

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY  
UNIVERSITY OF TECHNOLOGY  
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



## Programming Intergration Project (CO3103)

---

# Restaurant's Web Order and CRM system Report

Class CC01

---

**Instructor:** Quan Thanh Tho

**Group Members:**

Tieu Viet Trong Nghia	1852611
Pham Van Minh Toan	1953028
Hoang Ha Giang	1952659
Nguyen Khuong	1952310
Tran Quoc Duy	1952214

Ho Chi Minh city, 8th March 2022



## Contents

<b>1 INTRODUCTION</b>	<b>2</b>
1.1 General Introduction . . . . .	2
1.2 Web Order . . . . .	2
1.3 CRM system . . . . .	3
<b>2 WORKLOAD SHARING</b>	<b>4</b>
2.1 Web Order . . . . .	4
2.2 CRM System . . . . .	4
<b>3 TECHNOLOGIES</b>	<b>5</b>
3.1 Front-end: React Framework . . . . .	5
3.2 Back-end: Lavarel Framework + Database: MySQL . . . . .	5
3.3 Deployment: Google Cloud, Heroku and CloudFlare . . . . .	5
<b>4 UI AND SCREEN FLOW</b>	<b>7</b>
4.1 Web Order . . . . .	7
4.2 CRM Dashboard . . . . .	9
4.2.1 Registration/Login . . . . .	9
4.2.2 Customer Management . . . . .	10
4.2.3 Menu Management . . . . .	11
4.2.4 Voucher Management . . . . .	13
4.2.5 Order Management . . . . .	14
<b>5 FEATURE DETAILS</b>	<b>18</b>
5.1 Web Order . . . . .	18
5.1.1 Categorize Product . . . . .	18
5.1.2 Search Product . . . . .	18
5.1.3 Add to/Remove from Cart . . . . .	18
5.1.4 Implementing Payment . . . . .	19
5.2 CRM Dashboard . . . . .	20
5.2.1 Customer Management . . . . .	20
5.2.2 Menu Management . . . . .	20
5.2.3 Order Management . . . . .	20
5.2.4 Voucher Management . . . . .	21
5.2.5 Messenger Bot . . . . .	21
5.2.6 Privileges . . . . .	23
<b>6 Disclaimer</b>	<b>26</b>

# 1 INTRODUCTION

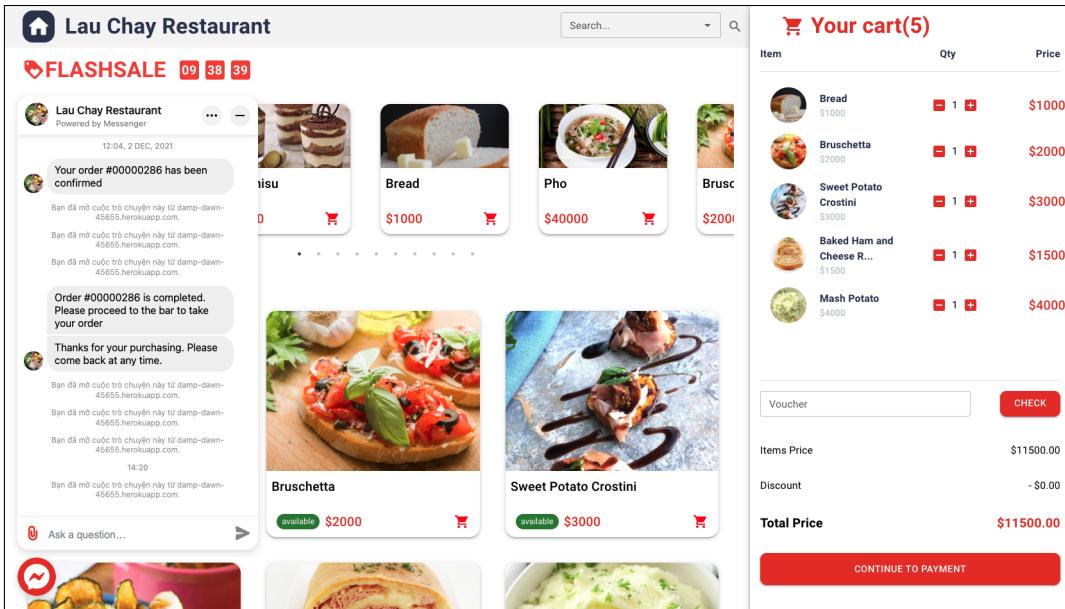
## 1.1 General Introduction

In the modern days, using computer systems to help managing a business is getting more and more popular due to its convenience and effectiveness. Especially in restaurants where thousands of items are sold every day. There's a need for a system to help with accepting payments and keeping track of customers and sales while collecting and analyzing customer's data to assist in the marketing and customer relationship department.

The aim of this project is to provide one out of many solutions which helps a restaurant business do what we mentioned above to run efficiently. It consists of two parts, the Web order and the CRM system.

## 1.2 Web Order

The purpose of this website is for customers to interact with the items sold by the restaurant. The website will display a menu of dishes that have been sorted by category. Customers can view an item's information in details and select multiple products at the same time with different quantities to add to the cart. In addition, customers can also enter the voucher code that the system has sent to the customer through the chatbot channel for each order. This will help the restaurant attract more customers with promotions. Finally, at the payment stage, customers can choose different payment methods to pay for their orders.



The screenshot shows the web order interface for Lau Chay Restaurant. On the left, there's a sidebar with a message from the restaurant confirming an order and another message from a customer asking a question. The main area displays a menu with items like Nisu, Bread, Pho, and Bruschetta. On the right, a "Your cart(5)" section shows five items: Bread (\$1000), Bruschetta (\$2000), Sweet Potato Crostini (\$3000), Baked Ham and Cheese R... (\$1500), and Mash Potato (\$4000). Below the cart, there's a "Voucher" input field and a "CHECK" button. The total price is listed as \$11500.00. At the bottom, there's a "CONTINUE TO PAYMENT" button.

Figure 1: Web Order interface



### 1.3 CRM system

The purpose of a CRM system is to store information regarding customers who have made transactions with the restaurant. From there, it is possible to make a list of potential customers for the restaurant so that reasonable business strategies can be devised. Therefore, within the scope of this project, we perform some basic functions that a CRM system should have, including: customer information management, order management, restaurant menu management, voucher management,... In addition, this CRM system will also decentralize administration based on employee roles in the restaurant, which include: Owner, Clerk and Chef - each role will be limited with only a handful of features to ensure the security of restaurant data.

