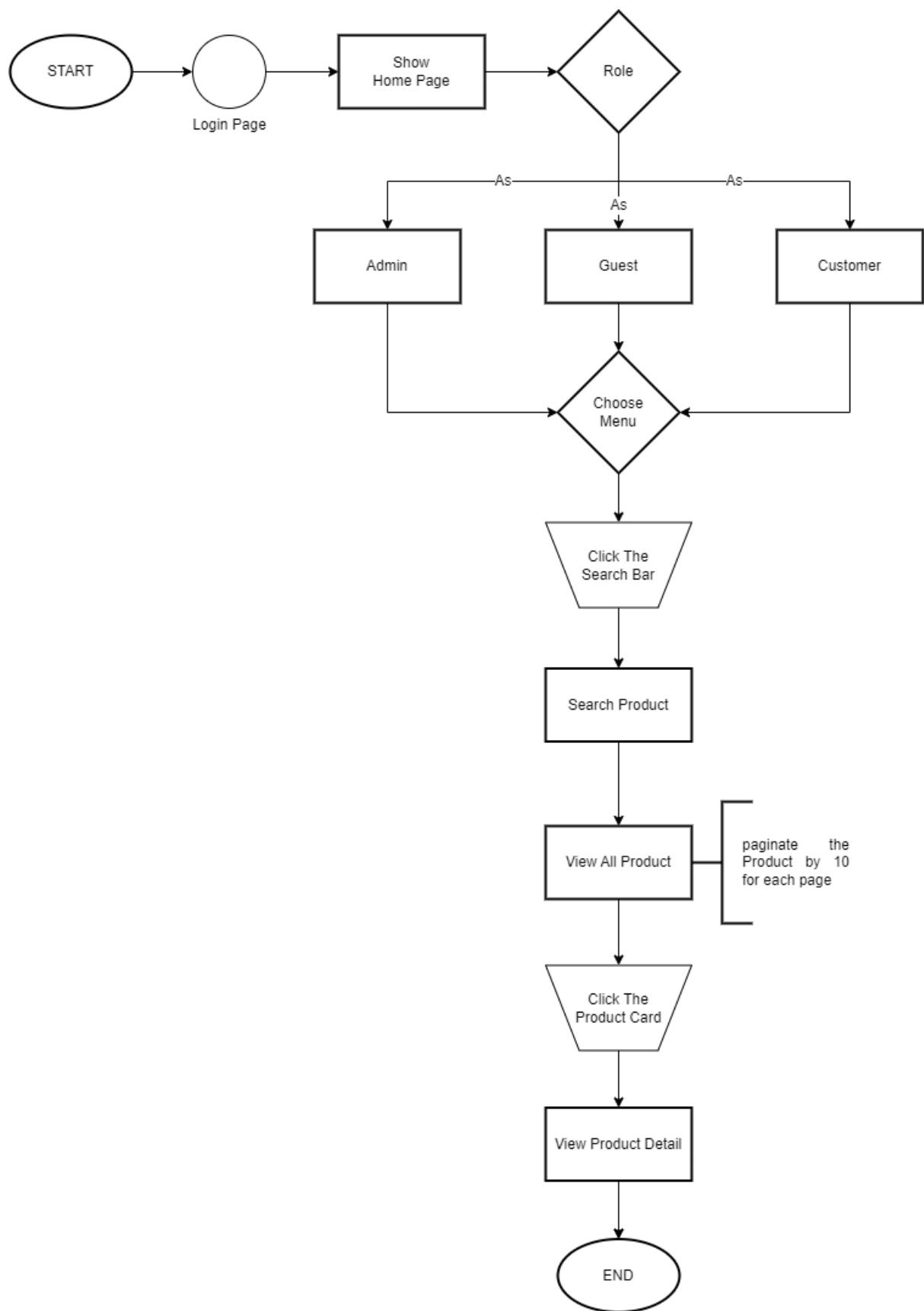
3.4 View Product by Category Page



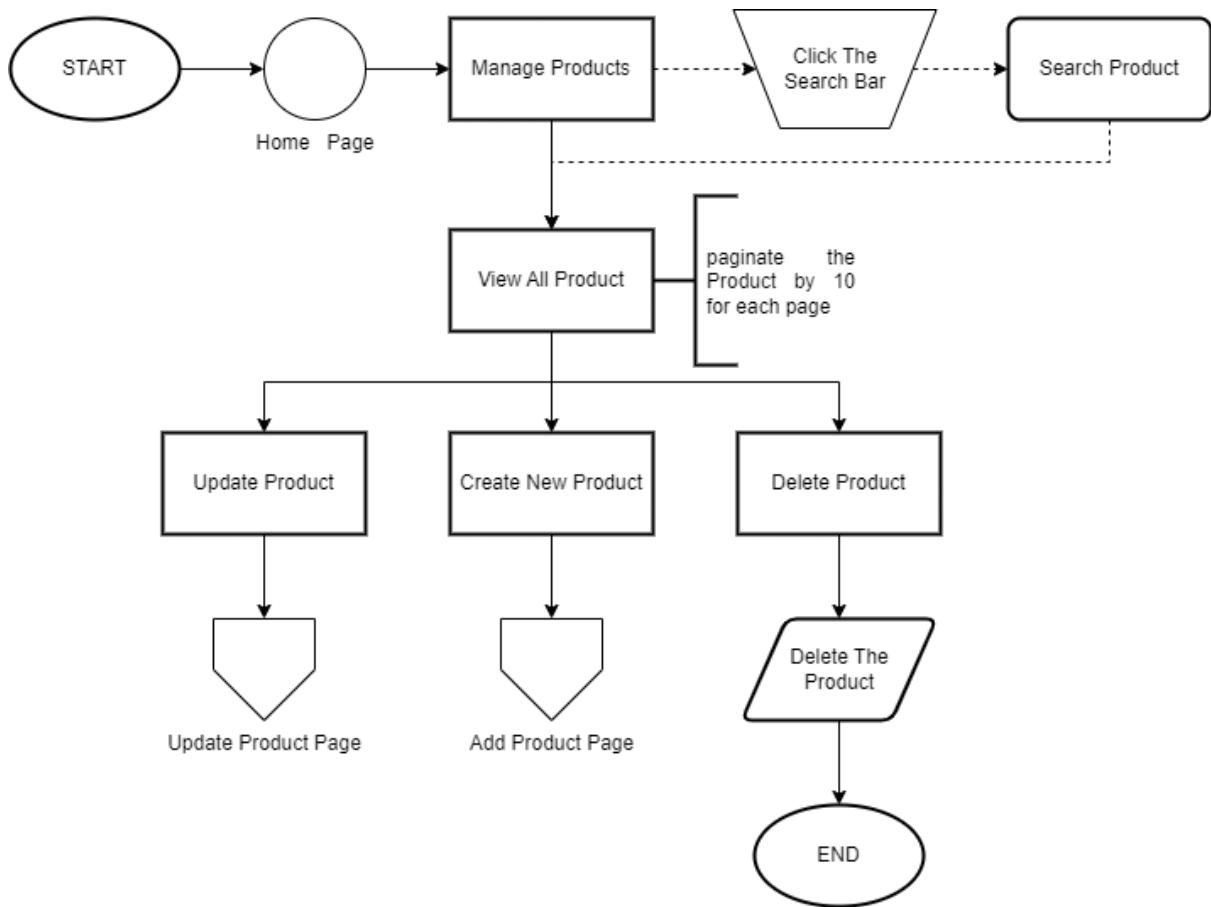
3.5 Product Detail Page



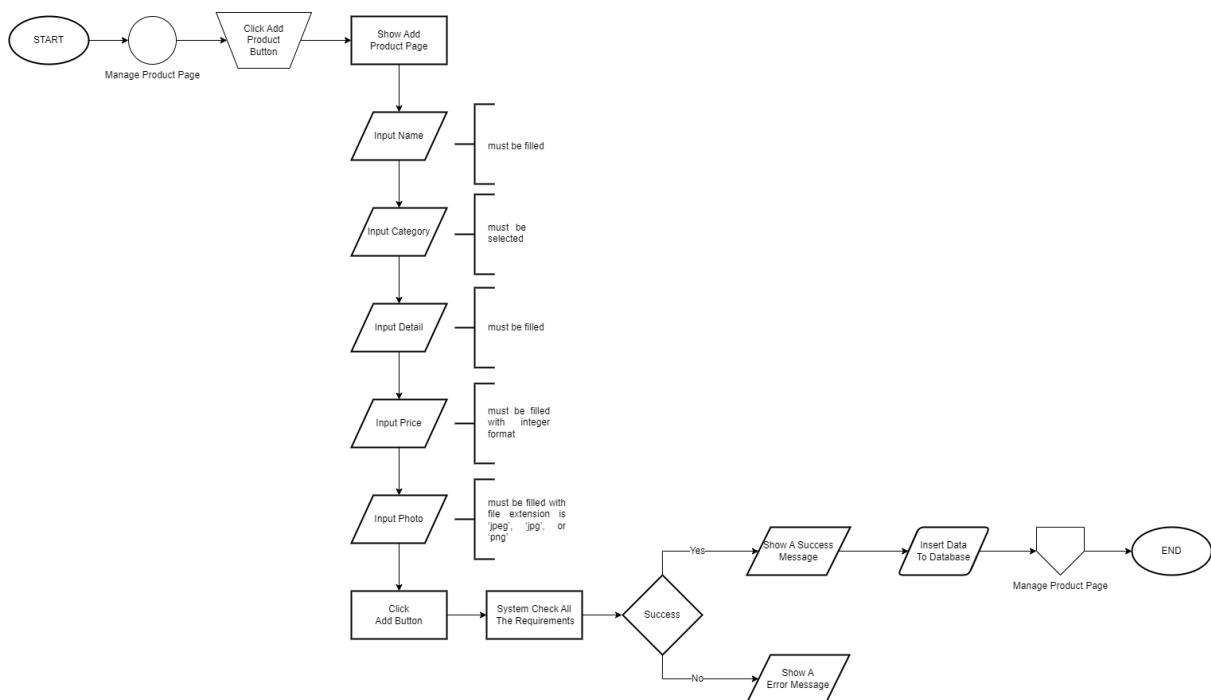
3.6 Search Product Page



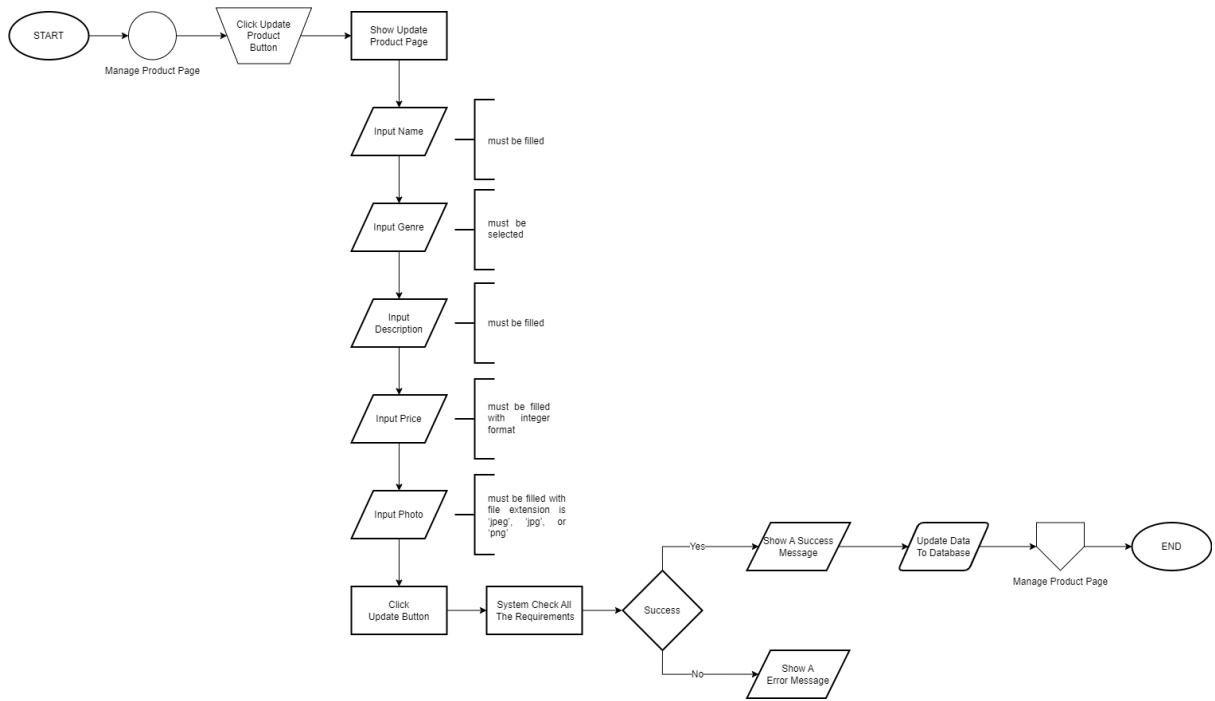
3.7 Manage Product Page



3.8 Add Product Page



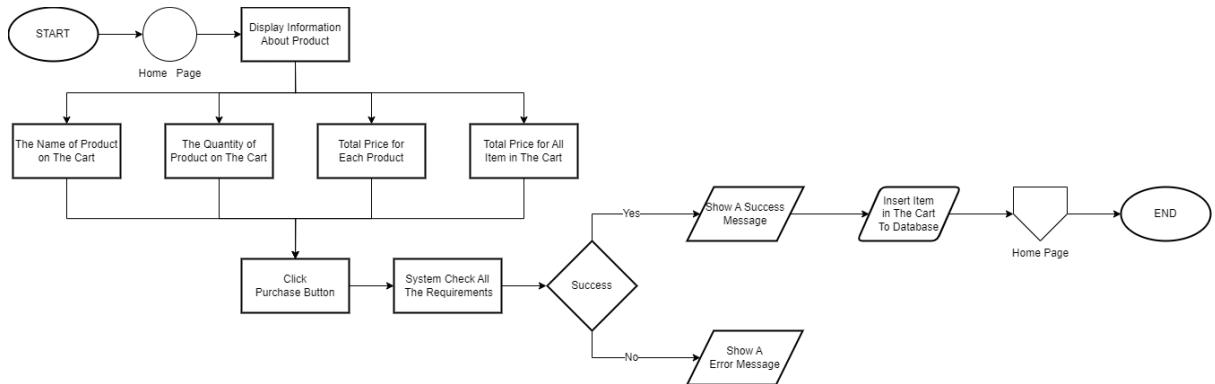
3.9 Update Product Page



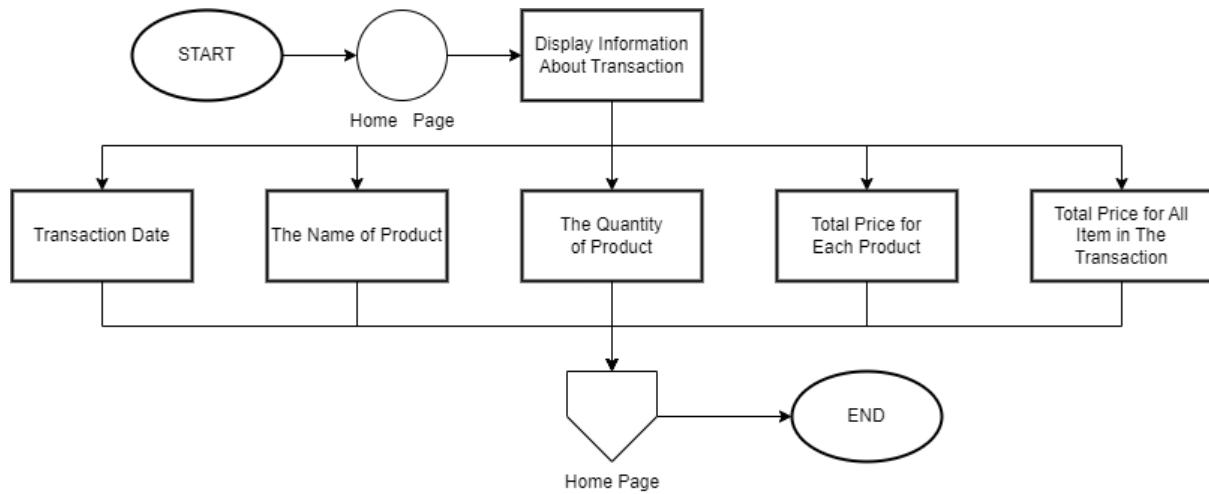
3.10 Profile Page



3.11 Cart Page



3.12 History Page



E. Website Result

1. Website Frontend

The following is a frontend design on the Barbatus Shop Website:

1.1 Login Page

Login Page: Empty State

A screenshot of a web browser showing the login page for the Barbatus Shop. The URL in the address bar is 127.0.0.1:8000/login. The page has a pink header bar with the shop's name 'Barbatus' on the left and navigation links 'Category', 'Login', and 'Register' on the right. Below the header is a white login form with a title 'Login'. The form contains fields for 'Email Address' (with a placeholder 'Email') and 'Password' (with a placeholder 'Password'). There is also a 'Remember Me' checkbox and a red 'Login' button at the bottom. The rest of the page is blank white space.

Login Page: Wrong Password

Login

Email Address !

These credentials do not match our records.

Password

Remember Me

Login

1.2 Register Page

Register Page: Empty State

← → ⌂ ⌂ 127.0.0.1:8000/register

Gmail YouTube Maps News Translate BINUSMAYA Binusmaya Practicum Upload Recording S... Learn C | SoloLearn Linktree Admin LKP Booster English... GWENTzy (Gerry Wi...

