LIKELION VIETNAM

PROJECT 1

RESTAURANT MANAGEMENT ON SERVER-SIDE

WITH SPRING BOOT AND THYMELEAF

By

BARCELONA TEAM

Submitted to: LIKELION VIETNAM

February 2023

Project 1: https://github.com/LocDuong112/FoodShinhanDS.git
Project 2: https://github.com/Lucky-Star-999/food-service-v1.git

ACKNOWLEDGMENTS

Table of Contents

I. Project Schedule

Week	Task
1	Training
	Team meeting with each other
2	Training
	Setting up project's environment
3	Training
	Come up with project requirements
4	Training
	Design the system
5-7	Training
	Teamwork and finish the project

II. Project Description

1. Description

"Restaurant management" is a tool for helping online ordering for a restaurant.

The technologies that "Restaurant management" used are:

- Spring Boot

- OracleDB

- Github

The project will eventually be turned into a microservice project, and the team plans to launch it using Docker.

2. Requirements

Admin

+ Account management

+ CRUD on accounts

Restaurant manager
+ Show list of available orders
+ Create new order
+ Change shipping status of order
+ Show bill of order
+ Paying
    + Payment confirmation (customer pay by cash)
    + Create QR code for paying
+ View sales, reports (monthly)
+ Menu management

+ CRUD on food of the restaurant

User
+ Login/logout
  + Create new account
  + Sign in with Google account
  + Reset password
  + Update account
+ Show trending food on homepage
+ Sort food by category
+ Add food to cart
+ Show cart
  + Show bill
  + Update food quantity
  + Delete food
+ Show order status
+ Create feedback on food

3. Tasks

a. Dương Minh Lộc

Sign in, sign up (don't need JWT yet)
  INPUT: username, password
  OUTPUT: Login notification (success?), account dto
Reset password, modified account information
  INPUT: account dto, @PathVariable accountId
  OUTPUT: new account dto
Customer see all products (5-10 products each load)
  Use function to check for customer role
  See all products (load 5-10)
    1. INPUT: last product id
    2. OUTPUT: 5-10 next products

b. Lê Hoàng Nhân
Admin manage account
  Function to check for admin role
  CREATE
    INPUT: account dto

OUTPUT:none

READ

+ List of account
    + INPUT: none
    + OUTPUT: all account in database with and without filter, pagination
+ Individual account
    + INPUT: accountId
    + OUTPUT: an account dto
+ Template account for creation

UPDATE

INPUT: account dto

OUTPUT: none

DELETE

INPUT: accountId

OUTPUT: none

c. Phan Hồng Hà

Manager sees all orders

Function to check for manager role

INPUT: none

OUTPUT: All order dto in db

Manager create new order

Use function above to check for manager role

INPUT: order dto

OUTPUT: new order dto

Customer sees cart

Delete product from cart

Use function to check for customer role

INPUT: product id, cart id

OUTPUT: All product dto in cart

d. Võ Văn Đăng Khoa

Manager change status of order

Use function to check for manager role

INPUT: order dto, @PathVariable order id

OUTPUT: new order dto

Manager sees bill of order

Use function to check for manager role

INPUT: @PathVariable order id

OUTPUT: order dto with bill dto (create new dto for this)

Customer sees order status

Use function to check for customer role

INPUT: account id

OUTPUT: All order dto with account role as customer

e. Nguyễn Hoàng Linh

Manager search for product

Use function to check for manager role

INPUT: @Param any String

OUTPUT: All product dto match product name

Customer see 5-10 most buy products

Function to check for customer role

INPUT: none

OUTPUT: 5-10 most buy product

Customer search for product

Function to check for customer role

INPUT: @Param any String

OUTPUT: All product dto match product name

f. Đỗ Việt Bách

Customer add product to cart

Use function to check for customer role

INPUT: account id, product id, product quantity

OUTPUT: All product dto in cart

Modify quantity of product in cart

Use function to check for customer role

INPUT: product id, product quantity, cart id

OUTPUT: All product dto in cart

Add loyalty point to customer after having an order (order price / 1000)

Use function to check for customer role

INPUT: order id

OUTPUT: account dto

5. Finished API

Restaurant management			
No.	**API**	**Function**	**Contributor**

1	GET /admin/viewAllAccount
	Request param:
	   + account_type
	   + page
		Return a model view for a list of accounts (include info of the total page number and current page, view all types of accounts by default)	Nhân

2	GET /admin/takeAllAccount
	Request param:
	   + account_type
	   + page
		Return a list of accounts only	

3	GET /admin/IndividualAccount Request param: + account_id	Return information of an individual account	
4	PUT /admin/IndividualAccount Request body: + AdminEditAccountDto	Check if old information is changed and update the account, then redirect to the previous page	
5	DELETE /admin/IndividualAccount Request param: + account_id	Soft delete 1 account and redirect to the previous page	
6	GET /admin/templateIndividualAccount	Return dummy information of an individual account	
7	POST /admin/IndividualAccount Request body: + AdminCreateAccountDto	Create a new account and redirect to the previous page	
8	PUT /api/updateAccount/{id} Path variable: id Request body: Account DTO	Update account	Lộc
9	GET /api/signIn	Sign in	

	Request param: + email + password		
10	POST /api/signUp/{role} Path variable: role Request body: Account DTO	Sign up	
11	GET /api/allProductsCustomer Request param: + lastProductsId + fectNum	Get all products with customer role	
12	GET /api/product/manager/{name} GET /api/product/manager/ GET /api/product/manager (branch linh-branch-1)	Manager search for product	Linh
13	GET /api/product/customer/{name} GET /api/product/customer/ GET /api/product/customer (branch linh-branch-1)	Customer search for product	
14	GET /api/product/customer/most-buy-product (branch linh-branch-1)	Customer see 5-10 most buy products	
15	GET /bill/order/{orderId} PathVariable: orderId	Get bill of order	Khoa
16	POST /bill/addBill RequestBody: billDto	Get all bill	
17	PUT /order/status/{id PathVariable: id RequestBody: orderDto	Update to change status of order	
18	GET/api/orders RequestBody: orderDto	Get all orders in database	Hà
19	GET/api/carts RequestBody: cartDto	Get all carts in database	
20	POST/order/createOrder RequestBody: orderDto	Create order	
21	DELETE/cart/product/{id} PathVariable: cart id, product id RequestBody: cartDto	Delete product in cart	
22	Connect database and auto generate table		