The screenshot shows the CRM system's interface. On the left is a dark sidebar with a navigation menu:

- Dashboard (selected)
- NEW
- MANAGE CUSTOMERS
- Customers
- MANAGE MENU
- Restaurant Menu
- MANAGE ORDERS
- Orders (selected)
- MANAGE VOUCHERS
- Voucher
- Logout

The main content area is titled "Home". It has a header with "Dashboard" and "NEW" buttons, a "Home" link, and three icons. Below the header is a "New" button. The main content is a table titled "ORDERS" with the following data:

ID	Phone Number	Address	Total purchased	Payment method	Created At	Status	Delete
299	9876543210	123 abcd	50000\$	PayPal	04:49 12-12-2021	Completed	
300	0987654321	abcdef	86000\$	PayPal	03:33 13-12-2021	Complete	
301	0123987465	asdfgh	35000\$	PayPal	03:23 13-12-2021	Prepare	
298	1234567890	abcdef	3000\$	PayPal	03:10 12-12-2021	Completed	
297	0123456789	abxyz	7000\$	PayPal	03:19 12-12-2021	Waiting	

Figure 2: CRM system interface

## 2 WORKLOAD SHARING

In this project, our team divided the work into 2 main parts: Web Order and CRM system.

### 2.1 Web Order

No.	Feature	Name
1	Design the interface (Figma)	Nghia
2	Convert from Figma to Web Order interface	Nghia
3	Divide products into categories	Nghia
4	Search for product by product's name	Nghia
5	Manage the cart	Nghia
6	Apply vouchers	Duy
7	Make payment (PayPal, Cash)	Khuong

Table 1: Workload Sharing for Web Order

### 2.2 CRM System

No.	Feature	Name
1	Design the interface (Figma)	Nghia
2	Convert from Figma to CRM interface	Nghia
3	Implement CRUD for Customer management	Khuong
4	Implement CRUD for Order management	Khuong
5	Implement CRUD for Voucher management	Duy
6	Implement CRUD for Menu management	Duy
7	Make API to link with Web Order	Toan
8	Modify payment interface logic	Toan
9	Implement add-ons for Customer management (search bar, exporter)	Khuong
10	Add livewire to the interface (responsiveness)	Giang
11	Add Customer subscription and Purchased History page	Giang
12	Make Chatbot integration	Toan
13	Divide admin, user role	Giang
14	Create order status	Toan
15	Deploy to the cloud	Toan
16	Make authentication	Toan

Table 2: Workload Sharing for CRM System

### 3 TECHNOLOGIES



#### 3.1 Front-end: React Framework

In this project, we decided to use React because it is a popular Javascript library for building user interfaces. Because of the nature of this website using a lot of the same components, for example product items are repeated a lot. React is very supportive in terms of components. Moreover, the application of Redux in shopping cart management is also the reason why we choose React, it helps to optimize redundant code and does not cause confusion in the product development process. Finally, React is so widely used that there are a lot of plugins developed with it, which makes the website design process much faster, without having to build small components from scratch.

#### 3.2 Back-end: Laravel Framework + Database: MySQL

Laravel is an open-source PHP framework that follows the MVC model with pre-built features such as authentication, routes, migrate database,... Additionally, PHP has a huge range of libraries that support common features.

This allows us to rapidly develop a website and deliver a full functional website to users while still finding it easy to maintain and reuse.

Laravel also goes very well with MySQL - a relational database management system based on SQL, it makes connecting with databases and running queries extremely simple. We can interact with MySQL database through ORM(Object-Relational-Mapping), which is more user-friendly compared to raw SQL commands and can help us minimize the risk of SQL Injections. Laravel also helps migrate and quickly deploy schemas to the server.

#### 3.3 Deployment: Google Cloud, Heroku and CloudFlare

##### Google Cloud:

- Using Google Cloud Platform in this project is very beneficial to us because it supports monitoring tools, easily monitoring the up and down status of the website. High response time and reliability. Moreover, when using Cloud VPS, it is very easy to edit whenever the project scales. In addition, it's free 3 months trial for users enough to complete the course project.
- However, the disadvantage of Cloud VPS is that it takes a lot of time to get used to its interface at first, and users must have some prerequisite knowledge of Linux to be able to work with Virtual Machines.

##### CloudFlare:

- As for CloudFlare, the reason we use it is because it offers free SSL certificates. When we deploy CRM to Google Cloud, they will give us an IP address <http://34.126.93.124/>, although can initialize API for this IP address <http://34.126.93.124/api/menu>, but they do not support Let's Encrypt (SSL certificate), meaning the website is not secure. Therefore, when we request the API to get the product, the browser will refuse with the following error:

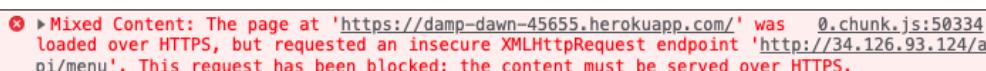
Mixed Content: The page at '<https://damp-dawn-45655.herokuapp.com/>' was loaded over HTTPS, but requested an insecure XMLHttpRequest endpoint '<http://34.126.93.124/api/menu>'. This request has been blocked; the content must be served over HTTPS.

Figure 3: HTTP Blocked Error



- That's why using CloudFlare will help keep our domain name secure <https://test.greenup.com.vn/api/menu> and browsers will allow it to load
- In addition, we use Heroku to deploy the Web Order interface

## 4 UI AND SCREEN FLOW

### 4.1 Web Order

The following images will show the flow order when the user visits the website, in order from top to bottom

- Step 1:** Users access the order page, here they can view product information, search for products, add or remove products in the cart. In addition, they can also make chats with the chatbot integrated into Facebook's messenger

Your cart(5)		
Item	Qty	Price
Bread	1	\$1000
Bruschetta	1	\$2000
Sweet Potato Crostini	1	\$3000
Baked Ham and Cheese R...	1	\$1500
Mash Potato	1	\$4000

Voucher  CHECK

Items Price \$11500.00

Discount -\$0.00

Total Price \$11500.00

CONTINUE TO PAYMENT

Figure 4: Order interface

SKU	Appetizer	Unit Price
#4	Bruschetta	\$2000

Available Item: 14

Description  
Grilled country bread with Roma tomatoes, olive oil, garlic and basil.

ADD TO CART

Your cart(1)		
Item	Qty	Price
Bruschetta	1	\$2000

Voucher  CHECK

Items Price \$2000.00

Discount -\$0.00

Total Price \$2000.00

CONTINUE TO PAYMENT

Figure 5: Product Detail

- Step 2:** After clicking "Continue to Payment", the user is navigated to the Checkout page. Here the user is asked to enter the shipping address information and click "Next"



The screenshot shows the 'Checkout' process for 'Lau Chay Restaurant'. The current step is 'Shipping address'. The user has entered their full name as 'Tieu Viet Trong Nghia' and their address as 'Tan Phu Ho Chi Minh'. A phone number '0933827830' is also provided. A checkbox labeled 'Use this address for payment details' is checked. A red 'NEXT' button is visible at the bottom right.