Barbatos Category ▾ Login Register

Register

Name

Email Address

Password

Confirm Password

Gender Male Female

Date of Birth

Country

Register

Copyright 2020 - ChikaClau

Register Page: Wrong Password Format

Register

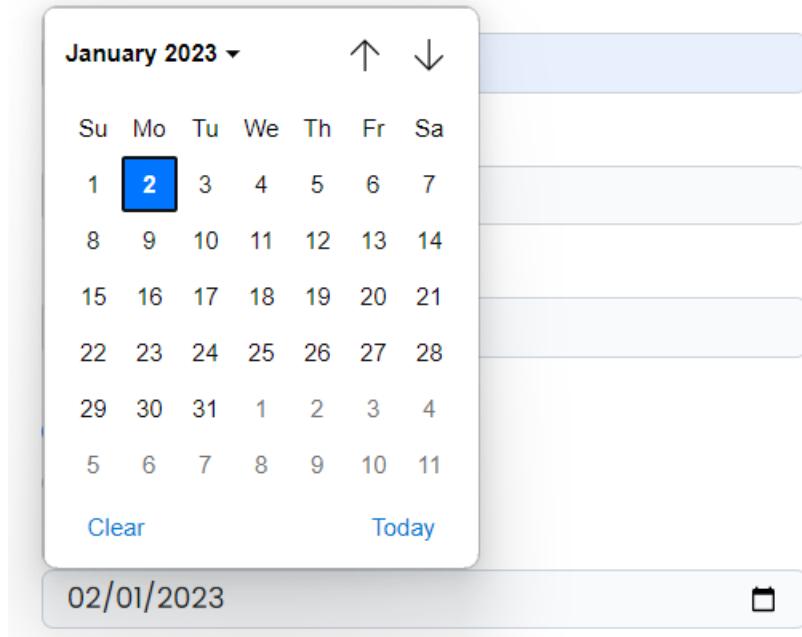
Name	Gerry William Nanlohy
Email Address	nanlohygerry@gmail.com
Password	<input type="password"/> ⓘ The password must only contain letters and numbers.
Confirm Password	<input type="password"/>
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Date of Birth	02/01/2023 <input type="button" value="Change"/> ⓘ The dob must be a date before today.
Country	Indonesia
Register	

Register Page: Wrong Password

Register

Name	Gerry William Nanlohy
Email Address	nanlohygerry@gmail.com
Password	<input type="password"/> ⓘ The password confirmation does not match.
Confirm Password	<input type="password"/>
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Date of Birth	02/01/2023 <input type="button" value="Change"/> ⓘ The dob must be a date before today.
Country	Indonesia
Register	

Register Page: Date Section



Register Page: Wrong Date Requirement

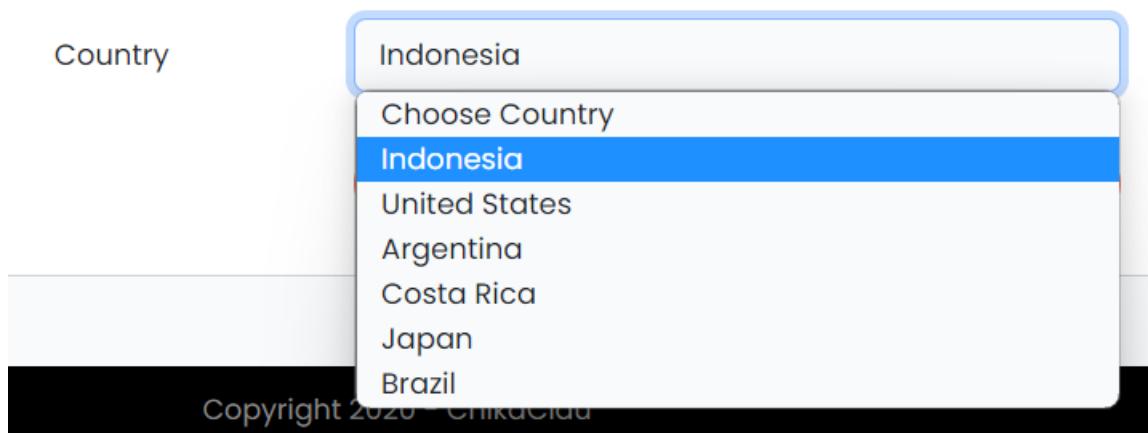
Date of Birth !

The dob must be a date before today.

Country

Register

Register Page: Country Section



Register Page: Progress State

Screenshot of the registration form on the progress state page.

The URL in the address bar is 127.0.0.1:8000/register.

The page title is "Barbatos".

The top navigation bar includes links for Gmail, YouTube, Maps, News, Translate, BINUSMAYA, Binusmaya Practicum, Upload Recording S..., Learn C | SoloLearn, Linktree Admin, LKP Booster English..., and GWENTzy (Gerry Wi...).

The user is logged in as "Category" and "Barbatos".

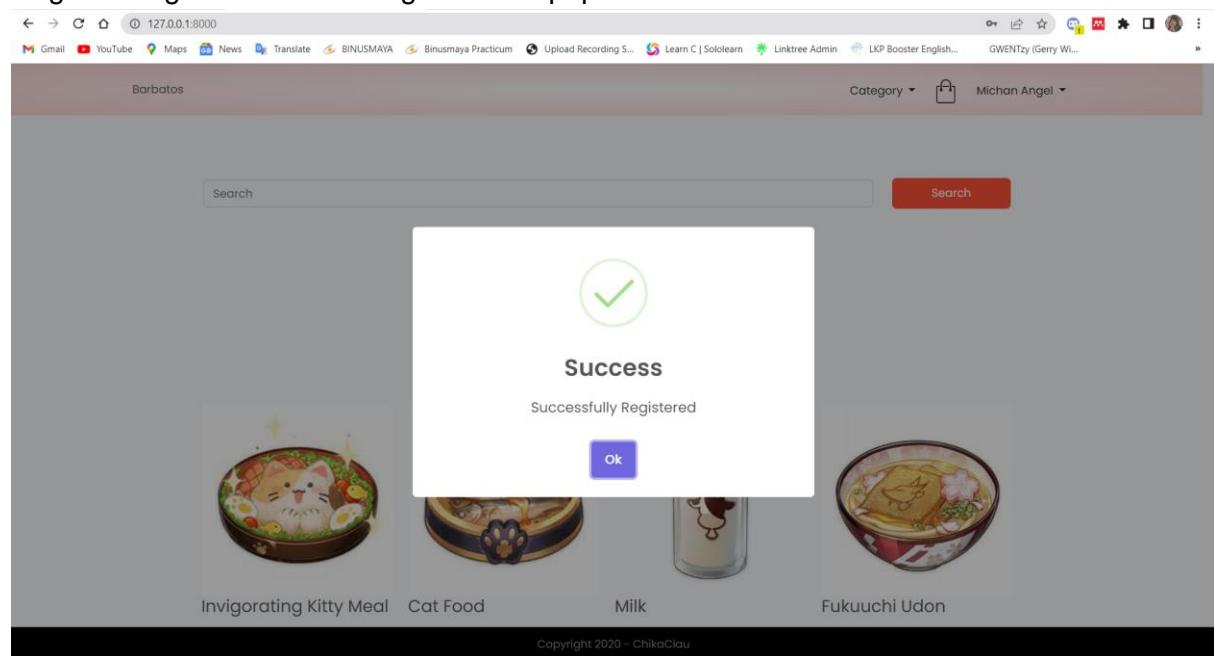
The registration form fields are:

- Name: Gerry William Nanlohy
- Email Address: nanlohygerry@gmail.com
- Password: (redacted)
- Confirm Password: (redacted)
- Gender: Male (radio button selected)
- Date of Birth: 01/07/2002
- Country: Indonesia

A red "Register" button is at the bottom of the form.

Copyright 2020 - ChikaClau is displayed at the bottom of the page.

Register Page: Successful Registered Popup



1.3 Home Page

Home Page: Guest State

127.0.0.1:8000

Gmail YouTube Maps News Translate BINUSMAYA Binusmaya Practicum Upload Recording S... Learn C | SoloLearn Linktree Admin LKP Booster English... GWENTzy (Gerry WL...

Barbatos Category Login Register

Search Search

Food

— View More

Invigorating Kitty Meal Cat Food Milk Fukuuchi Udon

Rp 15.000 Rp 20.000 Rp 10.000 Rp 20.000

Copyright 2020 - ChikaClau

Home Page: Admin State

127.0.0.1:8000

Gmail YouTube Maps News Translate BINUSMAYA Binusmaya Practicum Upload Recording S... Learn C | SoloLearn Linktree Admin LKP Booster English... GWENTzy (Gerry WL...

Barbatos Category Manage Product Admin2

Search Search

Food

— View More

Invigorating Kitty Meal Cat Food Milk Fukuuchi Udon

Rp 15.000 Rp 20.000 Rp 10.000 Rp 20.000

Copyright 2020 - ChikaClau

Home Page: Customer State

A screenshot of a web browser showing a customer state home page. The address bar displays "127.0.0.1:8000". The top navigation bar includes links for Gmail, YouTube, Maps, News, Translate, BINUSMAYA, BINUSMAYA Practicum, Upload Recording S..., Learn C | SoloLearn, Linktree Admin, LKP Booster English..., and GWENTzy (Gerry Wi...). The user is logged in as "Barbatos" and "Gerry William Nanlohy". A search bar with placeholder "Search" and a red "Search" button are visible. Below the search bar, the word "Food" is centered, with a "View More" link underneath. Four items are displayed: "Invigorating Kitty Meal" (an illustration of a cat in a ramen bowl), "Cat Food" (an illustration of a cat head on a bowl), "Milk" (an illustration of a bottle with a cow logo), and "Fukuuchi Udon" (an illustration of a bowl of udon soup). The footer contains the copyright notice "Copyright 2020 - ChikaClau".

Home Page: Admin Dropdown

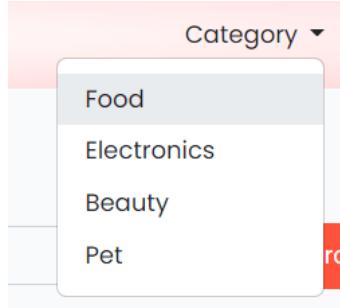
A screenshot of a web browser showing an admin dropdown menu. The top navigation bar includes "Category", "Manage Product", and "Admin2". The "Admin2" menu is open, showing options: "Profile" and "Logout".

Home Page: Customer Dropdown

A screenshot of a web browser showing a customer dropdown menu. The top navigation bar includes "Gerry William Nanlohy". The dropdown menu is open, showing options: "Profile", "Transaction", and "Logout".