Official project (Food Service V1)

No.	API	Function	Contributor
1	GET /api/admin	Admin homepage	Nhân
2	GET /api/admin/admins	Manage admins' account page	
3	GET /api/admin/customers	Manage customers' account page	
4	GET /api/admin/restaurants	Manage restaurants' account page	
5	GET /api/admin/create-admin	Create admin page	
6	GET /api/admin/update-admin/{email}	Admin information page	
7	GET /api/admin/create-customer	Create customer page	
8	GET /api/admin/update-customer/{email}	Customer information page	
9	GET /api/admin/create-restaurant	Create restaurant page	Bách
10	GET /api/admin/update-restaurant/{username}	Restaurant information page	
11	GET /api/admin/login	Admin login page	
12	POST /api/admin/admin	Create new admin	
13	POST /api/admin/customer	Create new customer	
14	POST /api/admin/restaurant	Create new restaurant	Hà
15	PUT /api/admin/admin	Update admin information	Bách
16	PUT /api/admin/customer	Update customer	Hà

		information	
17	PUT /api/admin/restaurant	Update restaurant information	
18	DELETE /api/admin/admin/{email}	Delete admin	Bách
19	DELETE /api/admin/customer/{email}	Delete customer	
20	DELETE /api/admin/restaurant/{username}	Delete restaurant	Hà
21	POST api/admin/login	Admin login	
22	GET /api/admin/logout	Admin logout	Bách
23	GET /api/customer	Customer homepage	
24	GET /api/customer/login	Customer login page	
25	GET /api/customer/restaurant/{username}	Menu of restaurant for customer page	
26	GET /api/customer/register	Customer register page	
27	GET /api/customer/profile	Customer profile page	
28	GET /api/customer/orders	Page for manage order for customer	Lộc
30	GET /api/customer/order-confirm/{orderId}	Page for order approval for customer	
31	GET /api/customer/order/{orderId}	Page for order detail for customer	
32	GET /api/customer/chat	Page for AI chat for customer	
33	POST /api/customer/customer	Create new customer	Hà
34	POST /api/customer/order	Create new order	
35	POST /api/customer/chat/request	Send message to AI chat	Linh
36	PUT /api/customer/customer	Update customer information	Bách
37	PUT /api/customer/customer/{email}	Delete customer	

38	POST /api/customer/login	Login for customer	Hà
39	GET /api/customer/logout	Logout for customer	Khoa
40	GET /api/restaurant	Restaurant homepage	
41	GET /api/restaurant/login	Restaurant login page	
42	GET /api/restaurant/register	Restaurant register page	
43	GET /api/restaurant/profile	Restaurant profile page	
44	GET /api/restaurant/create-food	Page to add new dish for restaurant	
45	GET /api/restaurant/update-food/{id}	Page for update dish for restaurant	
46	GET /api/restaurant/orders	Page for manage orders for restaurant	
47	GET /api/restaurant/order/{orderId}	Page for order detail for restaurant	
48	GET /api/restaurant/chat	Page for AI chat for restaurant	Linh
49	POST /api/restaurant/restaurant	Create new restaurant	
50	POST /api/restaurant/food	Create new dish for restaurant	
51	POST /api/restaurant/chat/request	Restaurant send message to AI	
52	PUT /api/restaurant/restaurant	Update restaurant information	
53	PUT /api/restaurant/food	Update dish detail for restaurant	
54	DELETE api/restaurant/restaurant/{username}	Delete restaurant	
55	POST api/restaurant/login	Login for restaurant	
56	GET /api/restaurant/logout	Logout for restaurant	

57		Multi language	
58		ChatGPT embedded	

III.   Result

Home page

First, we see the home page. The home page includes 3 roles: Admin, Restaurant,

Customer.
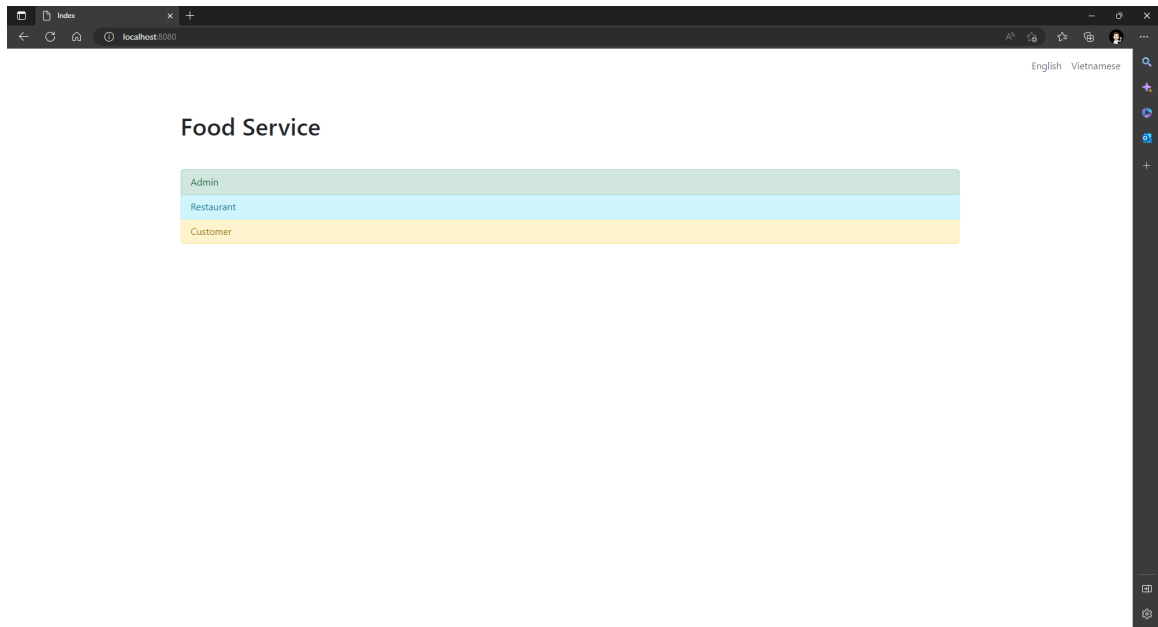
We will click the button corresponding to our role.