Figure 6: Shipping Address

- **Step 3:** In this step, the system allows users to review the order, if all information is correct, then click "Next"

The screenshot shows the 'Checkout' process for 'Lau Chay Restaurant'. The current step is 'Review your order'. The 'Shipping address' step is completed. The 'Order summary' section lists the following items:

- Bread: bread and butter - \$1000
- Bruschetta: Grilled country bread with Roma tomatoes, olive oil, garlic and basil. - \$2000
- Sweet Potato Crostini: Sweet Potato Crostini with Prosciutto Honey Roasted Figs - \$3000
- Baked Ham and Cheese Rolls: Pastry, cheese, ham, and a nice sauce on top and you have a crispy exterior with gooey interior that is always popular! - \$1500
- Mash Potato: Creamy mash potato bowl - \$4000

The total amount is \$11500. The 'Payment method' is set to 'Paypal'. A red 'NEXT' button is visible at the bottom right.

Figure 7: Review Order

- **Step 4:** In this step, users have 2 payment options: PayPal or Cash. After successful payment, the server side will handle the remaining processes

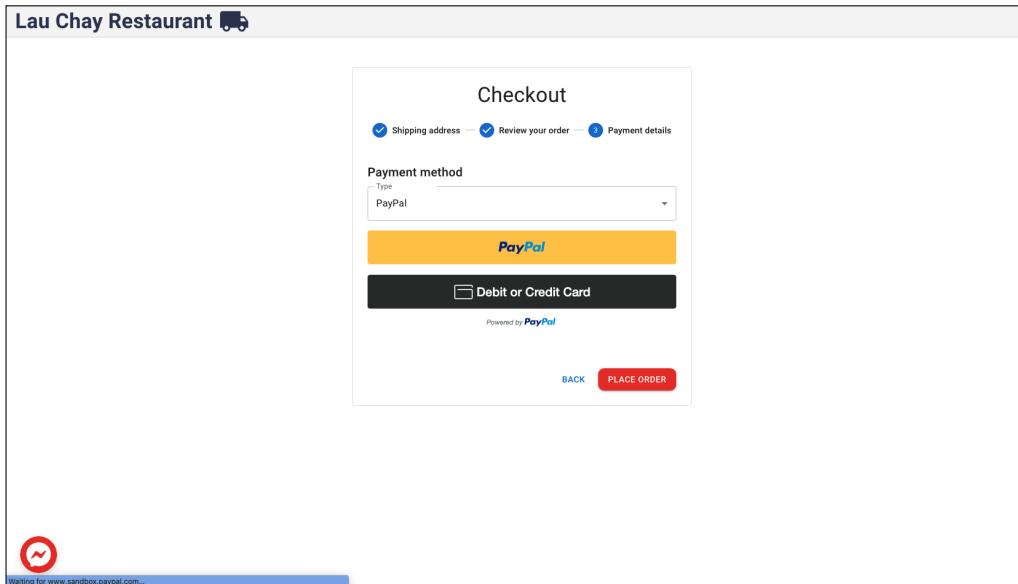


Figure 8: Payment

## 4.2 CRM Dashboard

For the flows of the CRM system, because there are many different functions, we divide them into many small flows for easy observation. These flows will be each function of the system, including: Registration/Login, Customer management, Restaurant menu management, Order management, Voucher management.

### 4.2.1 Registration/Login

- **Step 1:** User registers/logs in when accessing <https://test.greenup.com.vn/>

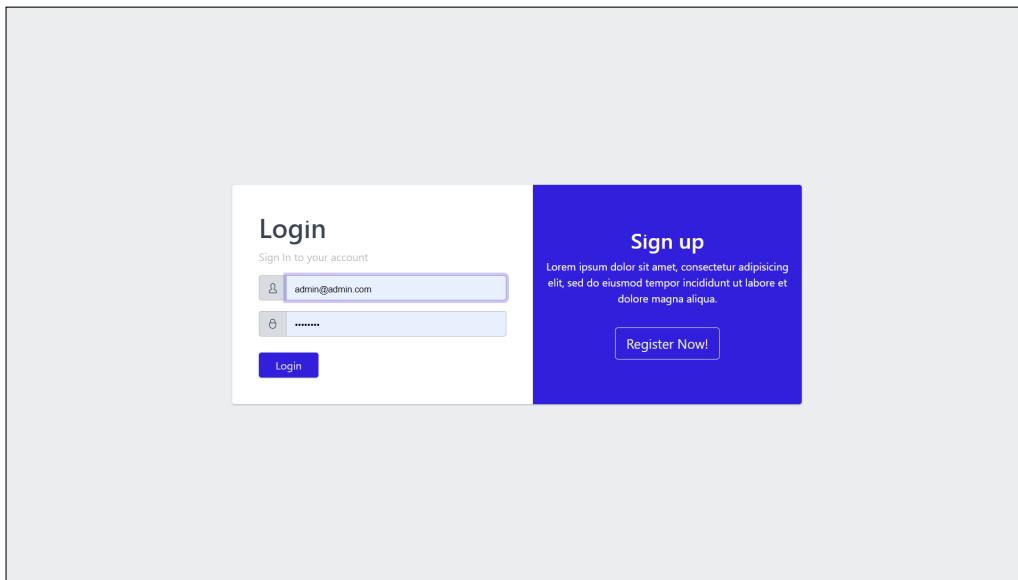


Figure 9: Log in with admin account

- **Step 2:** After successful login, the user is redirected to the home page of CRM