1.4 View Product by Category Page

View Product by Category Page: Dropdown



View Product by Category Page: Food Section

A screenshot of a web browser showing the "Food" section of a product category page. The URL is 127.0.0.1:8000. The page has a search bar at the top. Below it, the word "Food" is displayed with a "View More" link underneath. There are four product cards:

- Invigorating Kitty Meal** (Rp. 15,000) - An illustration of a cat inside a bowl of food.
- Cat Food** (Rp. 20,000) - An illustration of a cat inside a bowl of food.
- Milk** (Rp. 10,000) - An illustration of a small bottle with a cow on it.
- Fukuuchi Udon** (Rp. 30,000) - An illustration of a bowl of udon soup.

Copyright 2020 - ChikaClau

View Product by Category Page: Electronics Section

A screenshot of a web browser showing the "Electronics" section of a product category page. The URL is 127.0.0.1:8000. The page has a search bar at the top. Below it, the word "Electronics" is displayed with a "View More" link underneath. There are two product cards:

- Iphone 14 Pro Max** (Rp. 34,000,000) - An illustration of an iPhone 14 Pro Max.
- Asus TUF Gaming F15** (Rp. 16,500,000) - An illustration of an Asus TUF Gaming F15 laptop.

Copyright 2020 - ChikaClau

View Product by Category Page: Beauty Section

127.0.0.1:8000

Gmail YouTube Maps News Translate BINUSMAYA Binusmaya Practicum Upload Recording S... Learn C | SoloLearn Linktree Admin LKP Booster English... GWENTzy (Gerry Wi...

Rp. 34,000,000 Asus TUF Gaming F15 Rp. 16,500,000

Beauty

— View More

CosRX Acne Hero Kit Rp. 1,100,000

Hada Labo Gokujyun Ultimate Moisturizing Cleansing Oil Rp. 98,000

Hada Labo Gokujyun Alpha Ultimate Anti Aging Essence Rp. 115,000

Copyright 2020 - ChikaCiau

Register Page: Pet Section

127.0.0.1:8000

Gmail YouTube Maps News Translate BINUSMAYA Binusmaya Practicum Upload Recording S... Learn C | SoloLearn Linktree Admin LKP Booster English... GWENTzy (Gerry Wi...

Rp. 98,000 Rp. 115,000

Pet

— View More

Canned Pedigree Dog Food Rp. 66,000

Moochie Wet Cat Food Grain Free Rp. 45,000

Copyright 2020 - ChikaCiau

1.5 Product Detail Page

The screenshot shows a product detail page on a web browser. The URL in the address bar is 127.0.0.1:8000/product/detail/20. The page has a pink header bar with the text "Barbatos". On the right side of the header are links for "Category", "Login", and "Register". Below the header, the category "Beauty" is displayed. The main content features a product image of a white bottle of Hada Labo Gokujyun Ultimate Moisturizing Cleansing Oil. To the right of the image, the product name is written in large, bold, black font: "Hada Labo Gokujyun Ultimate Moisturizing Cleansing Oil". Below the name is the price "Rp. 98,000". Underneath the price, the section "PRODUCT DETAILS" is heading. A detailed description of the product follows: "Hada Labo Gokujyun Ultimate Moisturizing Cleansing Oil Mengandung virgin coconut oil dan improved hyaluronic acid. Tidak mengandung bahan pewarna, pewangi dan mineral oil sehingga lembut di kulit. Untuk membersihkan makeup, termasuk water proof mascara dan eyeliner." At the bottom of the page, there is a copyright notice: "Copyright 2020 - ChikaClau".

1.6 Search Product Page

The screenshot shows a search product page on a web browser. The URL in the address bar is 127.0.0.1:8000/product?search=Cat. The page has a pink header bar with the text "Cat". On the right side of the header are links for "Add Product" and a plus sign. Below the header, there are two product cards. The first card, titled "Cat Food", features an image of a bowl of cat food. The second card, titled "Moochie Wet Cat Food Grain Free", features an image of a bag of cat food. Both cards have a delete icon and an edit icon at the bottom right. At the bottom of the page, there is a copyright notice: "Copyright 2020 - ChikaClau".

1.7 Manage Product Page

Screenshot of a web browser showing the 'Products' page. The URL is 127.0.0.1:8000/product?page=2. The page title is 'Barbatos'. The main heading is 'Products'. There is a search bar with placeholder 'Search' and a red 'Search' button. A red button on the right says 'Add Product +'. Below the heading, there is a product card for an 'Asus TUF Gaming F15' laptop, showing its image, name, and two red circular buttons for delete and edit.

Screenshot of a web browser showing the 'Products' page. The URL is 127.0.0.1:8000/product?page=2. The page title is 'Barbatos'. The main heading is 'Products'. There are two product cards: one for 'Hada Labo Gokujyun Alpha Ultimate Anti Aging Essence' showing two red bottles, and another for 'Canned Pedigree Dog Food' showing a can of dog food. Both cards have names, descriptions, and red circular buttons for delete and edit. At the bottom, there are navigation buttons for pages 1, 2, 3, and 4, with page 2 highlighted in blue. A copyright notice 'Copyright 2020 - ChikaClau' is at the bottom.

Manage Product Page: Successful Deleted Popup

A screenshot of a web browser displaying a product management interface. The URL is 127.0.0.1:8000/product. The page title is "Products". In the center, a modal window titled "Success" contains a green checkmark icon and the message "successfully deleted the product". Below the message is a blue "Ok" button. At the top of the page, there is a navigation bar with links like "Category", "Manage Product", and "Admin2". On the left side, there is a search bar and a product card for a smartphone. On the right side, there are buttons for "Add Product" and "Delete" (with a trash icon) and "Edit" (with a pencil icon). The footer of the page includes the text "Copyright 2020 - ChikaClau".

Manage Product Page: View Product by Search

A screenshot of a web browser displaying a product management interface. The URL is 127.0.0.1:8000/?search=Moochie. The page title is "Products". A search bar at the top contains the text "Moochie". To the right of the search bar is a red "Search" button. Below the search bar, a product card for "Moochie Wet Cat Food Grain Free" is displayed. The product image shows a pink can of cat food with a cartoon cat on it. The product name is "Moochie Wet Cat Food Grain Free" and the price is "Rp. 45,000". The footer of the page includes the text "Copyright 2020 - ChikaClau".

1.8 Add Product Page

Add Product Page: Empty State

The screenshot shows a web browser window with the URL 127.0.0.1:8000/product/add. The page title is "Barbatos". The main content is a "New Product" form. It includes fields for Name (empty), Category (labeled "Choose category"), Detail (empty), Price (empty), and Photo (with a "Choose File" button and a "No file chosen" message). The top navigation bar has links like Gmail, YouTube, Maps, News, Translate, BINUSMAYA, Binusmaya Practicum, Upload Recording S..., Learn C | SoloLearn, Linktree Admin, LKP Booster English..., and GWENTzy (Gerry WL...). The bottom of the page has a copyright notice: Copyright 2020 - ChikaCiau.

Add Product Page: No Selected Category in Category Field

Category

Choose category



The category field is required.

Add Product Page: No File in Photo Field

Photo

Choose File

No file chosen



The photo field is required.

Add Product Page: Wrong Type of Photo Format

Photo

Choose File

No file chosen



The photo must be a file of type: jpg, jpeg, png.