Admin site

In this project, we assume that the admin role has permission to manage all accounts (view accounts registered, update information, and create or delete an account).

When users sign in successfully as the admin role, the Admin - Home page will be displayed with three types of accounts that the system has: Admin, Customer, and Restaurant.

Admin Home page
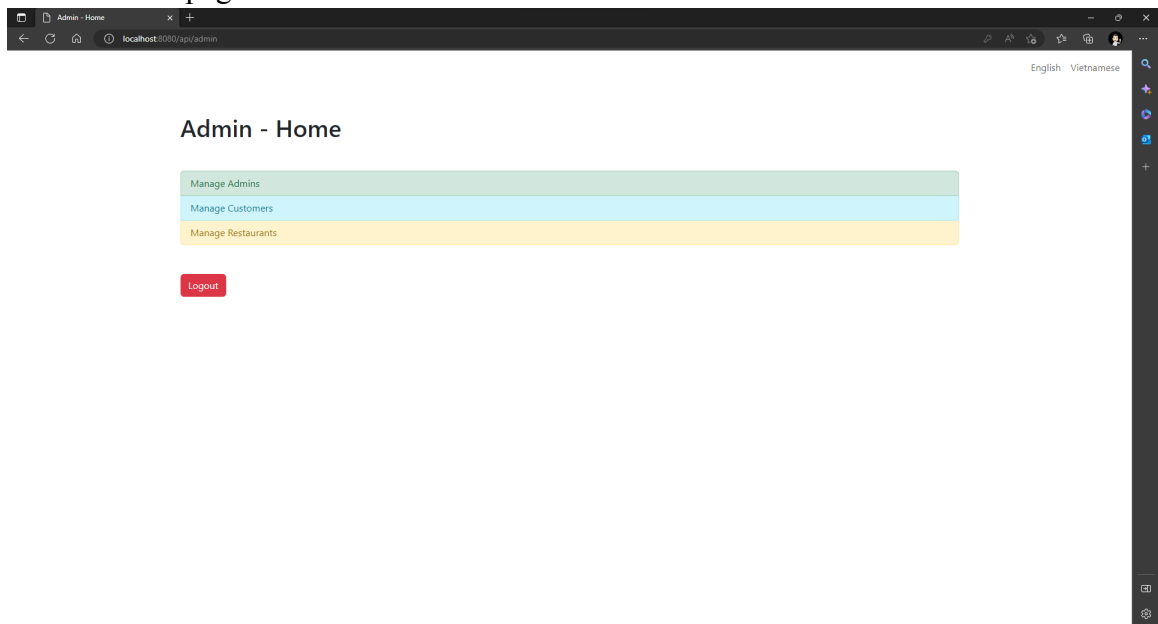
View all Admin accounts
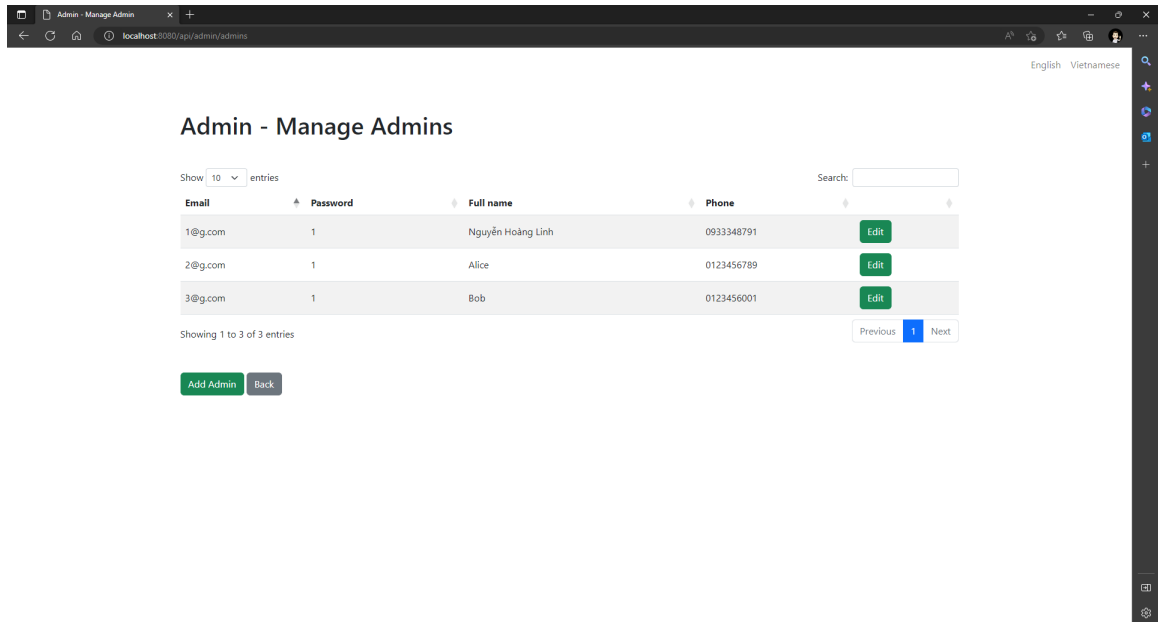
View details and edit or delete an account
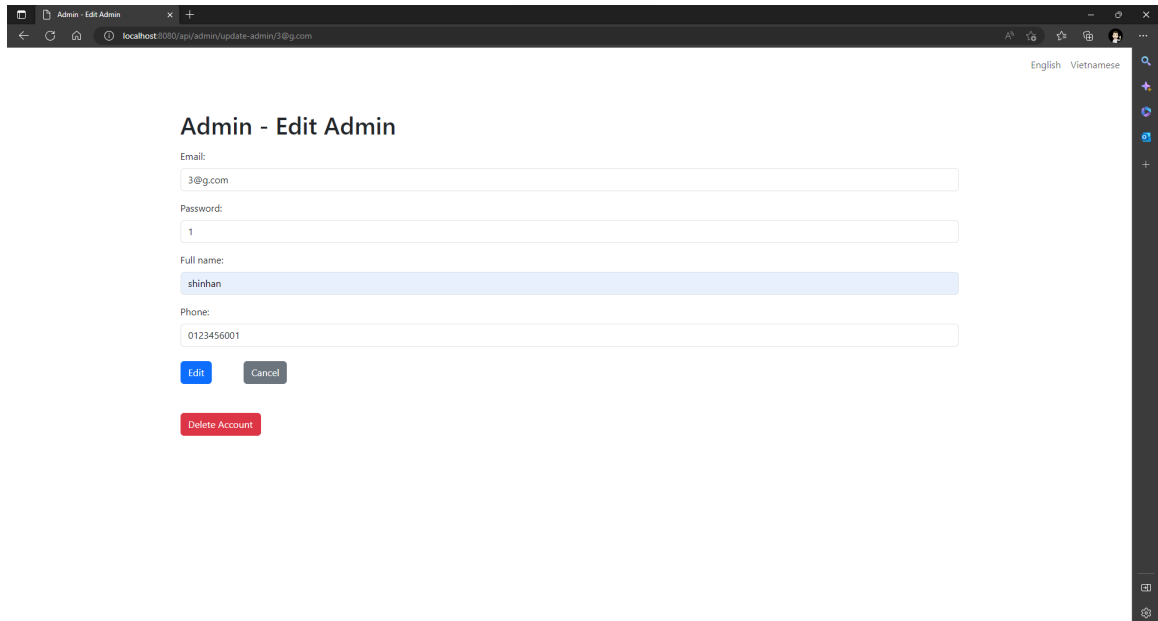
Create a new account as the admin role