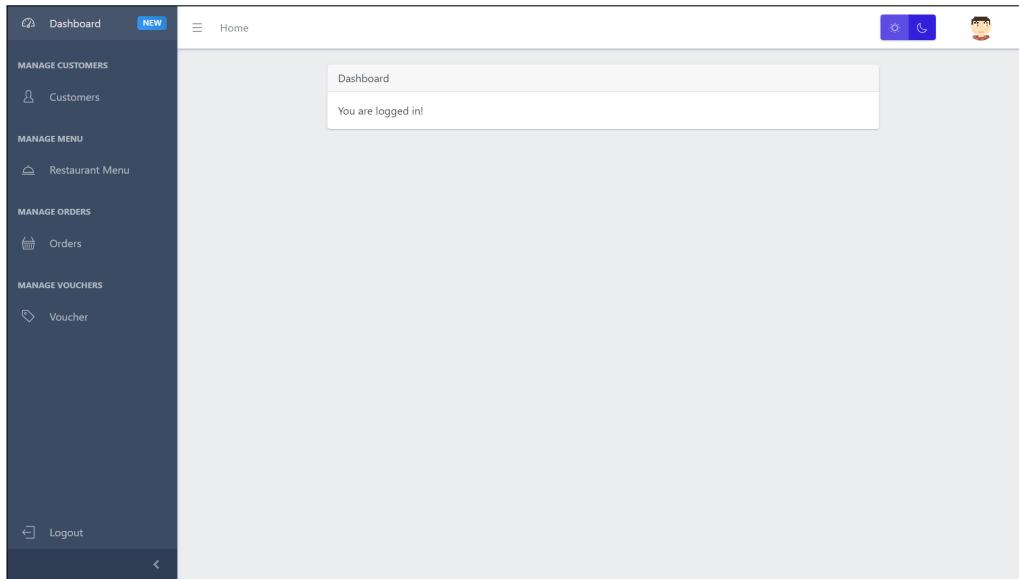


Figure 10: Admin home page

#### 4.2.2 Customer Management

- **Step 1:** From the Admin home page, click on "Customer" on the left to open the customer page. Customer records can be added, viewed, edited, deleted and exported to .xlsx file. On the right side of each row, click the "Delete" button to delete a row. There's also a search bar on the customer page to help search for a particular record and clicking the green "Export" button will automatically save the current record in an excel file.

Phone Number	Name	Address	Created At	Action
0123456789	N/A	abcxyz	12-12-2021	<button>Edit</button> <button>Delete</button>
1234567890	N/A	acdef	12-12-2021	<button>Edit</button> <button>Delete</button>
9876543210	N/A	123 abcd	12-12-2021	<button>Edit</button> <button>Delete</button>
0987654321	N/A	abcdef	13-12-2021	<button>Edit</button> <button>Delete</button>
0123987465	N/A	asdfgh	13-12-2021	<button>Edit</button> <button>Delete</button>
0912345678	N/A	qwert	13-12-2021	<button>Edit</button> <button>Delete</button>

Figure 11: Customer interface

- **Step 2:** From the customer page, click "New" to open the input page, after entering the information, click "Save" to save the record.



The screenshot shows a web-based application interface. On the left, a dark blue sidebar menu lists several categories: 'Dashboard', 'MANAGE CUSTOMERS' (with 'Customers' selected), 'MANAGE MENU' (with 'Restaurant Menu'), 'MANAGE ORDERS' (with 'Orders'), 'MANAGE VOUCHERS' (with 'Voucher'), and 'Logout'. The main content area is titled 'Home' and contains a form for adding a new customer. The form includes three input fields: 'Full Name(\*)', 'Phone Number(\*)', and 'Address', each with a placeholder text. Below the fields is a blue 'Save' button.

Figure 12: Add a new customer

- **Step 3:** From the main Customer tab, click "Edit" to open the edit button next to the "delete", to view and edit a particular customer record.

This screenshot shows the same application interface as Figure 12, but it displays a different form for editing an existing customer. The 'Customers' option in the sidebar is now highlighted. The main content area shows a customer record with pre-filled fields: 'Name' (empty), 'Phone Number' (0123456789), and 'Address' (abxyz). A blue 'Save' button is located at the bottom of the form.

Figure 13: View and edit the customer with phone number "0123456789"

#### 4.2.3 Menu Management

**Step 1:** From the Admin home page, click on "Menu" on the left to open the Menu page. From here, you can **view, add, edit or delete** any dishes that is on, you can also **switch** between the availability states of a products and filter the list by categories.



ID	Name	Category	Price	Quantity	Availability	Delete
1	Bread	Appetizer	1000\$	10		
2	Chocolate mousse	Desserts	20000\$	0		
3	Pho	Main Course	40000\$	20		
4	Bruschetta	Appetizer	2000\$	14		
5	Sweet Potato Crostini	Appetizer	3000\$	8		
6	Chicken Wings	Main Course	5000\$	18		
7	Zucchini Chips	Appetizer	5000\$	8		
8	Meatballs	Main Course	3000\$	13		
9	Tonkotsu Ramen	Main Course	7000\$	25		
10	Baked Ham and Cheese Rolls	Appetizer	1500\$	11		
11	Kueh Tulu	Desserts	5000\$	20		
12	Mash Potato	Appetizer	4000\$	30		
13	Rojak	Appetizer	20000\$	13		
14	Tortellini soup	Appetizer	3000\$	12		
15	Curry Puff	Appetizer	2000\$	10		
16	Bánh Khoái	Appetizer	2700\$	100		

Figure 14: Manage restaurant's menu

**Step 2:** From the menu page, click "New" to **create** new item, after entering the information, click "Save" to save the record.

The form contains the following fields:

- Name: Instant Noodle
- Description: Hao Hao Noodle only
- Price: 10000
- Quantity: 6
- Group: Main Course
- Image: Choose File HH-CANH-MI-HAI-SAN.jpg

Figure 15: Create a new item

**Step 3:** From the menu page, click on the name of products to open the respective information form, allowing user **view and edit** the information, click "Save" to save the record.



The screenshot shows the Admin interface for managing menu items. On the left, there's a sidebar with navigation links for Dashboard, NEW, Manage Customers, Manage Menu, Manage Orders, and Manage Vouchers. Under Manage Vouchers, there's a link for 'Voucher'. The main content area is titled 'Home' and shows a form for editing an existing item named 'Omachi Instant Noodle'. The form includes fields for Name, Description, Price (123000), Quantity (6), and Group (Main Course). A 'Save' button is at the bottom. To the right of the form is a photograph of a bowl of noodle soup with shrimp.

Figure 16: View and edit an existing item

#### 4.2.4 Voucher Management

- **Step 1:** From the Admin home page, click on "Voucher" on the left to open the Voucher management page. From here, user can **view, create, edit, delete** as well as **switch** between voucher's availability states.

The screenshot shows the Admin interface for managing vouchers. On the left, there's a sidebar with navigation links for Dashboard, NEW, Manage Customers, Manage Menu, Manage Orders, and Manage Vouchers. Under Manage Vouchers, there's a link for 'Voucher'. The main content area is titled 'Home' and shows a table of existing vouchers. The table has columns for Voucher ID, Name, Deduction Amount, Used/Total Voucher, Availability, and Delete. The data in the table is as follows:

Voucher ID	Name	Deduction Amount	Used/Total Voucher	Availability	Delete
ABC1234	Registered Membership	15000\$	6/100	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/>
ABC2492	Voucher_2	50%	5/20	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/>
ABC4138	Voucher_3	30000\$	0/10	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/>
ASD	Voucher_4	10%	0/1200	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/>
ABC5	Voucher_5	10000\$	0/100	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/>
BBB	Voucher BBB	5000\$	0/2000	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/>

Figure 17: Manage vouchers

- **Step 2:** From the voucher page, click "New" to **create** a new voucher, after entering the information, click "Save" to save the record.