Add Product Page: Progress State

Name

Category

Iphone 14 Pro Max from Apple - A16 Chipset

Detail

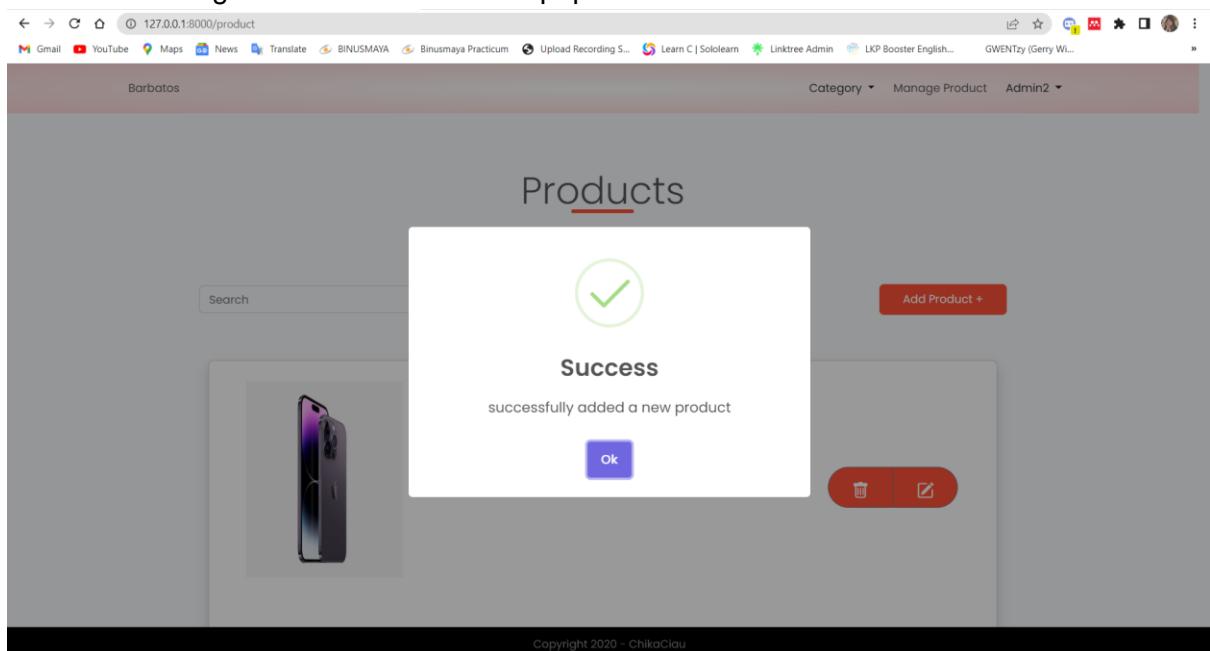
Price

Photo

Add Product

Copyright 2020 - ChikaCiau

Add Product Page: Successful Added Popup

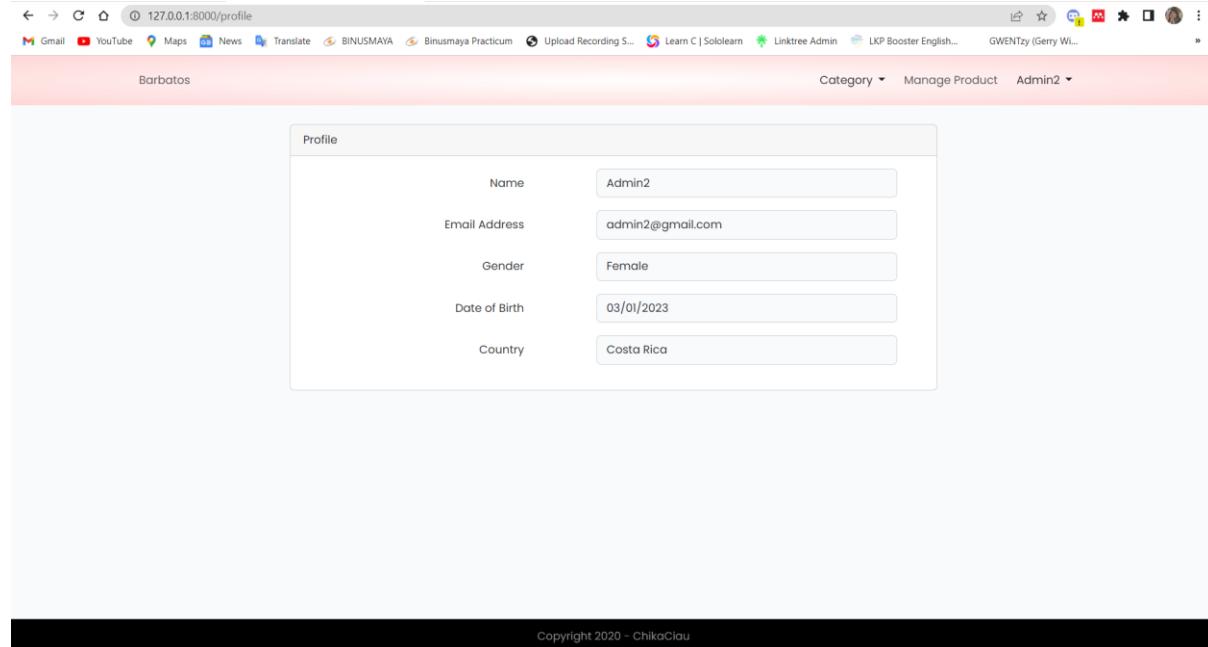


1.9 Update Product Page

Copyright 2020 - ChikaCiau

1.10 Profile Page

Profile Page: Admin State

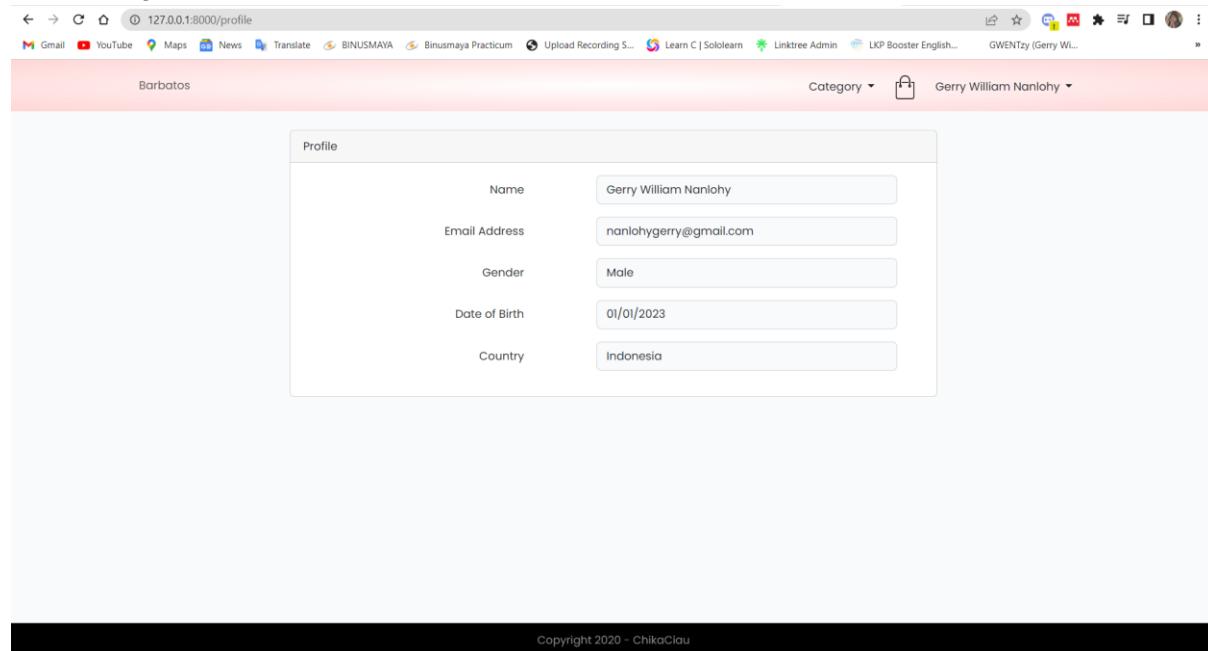


A screenshot of a web browser showing a profile page for an admin user named "Admin2". The page includes fields for Name, Email Address, Gender, Date of Birth, and Country. The browser's address bar shows the URL 127.0.0.1:8000/profile.

Profile	
Name	Admin2
Email Address	admin2@gmail.com
Gender	Female
Date of Birth	03/01/2023
Country	Costa Rica

Copyright 2020 - ChikaClau

Profile Page: Customer State



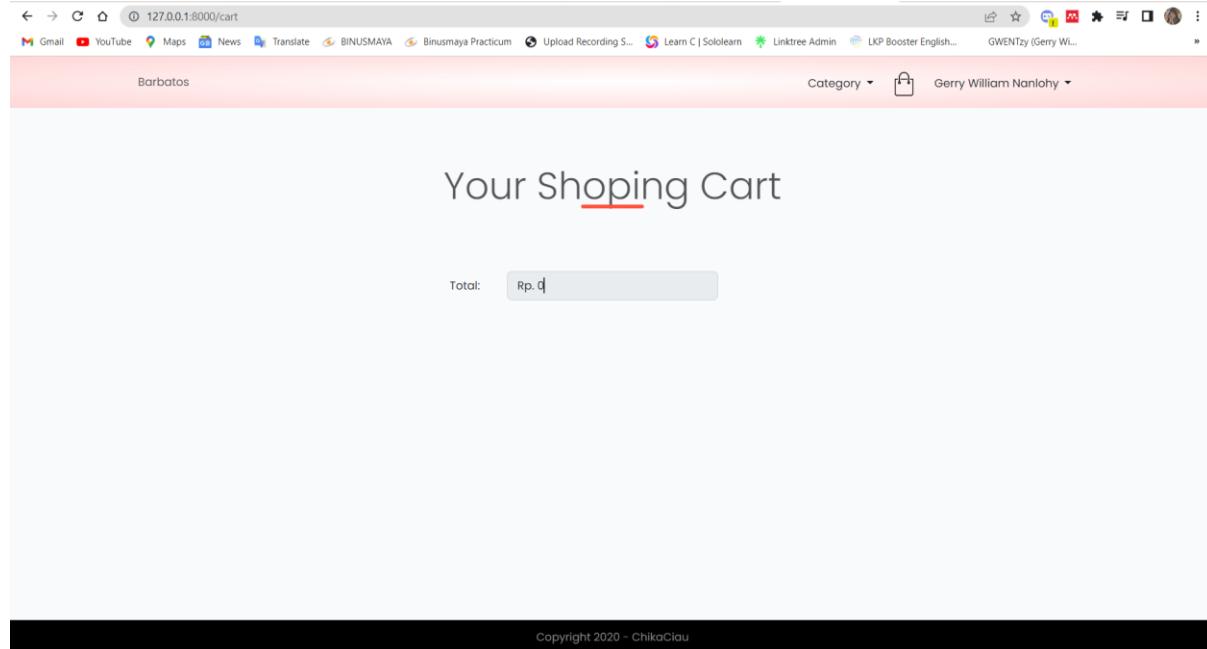
A screenshot of a web browser showing a profile page for a customer user named "Gerry William Nanlohy". The page includes fields for Name, Email Address, Gender, Date of Birth, and Country. The browser's address bar shows the URL 127.0.0.1:8000/profile.