The new account appears in the list after created
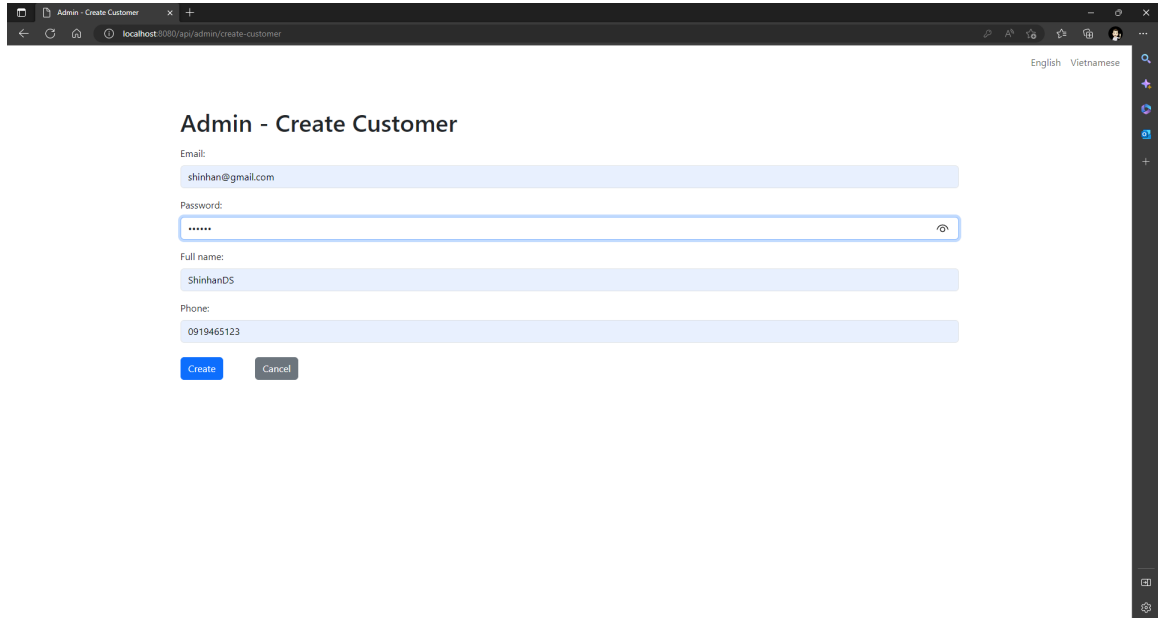
View all Customer accounts
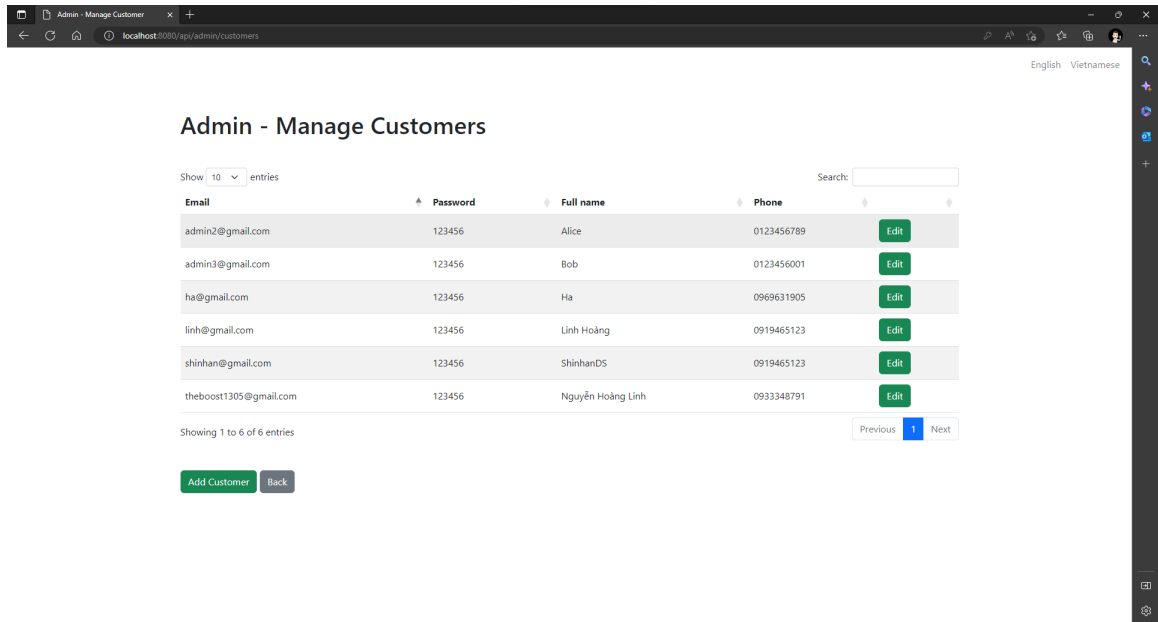
View details and edit or delete an account
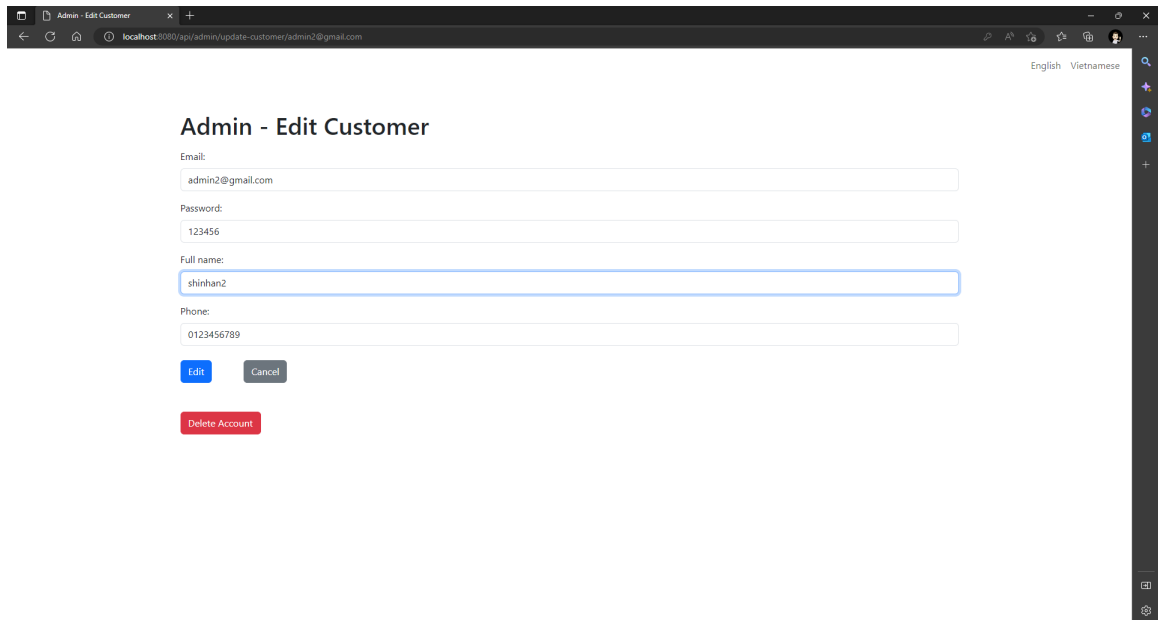
Create a new account as the customer role