Dashboard NEW

MANAGE CUSTOMERS

Customers

MANAGE MENU

Restaurant Menu

MANAGE ORDERS

Orders

MANAGE VOUCHERS

Voucher

Logout

Voucher ID: LAUCHAY10DIEM

Name: Gift Voucher

Type: Percentage

Deduction Amount: 100

Start From: 12/13/2021

End At: 12/19/2021

Number of Released Voucher: 5

Save

Figure 18: Create a new voucher

- **Step 3:** From the voucher page, click "New" to **view and edit**, click "Save" to save the record.

Dashboard NEW

MANAGE CUSTOMERS

Customers

MANAGE MENU

Restaurant Menu

MANAGE ORDERS

Orders

MANAGE VOUCHERS

Voucher

Logout

Voucher ID: LAUCHAYCHACCHAN10DIEM

Name: Gift Voucher

Type: Percentage

Deduction Amount: 100

Start From: 12/13/2021

End At: 12/19/2021

Number of Released Voucher: 5

Number of Used Voucher: 0

Save

Delete

Send voucher

Figure 19: Edit voucher

#### 4.2.5 Order Management

When logging in with an admin account (high privileges), orders can be created, viewed, edited, and deleted.

- **Step 1:** From the admin home page, click on "Order" from the left menu to open the Order tab. Here, the admin can **view, add, edit, delete or update** the status of the orders.



ID	Phone Number	Address	Total purchased	Payment method	Created At	Status	Delete
300	0987654321	abcdef	86000\$		03:33 13-12-2021	<button>Prepare</button>	
299	9876543210	123 abcd	50000\$		04:49 12-12-2021	<button>Complete</button>	
298	1234567890	acbdef	3000\$		03:10 12-12-2021	<button>Completed</button>	
297	0123456789	abxyz	7000\$		03:19 12-12-2021	<button>Waiting</button>	

Figure 20: Manage orders

- **Step 2:** On the top left corner, click on the "New" button to **create** a new order.

Customer's Name (Name will be ignored if customers has been created)

Customer's Phone Number

Address

Voucher  
 Payment method

Item ID  
 Quantity

Figure 21: Create a new order

- **Step 3:** Click on the ID number of an order to **view and edit** its information.



The screenshot shows the 'Orders' section of the application. On the left sidebar, there are links for 'Customers', 'Restaurant Menu', 'Orders', and 'Voucher'. The main area has fields for 'Customer's Name' (placeholder: 'Name will be ignored if customers has been created'), 'Customer's Phone Number' (value: '0987654321'), 'Address' (value: 'abcdef'), 'Voucher' (empty), 'Payment method' (selected: 'Paypal'), 'Item ID' (value: '10'), 'Quantity' (value: '4'), 'Item ID' (value: '3'), and 'Quantity' (value: '2'). A red 'Delete' icon is next to the second item entry. Below these fields is a blue 'Add' button. At the bottom, a blue bar displays the total: 'Total : 86000\$'.

Figure 22: View and edit an order

When logging in with a clerk account (lower than admin privileges), orders can only be created and confirmed.

- **Step 1:** Sign in with the clerk account, then click on "Order" from the left menu to open the Order tab. Here, the clerk can **add and confirm** the orders.

The screenshot shows the 'Orders' tab. The left sidebar has a link for 'New Order'. The main area displays a table with columns: ID, Phone Number, Address, Total purchased, Payment method, Created At, and Status. Two orders are listed:  
Order ID 300: Phone Number 0987654321, Address abcdef, Total purchased 86000đ, Payment method Paypal, Created At 13-12-2021, Status Confirmed (button: 'Confirmed').  
Order ID 297: Phone Number 0123456789, Address abxyz, Total purchased 7000đ, Payment method Paypal, Created At 12-12-2021, Status Pending (button: 'Confirm').

Figure 23: Order tab with a create and confirm buttons for clerk

- **Step 2:** Click on the "New" button from the top left corner or the "New Order" tab from the dashboard to **create** a new order.



The screenshot shows a web-based application interface for creating a new order. On the left, a dark sidebar titled 'MANAGE ORDERS' contains links for 'Dashboard', 'Orders', and '+ New Order'. The main area is titled 'Home' and has a 'NEW' button at the top right. It includes fields for 'Customer's Name' (containing 'Customer 00'), 'Customer's Phone Number' (containing '0123987465'), 'Address' (containing 'asdfgh'), 'Voucher' (empty), 'Payment method' (set to 'Paypal'), 'Item ID' (set to 7), 'Quantity' (set to 7), and a blue 'Add' button. At the bottom is a blue 'Save' button.

Figure 24: Create a new order

When logging in with chef account (lower privileges of admin and clerk), chef can only **change** the status of an order as "Prepare" or "Complete" by clicking on the status button at the end of the tab.

ID	Phone Number	Address	Total purchased	Payment method	Created At	Status
301	0123987465	asdfgh	35000 ₫		13-12-2021	<button>Prepare</button>
300	0987654321	abcdef	86000 ₫		13-12-2021	<button>Prepare</button>
299	9876543210	123 abcd	50000 ₫		12-12-2021	<button>Complete</button>

Figure 25: Order tab with status buttons for chef to update the order's progress

## 5 FEATURE DETAILS

### 5.1 Web Order

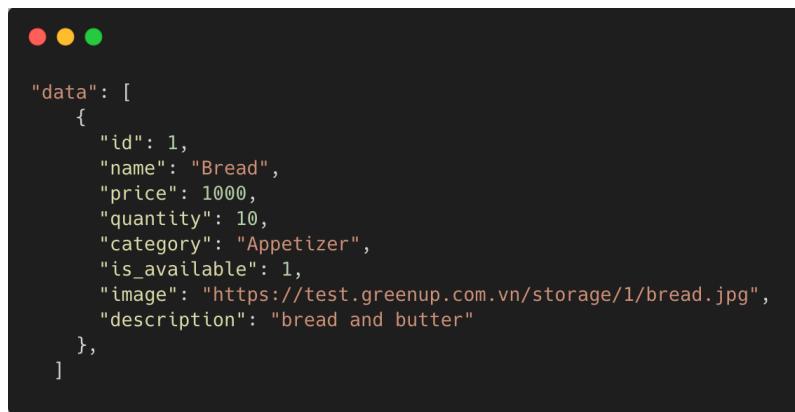
#### 5.1.1 Categorize Product

To give our customers a quicker view of the food categories, we sort the products by product category. These product categories will be managed on the CRM side.