Profile	
Name	Gerry William Nanlohy
Email Address	nanlohygerry@gmail.com
Gender	Male
Date of Birth	01/01/2023
Country	Indonesia

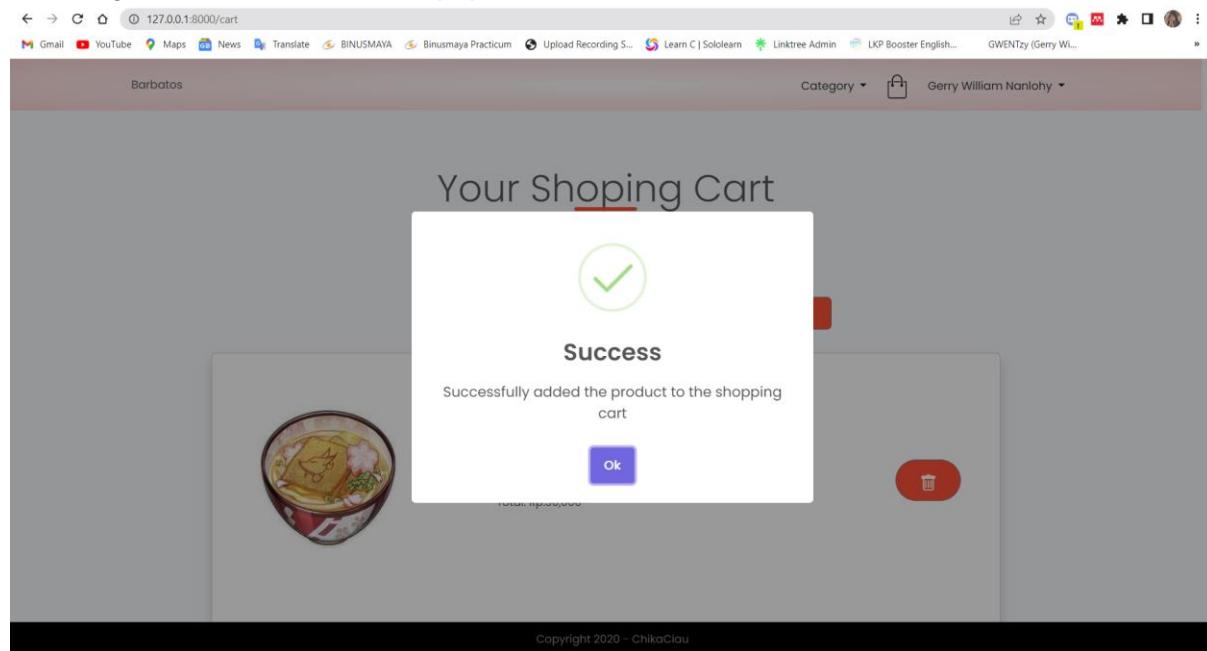
Copyright 2020 - ChikaClau

1.11 Cart Page

Cart Page: Empty State



Cart Page: Successful Added Popup



Cart Page: Progress State

Your Shopping Cart

Total: Rp.15,000 [Purchase](#)

Invigorating Kitty Meal

Qty: 1 Total: Rp.15,000

Copyright 2020 - ChikaClau

1.12 History Page

History Page: Empty State

Your Transaction

Copyright 2020 - ChikaClau

History Page: Successful Purchased Popup

The screenshot shows a web browser window with a pink header bar. The address bar displays the URL `127.0.0.1:8000/transaction`. The main content area is titled "Yor Transaction". A central modal window is open, featuring a green checkmark icon and the word "Success". Below it, the text "Successfully Purchased" is displayed. At the bottom of the modal is a blue "Ok" button. In the background, there are two tables representing transaction histories. The first table shows a single item: "Invigorating Kitty Meal" with a quantity of 1 and a sub total of Rp. 15,000. The second table shows a single item: "Asus TUF Gaming F15" with a quantity of 1 and a sub total of Rp. 16,500,000. Both tables have a "Transaction Date" header at the top. The footer of the page includes the text "Copyright 2020 - ChikaClau".

History Page: Progress State

The screenshot shows a web browser window with a pink header bar. The address bar displays the URL `127.0.0.1:8000/transaction`. The main content area is titled "Yor Transaction". A modal window is partially visible, showing a table with one item: "Invigorating Kitty Meal" with a quantity of 1 and a sub total of Rp. 15,000. Below this, another modal window is visible, showing a table with one item: "Asus TUF Gaming F15" with a quantity of 1 and a sub total of Rp. 16,500,000. Both tables have a "Transaction Date" header at the top. The footer of the page includes the text "Copyright 2020 - ChikaClau".

2. Website Backend

The following is a backend design on the Barbato Shop Website:

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1 » Database: olshop
- Structure:** The main tab selected.
- SQL:** SQL editor tab.
- Search:** Search bar.
- Query:** Query builder tab.
- Export:** Export data tab.
- Import:** Import data tab.
- Operations:** Operations tab.
- Privileges:** Privileges tab.
- Routines:** Routines tab.
- Events:** Events tab.
- Triggers:** Triggers tab.

Tables:

Table	Action	Rows	Type	Collation	Size	Overhead
carts	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
migrations	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
products	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
transactions	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
transaction_details	Browse Structure Search Insert Empty Drop	22	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
users	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
7 tables	Sum	68	InnoDB	utf8mb4_general_ci	240.0 Kib	0 B

3. Website Code

3.1 Controller

CartController.php

```
12 class CartController extends Controller
13 {
14     public function __construct()
15     {
16         $this->middleware('is_user');
17     }
18
19     public function index()
20     {
21         $id = Auth::id();
22
23         $data = Cart::with('product')->where('user_id', $id)->get();
24         $total = Cart::where('user_id', $id)->sum('sub_total');
25
26         return view('cart.index', ["data" => $data, "total" => $total]);
27     }
28
29     public function store(Request $request)
30     {
31         $id = Auth::id();
32
33         $data = [
34             'product_id' => $request->id,
35             'user_id' => $id,
36             'qty' => $request->qty,
37             'sub_total' => $request->qty*$request->price,
38         ];
39
40         Cart::create($data);
41         Session::flash('success-add');
42
43         return redirect()->route('cart');
44     }
}
```

```
47     public function destroy($id)
48     {
49         Cart::destroy($id);
50
51         Session::flash('success-delete');
52
53         return redirect()->route('cart');
54     }
55
56     public function purchase()
57     {
58         $id = Auth::id();
59         $total = Cart::where('user_id', $id)->sum('sub_total');
60
61         $data = [
62             'user_id' => $id,
63             'total' => $total,
64         ];
65
66         $transaction = Transaction::create($data);
67
68         $transaction_detail = [];
69
70         $transaction_detail_data = Cart::where('user_id', $id)->get();
71
72         foreach($transaction_detail_data as $dt){
73             $detail_arr=[
74                 'transaction_id'=> $transaction->id,
75                 'product_id' => $dt->product_id,
76                 'qty' => $dt->qty,
77                 'sub_total' => $dt->sub_total
78             ];
79
80             array_push($transaction_detail, $detail_arr);
81         }
82
83         TransactionDetail::insert($transaction_detail);
84
85         Cart::where('user_id', $id)->delete();
86
87         Session::flash('success-purchase');
88
89         return redirect()->route('transaction');
90     }
91 }
```

HomeController.php

```
8  class HomeController extends Controller
9  {
10     /**
11      * Create a new controller instance.
12      *
13      * @return void
14      */
15     public function __construct()
16     {
17         // $this->middleware('auth');
18     }
19
20     /**
21      * Show the application dashboard.
22      *
23      * @return \Illuminate\Contracts\Support\Renderable
24      */
25     public function index(Request $request)
26     {
27         $category = collect(['Food', 'Electronics', 'Beauty', 'Pet']);
28         $product = Product::search($request->search)->get();
29
30         $data = [
31             'categories' => $category,
32             'products' => $product,
33             'search' => $request->search
34         ];
35
36         return view('home', $data);
37     }
38
39     public function show($id)
40     {
41         $data = Product::find($id);
42
43         return view('product.detail', ['data' => $data]);
44     }
45
46     public function category($category)
47     {
48         $product = Product::where('category', $category)->paginate(10);
49
50         $data = [
51             'products' => $product
52         ];
53
54         return view('product.productByCategories', $data);
55     }
56 }
```