View all Restaurent accounts

View details and edit or delete an account

Create a new account as the restaurant role
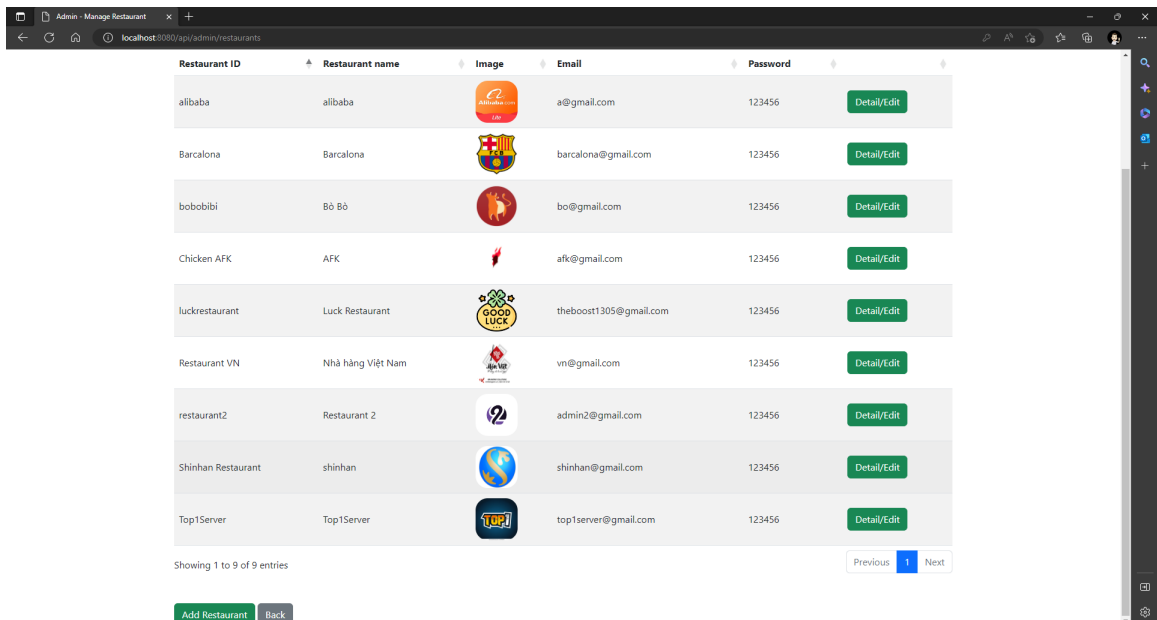
The new restaurant appears on the list after created