- **How do we fetch products on Web Order?**

Any product that is displayed on Web Order is GET via API: <https://test.greenup.com.vn/api/menu>.

When calling this API, the server will return an array of products in the following format:



```
"data": [ { "id": 1, "name": "Bread", "price": 1000, "quantity": 10, "category": "Appetizer", "is_available": 1, "image": "https://test.greenup.com.vn/storage/1/bread.jpg", "description": "bread and butter" } ]
```

Figure 26: API menu

#### 5.1.2 Search Product

The product search box supports two different asynchronous use-cases:

- **Load on open:** waits for the components to be interacted with to load product options for the user. It will show the progress status as long as the network request is pending. This will improve the user experience even more.
- **Search as you type:** every time the user enters a symbol in the box, the system will give different suggestions about the name of the product

#### 5.1.3 Add to/Remove from Cart

- **What are the main criteria for a shopping cart?**

1. The shopping cart allows users to add and remove products without having to reload the page
2. Add and remove operations are displayed in the shopping cart and product details view
3. If the number of products in the cart is zero, the product is excluded from the cart
4. The maximum number of products allowed in the cart is equal to the number of products in stock (shown by the variable "quantity" in Figure 18)

- **How to make a shopping cart?**

Since the Shopping Cart is the main component of the website, the values in it will be reused a lot on other pages. For example, the Total Price is reused on the Checkout page, the details of the products in the Cart are also reused on the Review Order page. Therefore, when passing variables between pages, it is very easy to get confused and lead to incorrect logic. The solution to this problem is to use Redux to store the state of variables in the Global

where every page requires the variable to be used.

Here we use an initialization object, this object consists of an array (**cart[ ]**) of the products in the cart, the total price of all the products (**total**) in that cart, and the total price after the discount (**discount**). Details are as follows:

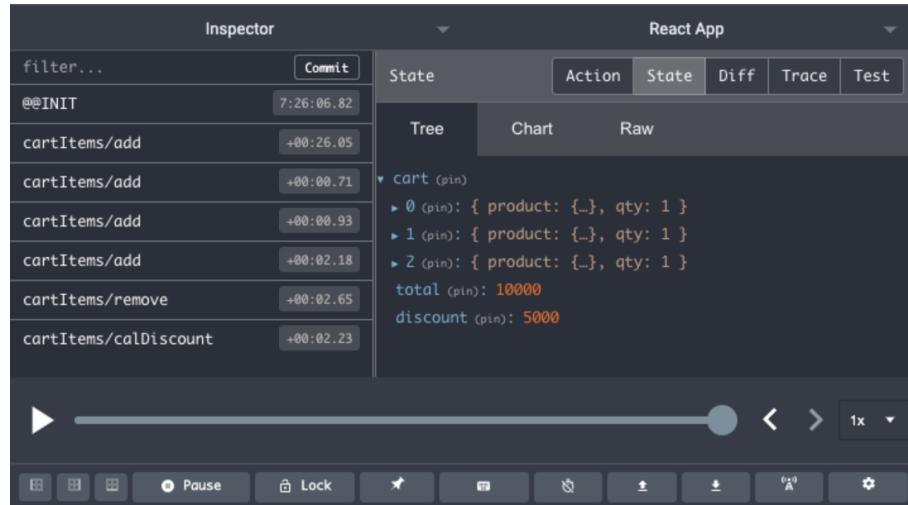


Figure 27: Redux Store for Adding, Removing Products

The variables in this object will be updated via the ACTION. When we need to perform an action, we just use dispatch to call the functions in ACTION (add / remove / calDiscount). In addition, in case the user uses the Voucher, the system will first send a request to the server to check whether it is valid or not. If it is correct, it will re-send an object of information about that Voucher and this **discount** variable will automatically deduct the discount from the cart.

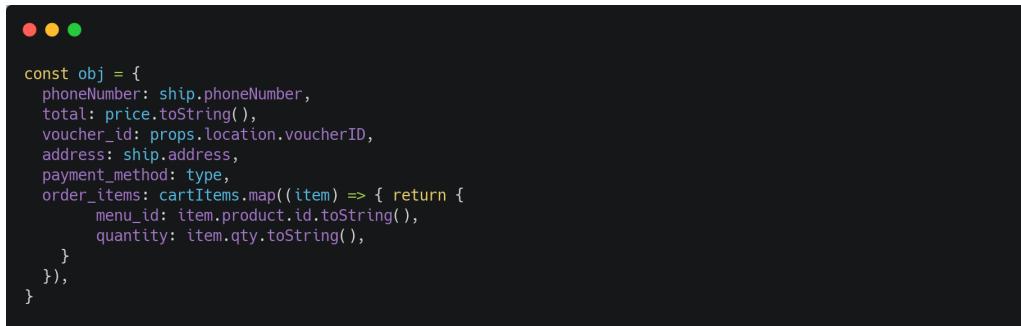
#### 5.1.4 Implementing Payment

- **How we integrated PayPal into our website?**

Integrating PayPal into our project is quite straight forward with the help of PayPal's React library. First we installed the needed component, created a developer account to get the API key and then generate some dummy transaction for testing purposes.

- **What happened after the click on the payment button?**

Once the payment button is clicked, the prices would first be validated on the PayPal side. Once the transaction same through, a request would then be sent to the server of the CRM system, which includes:



```
const obj = {
  phoneNumber: ship.phoneNumber,
  total: price.toString(),
  voucher_id: props.location.voucherID,
  address: ship.address,
  payment_method: type,
  order_items: cartItems.map((item) => { return {
    menu_id: item.product.id.toString(),
    quantity: item.qty.toString(),
  }},
),
}
```

Figure 28: Object Orders sent to CRM server after payment successful



- What are the advantages and disadvantages of integrating PayPal?

- **Advantages:** The integration is pretty simple.
- **Disadvantages:** One big problem with PayPal is how long it takes to render its' component. This led to a delay in loading the web page since the system has to load the PayPal's script first.

## 5.2 CRM Dashboard

### 5.2.1 Customer Management

- What are the key criteria for Client Management?

1. **Create:** create a new customer with fields such as "Full Name", "Phone Number" and "Address". The format of these fields is strings. The information will be stored in the database.
2. **Edit:** Allow to change the information created above and update it back into the database
3. **Delete:** Delete customer information from the database
4. **Filter:** Allows to find customer information based on the keywords Name and Phone Number of the customer
5. **Export:** Export all information of all customers in the database to a file in .xlsx format

In addition, customer information can also be generated through chatbots, which will be discussed in more detail in the Chatbot section in section 5.2.5.

### 5.2.2 Menu Management

We manage all the items displayed in the menu on our "Web Order" page using the Menu page in our CRM system.