ProductController.php

```
13  class ProductController extends Controller
14  {
15      /**
16      * Display a listing of the resource.
17      *
18      * @return \Illuminate\Http\Response
19      */
20
21      public function __construct()
22      {
23          $this->middleware('is_admin');
24      }
25
26      public function index(Request $request)
27      {
28          $product = Product::search($request->search)->paginate(5);
29
30          $data = [
31              'products' => $product
32          ];
33
34          return view('product.index', $data);
35      }
36
37
38      /**
39      * Show the form for creating a new resource.
40      *
41      * @return \Illuminate\Http\Response
42      */
43      public function create()
44      {
45          $category = collect(['Food', 'Electronics', 'Beauty', 'Pet']);
46          return view('product.add', ['category' => $category]);
47      }
48
49
50
51      public function store(Request $request)
52      {
53          $validator = Validator::make($request->all(), [
54              'name' => 'required',
55              'category' => 'required',
56              'detail' => 'required',
57              'price' => 'required|integer',
58              'photo' => 'required|mimes:jpg,jpeg,png'
59          ]);
60
61          if ($validator->fails()) {
62              return redirect()
63                  ->route('product_add')
64                  ->withErrors($validator)
65                  ->withInput();
66          }
67          $validated = $validator->validated();
68
69          $new_name = Storage::disk('public')->put('images' , $request->file('photo'));
70          $explode = explode('/', $new_name);
71          $validated['photo'] = $explode[1];
72
73          Product::create($validated);
74
75          Session::flash('success-store');
76
77          return redirect()->route('product');
78
79
80
81      }
82
83  }
```

```
91 |     public function edit($id)
92 |     {
93 |         $data = Product::find($id);
94 |         $category = collect(['Food', 'Electronics', 'Beauty', 'Pet']);
95 |
96 |         return view('product.edit', ['data' => $data, 'category' => $category]);
97 |     }
98 |
99 |     /**
100 |      * Update the specified resource in storage.
101 |      *
102 |      * @param \Illuminate\Http\Request $request
103 |      * @param \App\Models\Product $product
104 |      * @return \Illuminate\Http\Response
105 |      */
106 |     public function update(Request $request)
107 |     {
108 |         $product = Product::find($request->id);
109 |
110 |         $validator = Validator::make($request->all(), [
111 |             'name' => 'required',
112 |             'category' => 'required',
113 |             'detail' => 'required',
114 |             'price' => 'required|integer',
115 |             'photo' => 'required|mimes:jpg,jpeg,png'
116 |         ]);
117 |
118 |         if ($validator->fails()) {
119 |             return redirect()
120 |                 ->route('product_edit', ['id'=>$request->id])
121 |                 ->withErrors($validator)
122 |                 ->withInput();
123 |         }
124 |         $validated = $validator->validated();
125 |         File::delete('images/' . $product->photo);
126 |         $new_name = Storage::disk('public')->put('images' , $request->file('photo'));
127 |         $explode = explode('/', $new_name);
128 |         $validated['photo'] = $explode[1];
129 |
130 |         $product->update($validated);
131 |
132 |         Session::flash('success-update');
133 |
134 |         return redirect()->route('product');
135 |     }
136 |
137 |     /**
138 |      * Remove the specified resource from storage.
139 |      *
140 |      * @param \App\Models\Product $product
141 |      * @return \Illuminate\Http\Response
142 |      */
143 |     public function destroy($id)
144 |     {
145 |         $product = Product::find($id);
146 |
147 |         File::delete('images/' . $product->photo);
148 |         $product->delete();
149 |
150 |         Session::flash('success-delete');
151 |         return redirect()->route('product');
152 |     }
153 }
```

ProfileController.php

```
9  class ProfileController extends Controller
10 {
11     public function __construct()
12     {
13         $this->middleware('auth');
14     }
15
16     public function index()
17     {
18         $id = Auth::id();
19
20         $user = User::find($id);
21
22         $data = [
23             'user' => $user
24         ];
25
26         return view('profile.index', $data);
27
28     }
29 }
```

TransactionController.php

```
9  class TransactionController extends Controller
10 {
11     public function __construct()
12     {
13         $this->middleware('is_user');
14     }
15
16     public function index()
17     {
18         $id = Auth::id();
19         $transaction = Transaction::with('detail.product')->where('user_id', $id)->get();
20         $quantity = 0;
21
22         $data = [
23             'transaction' => $transaction,
24             'quantity' => $quantity,
25         ];
26
27         // dd($transaction);1
28
29         return view('transaction.index', $data);
30
31     }
32 }
```

3.2 Middleware

is_admin.php

```
8  class is_admin
9  {
10     public function handle(Request $request, Closure $next)
11     {
12         if(!auth()->check() || !auth()->user()->is_admin){
13             abort(403);
14         }
15         return $next($request);
16     }
17 }
```

Is_user.php

```
8  class is_user
9  {
10     /**
11      * Handle an incoming request.
12      *
13      * @param  \Illuminate\Http\Request  $request
14      * @param  \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
15      * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
16      */
17     public function handle(Request $request, Closure $next)
18     {
19         if(!auth()->check() || auth()->user()->is_admin){
20             abort(403);
21         }
22         return $next($request);
23     }
24 }
```

3.3 Models

Cart.php

```
9  class Cart extends Model
10 {
11     use HasFactory;
12
13     protected $guarded = [ '' ];
14
15     public function product()
16     {
17         return $this->belongsTo(Product::class);
18     }
19 }
```

Product.php

```
8  class Product extends Model
9  {
10 |     use HasFactory;
11 |
12 |     protected $guarded = [''];
13 |
14 |     public function scopeSearch($query, $search)
15 |     {
16 |         $query->when($search ?? false, function($query, $search){
17 |             return $query->where('name', 'like', "%$search%");
18 |         });
19 |     }
20 }
```

Transaction.php

```
9  class Transaction extends Model
10 {
11 |     use HasFactory;
12 |
13 |     protected $guarded = [''];
14 |
15 |     public function detail()
16 |     {
17 |         return $this->hasMany(TransactionDetail::class);
18 |     }
19 }
```

TransactionDetail.php

```
8  class TransactionDetail extends Model
9  {
10 |     use HasFactory;
11 |
12 |     protected $guarded = [''];
13 |
14 |     public function product()
15 |     {
16 |         return $this->belongsTo(Product::class);
17 |     }
18 }
```

User.php

```
11  class User extends Authenticatable
12  {
13      use HasApiTokens, HasFactory, Notifiable;
14
15      /**
16      * The attributes that are mass assignable.
17      *
18      * @var array<int, string>
19      */
20      protected $guarded = [];
21
22      /**
23      * The attributes that should be hidden for serialization.
24      *
25      * @var array<int, string>
26      */
27      protected $hidden = [
28          'password',
29          'remember_token',
30      ];
31
32      /**
33      * The attributes that should be cast.
34      *
35      * @var array<string, string>
36      */
37      protected $casts = [
38          'email_verified_at' => 'datetime',
39      ];
40 }
```

3.4 Database

3.4.1 Factories

UserFactory.php

```
11 class UserFactory extends Factory
12 {
13     /**
14      * Define the model's default state.
15      *
16      * @return array<string, mixed>
17      */
18     public function definition()
19     {
20         return [
21             'name' => fake()->name(),
22             'email' => fake()->unique()->safeEmail(),
23             'email_verified_at' => now(),
24             'password' => '$2y$10$92IXUNpkj00rQ5byMi.Ye4oKoEa3RqallC/.og/at2.uheWG/igi', // password
25             'remember_token' => str::random(10),
26         ];
27     }
28
29     /**
30      * Indicate that the model's email address should be unverified.
31      *
32      * @return static
33      */
34     public function unverified()
35     {
36         return $this->state(fn (array $attributes) => [
37             'email_verified_at' => null,
38         ]);
39     }
40 }
```

3.4.2 Migrations

2014_10_12_000000_create_users_table.php

```
7     return new class extends Migration
8     {
9         /**
10          * Run the migrations.
11          *
12          * @return void
13          */
14         public function up()
15         {
16             Schema::create('users', function (Blueprint $table) {
17                 $table->id();
18                 $table->string('name');
19                 $table->string('email')->unique();
20                 $table->string('password');
21                 $table->enum('gender', ['Male', 'Female']);
22                 $table->date('dob');
23                 $table->enum('country', ['Indonesia', 'United States', 'Argentina', 'Costa Rica', 'Japan', 'Brazil']);
24                 $table->boolean('is_admin');
25                 $table->rememberToken();
26                 $table->timestamps();
27             });
28         }
29
30         /**
31          * Reverse the migrations.
32          *
33          * @return void
34          */
35         public function down()
36         {
37             Schema::dropIfExists('users');
38         }
39     };
```

2022_11_28_202919_product.php

```
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14      public function up()
15      {
16          Schema::create('products', function (Blueprint $table) {
17              $table->id();
18              $table->string('name');
19              $table->enum('category', ['Food', 'Electronics', 'Beauty', 'Pet']);
20              $table->text('detail');
21              $table->bigInteger('price');
22              $table->text('photo');
23              $table->timestamps();
24          });
25      }
26
27      /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32      public function down()
33      {
34          Schema::dropIfExists('products');
35      }
36  };
```

2022_12_28_202456_create_carts_table.php

```
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14      public function up()
15      {
16          Schema::create('carts', function (Blueprint $table) {
17              $table->id();
18              $table->foreignId('product_id')->constrained()
19                  ->onUpdate('cascade')
20                  ->onDelete('cascade');
21              $table->foreignId('user_id')->constrained()
22                  ->onUpdate('cascade')
23                  ->onDelete('cascade');
24              $table->integer('qty');
25              $table->bigInteger('sub_total');

26              $table->timestamps();
27          });
28      }
29
30      /**
31      * Reverse the migrations.
32      *
33      * @return void
34      */
35      public function down()
36      {
37          Schema::dropIfExists('carts');
38      }
39  };
```

2022_12_28_202946_transaction.php

```
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14      public function up()
15      {
16          Schema::create('transactions', function (Blueprint $table) {
17              $table->id();
18              $table->foreignId('user_id')->constrained()
19                  ->onUpdate('cascade')
20                  ->onDelete('cascade');
21              $table->bigInteger('total');
22              $table->timestamps();    Richard Tandy Japutra, 6 days
23          });
24      }
25
26      /**
27      * Reverse the migrations.
28      *
29      * @return void
30      */
31      public function down()
32      {
33          Schema::dropIfExists('transactions');
34      }
35  };

```

2022_12_28_203010_transaction_detail.php

```
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14      public function up()
15      {
16          Schema::create('transaction_details', function (Blueprint $table) {
17              $table->id();
18              $table->foreignId('transaction_id')->constrained()
19                  ->onUpdate('cascade')
20                  ->onDelete('cascade');
21              $table->foreignId('product_id')->nullable()
22                  ->constrained()
23                  ->onUpdate('cascade')
24                  ->nullOnDelete();
25              $table->integer('qty');
26              $table->bigInteger('sub_total');
27              $table->timestamps();
28          });
29      }
30
31      /**
32      * Reverse the migrations.
33      *
34      * @return void
35      */
36      public function down()
37      {
38          Schema::dropIfExists('transactionDetails');
39      }
40  };