The admin site also included AI chat bot and multi language services to help better the experience of the admin
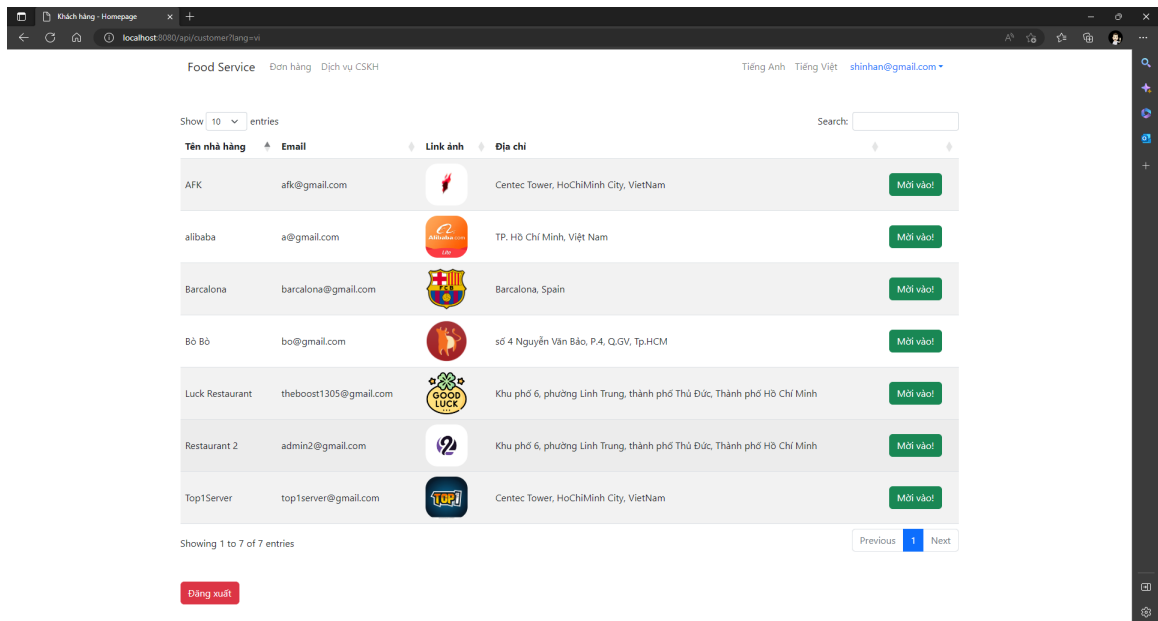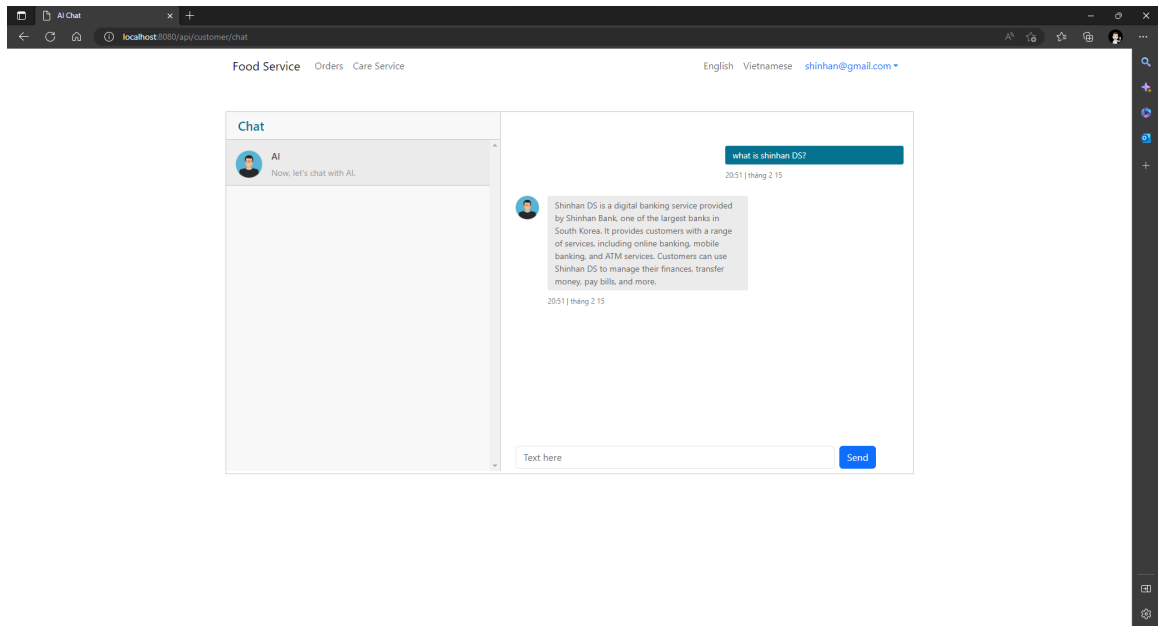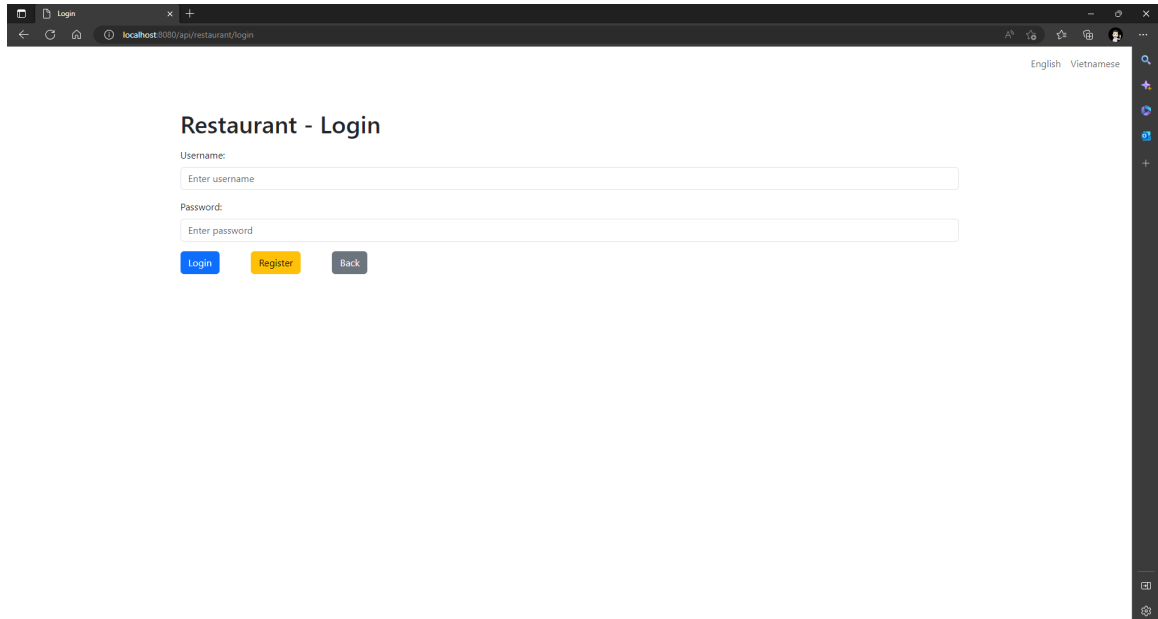
Restaurant site