- What are the key criteria for Menu Management?

1. **Create:** Creates a new product based on the following: "Name", "Description" (string data type), "Price" and "Quantity" (integer data type), "Group" (select data type data - dropdown), "Image" (upload product images to the server). Save this information in the database
2. **Edit:** Edit the product information created above and update it in the database.
3. **Delete:** Allows deleting products from the database. Note: a warning box must be given to the user before deleting, if the user confirms it, it will be deleted
4. **Filter:** Search for products based on product categories
5. **Availability Indicator:** The state of availability or not. If the product quantity is sold out, this status button will switch to off mode and vice versa. In addition, users can also switch to off mode if desired.

### 5.2.3 Order Management

The order page is used to handle all orders that have been placed on our system. It contains some major functions as follow:

- **Create:** Creating a new order requires the following information: customer name, Phone number, Address, Payment method (PayPal or Cash), Voucher (optional). For items, allow adding new products by adding "Add" button and removing products with "Delete" button
- **Edit:** Allows to change information about orders.



- **Delete:** Allows deleting each order from the database
- **Update status:** Each order has a button to display its status which will be discuss in detail later.

The system can also manages real-time resources. When create or edit an order, the system may check the condition of the menu's items and vouchers to handle the request correctly. For instance, if a selected item is sold out or an applied voucher is out-of-date, then the order can not be placed. Otherwise, the order will be created normally and the product's quantity will be decrease.

Every order will have a button to update their status located at the right corner of the Order tab. In this project, we use 4 states which are listed below to show the progress of preparing an order:

- **Waiting:** When a new order has been made, it appear with this state. The admin and clerk can click on this button to confirm the order.
- **Prepare:** After confirming the order, its status changes to "Prepare" and the chef can click on this to prepare the order.
- **Complete:** By clicking "Complete" status button, the chef claim that the order is completely prepare and ready to serve.
- **Completed:** After the chef choose the "Complete" status, that order comes to this final status on the order management system.

As for the status of the orders, the system will assign corresponding permissions to each role in the restaurant, which will be discussed in more detail later.

#### 5.2.4 Voucher Management

The voucher tab is used to generate and manage vouchers that can be redeemed for discount on our site. It has the same fundamental features as the other tabs on CRM dashboard, such as create, edit and delete. On the list of voucher, each of them also has a switch in order to control their availability just by a click on it.

A voucher consist of some major fields:

- The ID used to redeem for a discount.
- The name.
- The type divided into "Amount" and "Percentage".
- The deduction amount.
- The remaining fields namely the start-end date, the availability, the number of released and used voucher are used to check for the its validation.

When a voucher is applied to an order, the system first validates it and then reduces the deduct amount to the total price of that order.

In the edit view, the admin can send voucher to all registered customer via chatbot on Facebook by clicking the "Send Voucher" button at the end of the edit form.

#### 5.2.5 Messenger Bot

Messenger bot is used to interact and maintain presence with customers, make the purchase experience more pleasant.

- How to interact with chatbots?

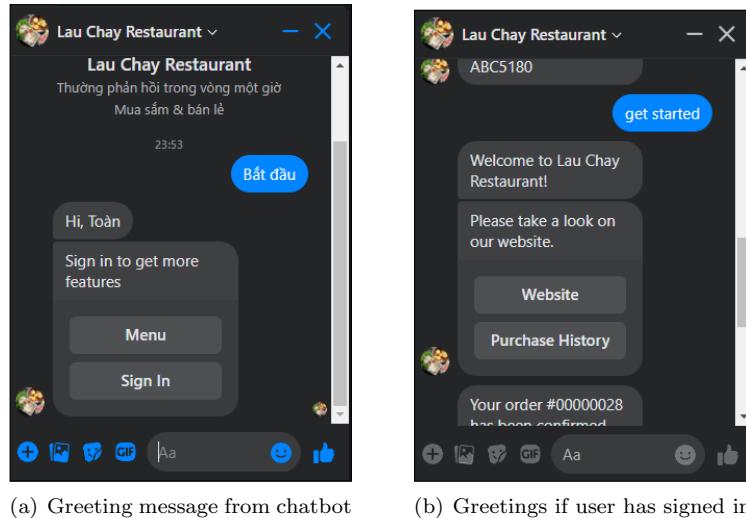


Figure 29: Make greeting with customer

- How to invite customers to register as a member?

When user click to Sign In button, browser will open new page to CRM system with a sign-in form.

Notice on the url, there is a psid parameter. This is the ID of user whom click to the button. This ID will be later attach with user in database

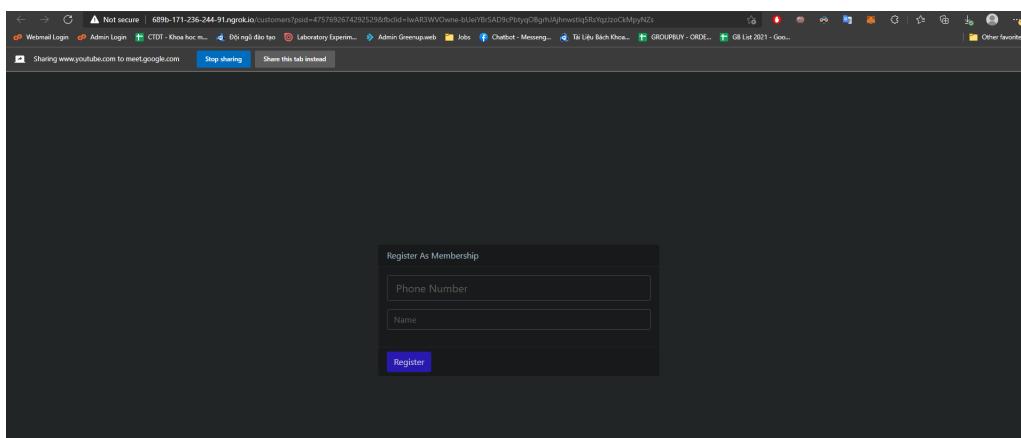


Figure 30: Membership Register

- What are the types of notifications that the chatbot will send to the customer?