```

3.4.3 Seeders

CartSeeder.php

```
9  class CartSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      *
14      * @return void
15      */
16     public function run()
17     {
18         for($a=1;$a<=10;$a++){
19             DB::table('carts')->insert([
20                 'product_id' => rand(1,12),
21                 'user_id' => rand(3,5),
22                 'qty' => 2,
23                 'sub_total' => 20000,
24             ]);
25         }
26     }
27 }
```

DatabaseSeeder.php

```
8  class DatabaseSeeder extends Seeder
9  {
10     /**
11      * Seed the application's database.
12      *
13      * @return void
14      */
15     public function run()
16     {
17         $this->call([
18             UserSeeder::class,
19             ProductSeeder::class,
20             CartSeeder::class,
21             TransactionSeeder::class,
22             TransactionDetailSeeder::class,
23         ]);
24     }
25 }
```

ProductSeeder.php

```
9  class ProductSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      *
14      * @return void
15      */
16     public function run()
17     {
18         $category = ['Food', 'Electronics', 'Beauty', 'Pet'];
19         for($a=1;$a<=12;$a++){
20             DB::table('products')->insert([
21                 'name' => "Product $a",
22                 'category' => $category[rand(0, 3)],
23                 'Detail' => "Lorem, ipsum dolor sit amet consectetur adipisicing elit.  
Iusto natus accusamus, veritatis corporis expedita eveniet ab esse. Possimus, minimal!  
Labore reprehenderit quo aliquam ipsam voluptates excepturi recusandae porro ut eveniet?",
24                 'price' => 10000,
25                 'photo' => "product-$a.jpg"
26             ]);
27         }
28     }
29 }
30 }
31 }
```

TransactionDetailSeeder.php

```
9  class TransactionDetailSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      *
14      * @return void
15      */
16     public function run()
17     {
18         for($a=1;$a<=20;$a++){
19             DB::table('transaction_details')->insert([
20                 'transaction_id' => rand(1,10),
21                 'product_id' => rand(1,12),
22                 'qty' => 2,
23                 'sub_total' => 20000,
24             ]);
25         }
26     }
27 }
```

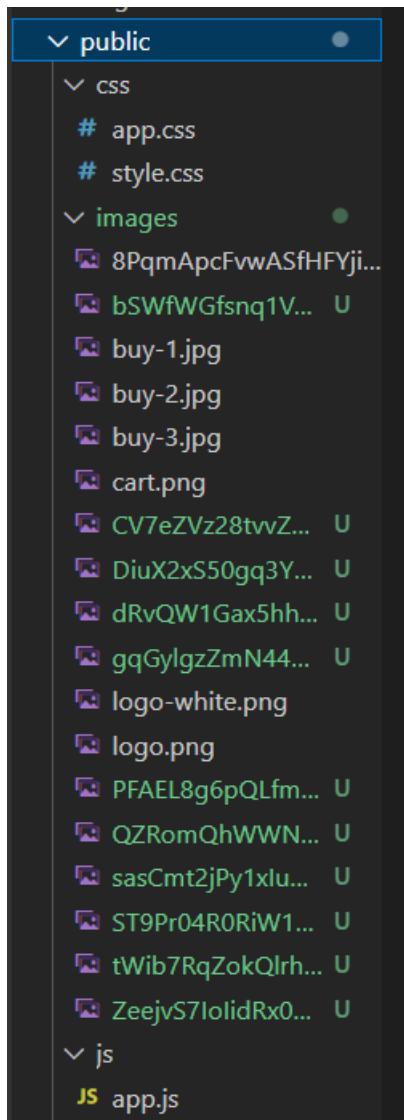
TransactionSeeder.php

```
 9  class TransactionSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      *
14      * @return void
15      */
16     public function run()
17     {
18         for($a=1;$a<=10;$a++){
19             DB::table('transactions')->insert([
20                 'user_id' => rand(3,5),
21                 'total' => 200000,
22                 'created_at' => date("Y-m-d H:i:s")
23             ]);
24         }
25     }
26 }
```

UserSeeder.php

```
10  class UserSeeder extends Seeder
11 {
12     /**
13      * Run the database seeds.
14      *
15      * @return void
16      */
17     public function run()
18     {
19         $country = ['Indonesia', 'United States', 'Argentina', 'Costa Rica', 'Japan', 'Brazil'];
20         $gender = ['Male', 'Female'];
21         DB::table('users')->insert(
22             [
23                 'name' => "Admin1",
24                 'email' => "admin@gmail.com",
25                 'password' => Hash::make("admin1"),
26                 'gender' => "Male",
27                 'dob' => date('Y-m-d'),
28                 'country' => $country[rand(0,5)],
29                 'is_admin' => true
30             ]
31         );
32         DB::table('users')->insert(
33             [
34                 'name' => "Admin2",
35                 'email' => "admin2@gmail.com",
36                 'password' => Hash::make("admin2"),
37                 'gender' => "Female",
38                 'dob' => date('Y-m-d'),
39                 'country' => $country[rand(0,5)],
40                 'is_admin' => true
41             ]
42         );
43         for($a=1;$a<=3;$a++){
44             DB::table('users')->insert([
45                 'name' => "User$a",
46                 'email' => "user$a@gmail.com",
47                 'password' => Hash::make("user$a"),
48                 'gender' => $gender[rand(0,1)],
49                 'dob' => date('Y-m-d'),
50                 'country' => $country[rand(0,5)],
51                 'is_admin' => false
52             ]);
53         }
54     }
}
```

3.5 Public



3.6 Routes

api.php

```
17     Route::middleware('auth:sanctum')->get('/user', function (Request $request) {  
18         return $request->user();  
19     });
```

channels.php

```
16     Broadcast::channel('App\Models\User.{id}', function ($user, $id) {  
17         return (int) $user->id === (int) $id;  
18     });
```

console.php

```
17  Artisan::command('inspire', function () {
18      $this->comment(Inspiring::quote());
19  })->purpose('Display an inspiring quote');
```

web.php

```
20 Route::get('/welcome', function () {
21     return view('welcome');
22 });
23
24 Auth::routes();
25
26 //home
27 Route::get('/', [HomeController::class, 'index'])->name('home');
28 Route::get('/profile', [ProfileController::class, 'index'])->name('profile');
29 Route::get('/product/category/{category}', [HomeController::class, 'category'])->name('product_category');
30
31 //product
32 Route::get('/product', [ProductController::class, 'index'])->name('product');
33 Route::get('/product/detail/{id}', [HomeController::class, 'show'])->name('product_detail');
34 Route::get('/product/add', [ProductController::class, 'create'])->name('product_add');
35 Route::post('/product/store', [ProductController::class, 'store'])->name('product_store');
36 Route::get('/product/edit/{id}', [ProductController::class, 'edit'])->name('product_edit');
37 Route::post('/product/update', [ProductController::class, 'update'])->name('product_update');
38 Route::get('/product/destroy/{id}', [ProductController::class, 'destroy'])->name('product_destroy');
39
40 //cart
41 Route::get('/cart', [CartController::class, 'index'])->name('cart');
42 Route::post('/cart/store', [CartController::class, 'store'])->name('cart_store');
43 Route::get('/cart/purchase', [CartController::class, 'purchase'])->name('cart_purchase');
44 Route::get('/cart/destroy/{id}', [CartController::class, 'destroy'])->name('cart_destroy');
45
46 //transaction
47 Route::get('/transaction', [TransactionController::class, 'index'])->name('transaction');
```

3.7 .env

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:RmD9JTRwGUfuA8llUyoyhwJBWcL45XBxin3ddz6/djQ=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=olshop
15 DB_USERNAME=root
16 DB_PASSWORD=
```

F. Conclusion

In this final project, it is concluded that the use of website-based information technology systems can support the development of information technology in the industrial era 4.0. This is very helpful, especially because of the effects of the pandemic where there is a social distancing policy to reduce people's mobility, making this modern society use all the ways that can be done from home, one of which is shopping activities carried out through websites or applications. So hopefully with the website that we have created using the Laravel 8 framework, it can help minimize existing problems.

G. References

Framework: Laravel 8 <https://laravel.com/>

Code Editor: Visual Studio Code (VSCode) <https://code.visualstudio.com/>

Other Tools:

XAMPP Control Panel <https://www.apachefriends.org/download.html>

App Diagram <https://app.diagrams.net/>

References for Theory Explanation:

<https://www.lucidchart.com/blog/types-of-UML-diagrams>

<https://www.ibm.com/topics/database-schema>

<https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>

<https://asq.org/quality-resources/flowchart>

<https://www.zenflowchart.com/symbol-flowchart>

Image Assets:

<https://genshin.gg/cooking/>

<https://www.asus.com/id/>

<https://www.apple.com/id/iphone/>

<https://www.sociolla.com/>

<https://www.cosrx.com/>

<https://rohto.co.id/product/group/hada-labo>

<https://www.ruparupa.com/>

H. Others

Project:

<https://github.com/chikaciau/WP-FP>

*The project can be opened by clone and open with GitHub Desktop or download the ZIP file

*If there are errors from the project such as unable to open / 500 errors / loading continuously, please inform us

<https://www.niagahoster.co.id/blog/laravel-error-500/>

<https://stackoverflow.com/questions/18022809/how-to-solve-error-mysql-shutdown-unexpectedly>