First, we see the home page. We'll click the "Restaurant" button.
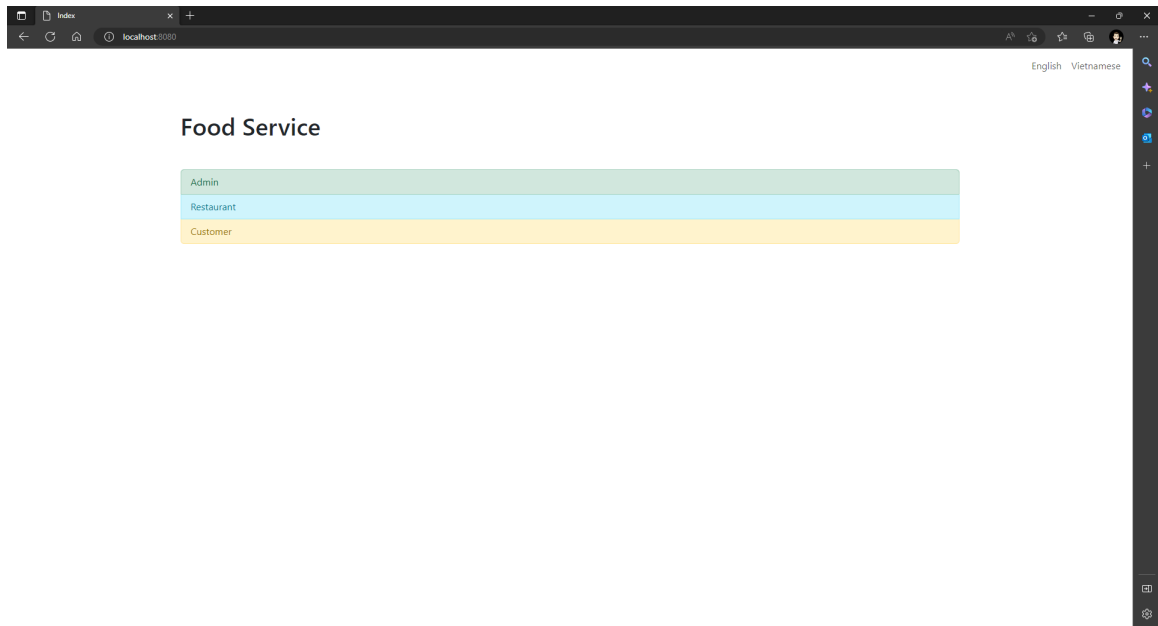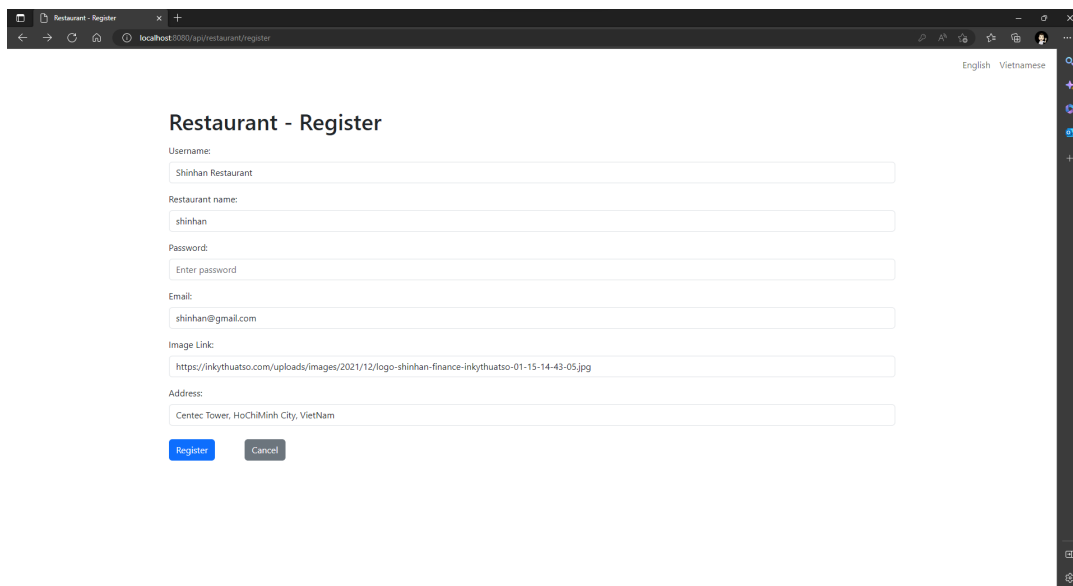
Clicking on Restaurant will display Restaurant's Login page.