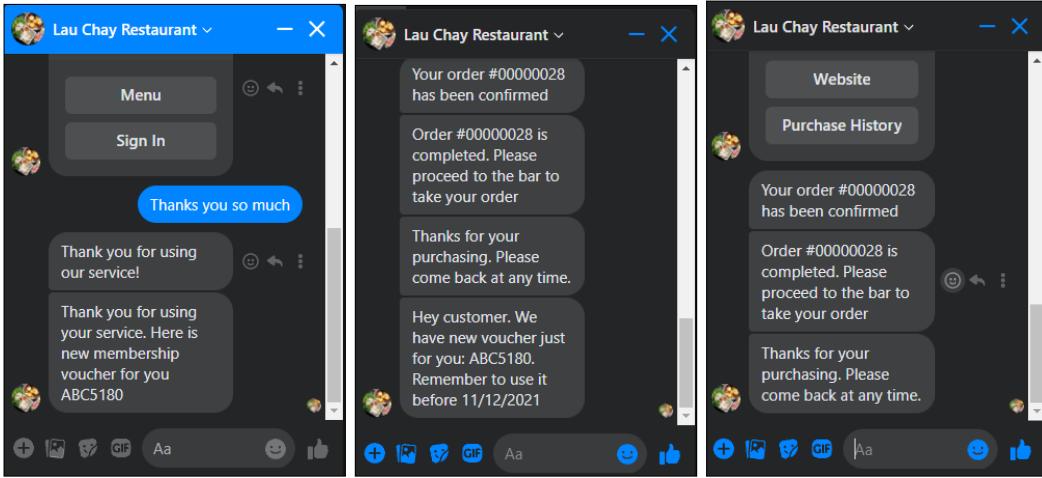


Figure 31: Make notifications

- **How to view purchase history and track orders?** (if user has register already)  
Users only need to enter the phone number they have registered in the box, the information will be displayed in order

ID	Shipping Address	Total	Created At	Status
#00000027	20202	5000\$	2021-12-02 08:58:08	Completed
#00000028	20202	10000\$	2021-12-02 09:01:38	Completed

Figure 32: Check purchase history through phone number

#### 5.2.6 Privileges

There are 3 types of account that available in our system: admin, chef, clerk. For demonstration purpose only, all of them are generated at the beginning of database initialization. You can access the account with information below:

- **Admin :** Full control of the CRM system , including all the functions discussed above (Menu, Voucher, Order, Customer).  
Account: admin@admin.com | admin123
- **Chef :** Restrict permissions, only allowing "Preparing" and "Complete" status confirmations for each incoming order. In addition, other functions cannot be operated  
Account: chef@chef.com | chef123
- **Clerk :** Restrict permissions, only allow confirmation of order statuses (all 4 states above) and allow creation of new orders  
Account: clerk@clerk.com | clerk123



```
1 Route::group(['prefix' => 'admin','as' => 'admin.'],'middleware' => 'check_role:admin'], function(){
2     Route::get('dashboard', 'DashboardController@index')->name('dashboard');
3     Route::get('export/customers', 'ExportController@export')->name('export.customers');
4
5     Route::resource('customers', CustomerController::class);
6     Route::resource('menu', MenuController::class);
7     Route::resource('orders', OrderController::class);
8     Route::resource('vouchers', VoucherController::class);
9
10    //Switch status of orders
11    Route::post('orders/elevate/{id}', 'OrderController@elevateStatus')->name('orders.elevateStatus');
12
13    //Send voucher to users
14    Route::post('vouchers/send/{voucher}', 'VoucherController@sendUsers')->name('vouchers.sendUsers');
15
16    //Update image
17    Route::post('image/{menu_id}', 'ImageController@update')->name('menu.imageUpload');
18    Route::delete('image/{menu_id}', 'ImageController@destroy')->name('menu.imageDestroy');
19 });
20
21 Route::group(['prefix' => 'chef','as' => 'chef.'],'middleware' => 'check_role:chef'], function(){
22     Route::get('orders', 'OrderController@indexAsChef')->name('orders.index');
23     Route::post('orders/elevate/{id}', 'OrderController@confirmAsChef')->name('orders.confirmAsChef');
24
25 });
26
27 Route::group(['prefix' => 'clerk','as' => 'clerk.'],'middleware' => 'check_role:clerk'], function(){
28     Route::get('orders', 'OrderController@indexAsClerk')->name('orders.index');
29     Route::get('orders/create', 'OrderController@createAsClerk')->name('orders.create');
30     Route::get('orders/{order}', 'OrderController@edit')->name('orders.edit');
31     Route::post('orders/elevate/{id}', 'OrderController@confirmAsClerk')->name('orders.confirmAsClerk');
32 });
33
34
35 Route::get('customers', 'Customer\CustomerController@create')->name('customers.register');
36 Route::post('customers', 'Customer\CustomerController@store')->name('customers.store');
37 Route::get('history/purchase', 'Customer\HistoryController@index')->name('customers.purchase');
```

Figure 33: Routing in web.php

Route::group() function is used to group routes into groups. In the first group(line 1), we named it as "admin", with prefix = 'admin.', therefore route name inside will start with 'admin.'. For example, admin.customers.create. Before access the routes inside, we have to group through a middleware name "check\_role" with argument "admin", this make sure we have the admin role in order to proceed inside.

In line 21 and 27, there are clerk and chef route group. The middleware also check the respectively roles of users whether it matches the argument given before allow user to proceed inside. Clerk can index, create, edit orders and confirm order. Meanwhile, chef can only see the order and confirm them.

Due to the interference of check\_role middleware, user with different role can't access other role's route. For example, a user with chef role can't access to the route admin.export.customers from admin group. User will be redirected home page.



The screenshot shows a web browser window with the URL <https://test.greenup.com.vn/chef/orders>. The page title is "Home". On the left, there's a sidebar with "Dashboard" and "MANAGE ORDERS" sections, and a "Orders" button. The main content area displays a table of orders:

ID	Phone Number	Address	Total purchased	Payment method	Created At	Status
300	0987654321	abcdef	86000 ₫	Bank	13-12-2021	Complete
301	0123987465	asdfgh	35000 ₫	Bank	13-12-2021	Prepare

Figure 34: Chef account

The screenshot shows a web browser window with the URL <https://test.greenup.com.vn/home>. The page title is "Home". On the left, there's a sidebar with "Dashboard" and "MANAGE ORDERS" sections, and a "Orders" button. The main content area has a "Dashboard" header and a message: "You are logged in!"

Figure 35: Get redirect after attempted to access /admin/customers/export



## 6 Disclaimer

Our products are hosted on Git under the following links:

- **For Web Order:** [Github](#)
- **For CRM Dashboard:** [Github](#)

We use 2 main platforms to deploy products to:

- **For Web Order:** [Client UI](#)
- **For CRM Dashboard:** [Manager UI](#)  
Login account: admin@admin.com | admin123

Our CRM system is entirely dependent on Google Cloud Platform. We are only given 3 months trial and it may expires before you can take a look. This same may happen to our Web Order and chatbot. Please let us know if you experience any trouble related to server.  
Contact me at [toan.pham.79125@hcmut.edu.vn](mailto:toan.pham.79125@hcmut.edu.vn)