Next, we choose "Register" to create Restaurant's account. Fill in the fields according to the correct title and click "Register" or "Cancel" to return to the Restaurant's Login page.

Back to the Restaurant's Login page, we enter the account we created earlier and click "Login" or "Back" to return to the home page.
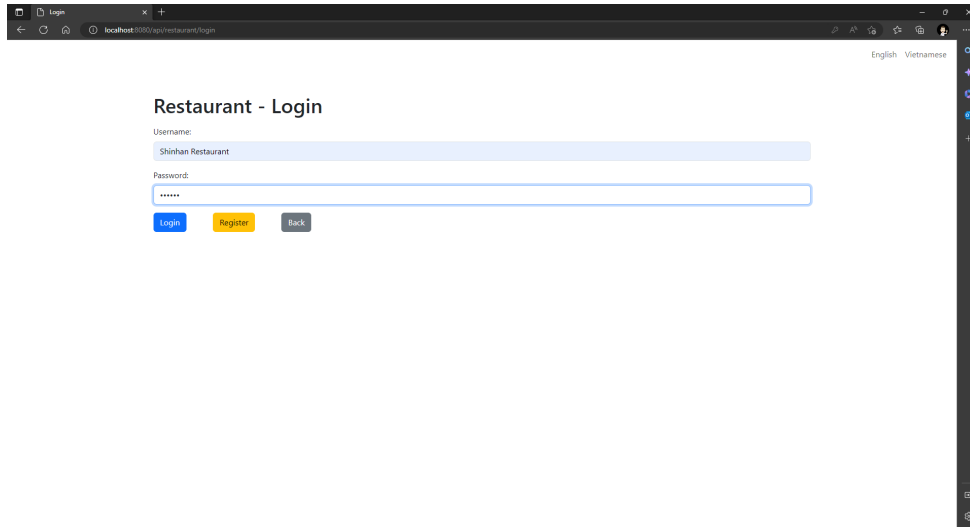
Then, the Website shows us the food list of Restaurant in the data.
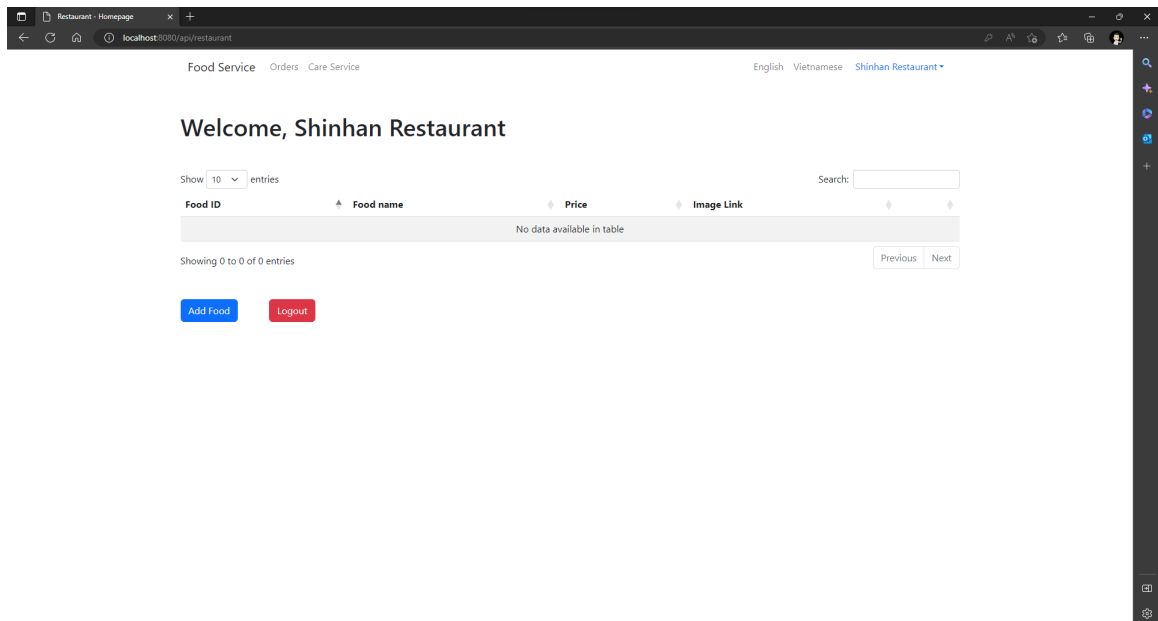
We can add food with the button "Add Food" or log out to return to the Restaurant's Login page with the button "Logout". If you click the button to add food, fill in the fields according to the correct title and click "Create" or "Cancel" to return to the food list or restaurant.
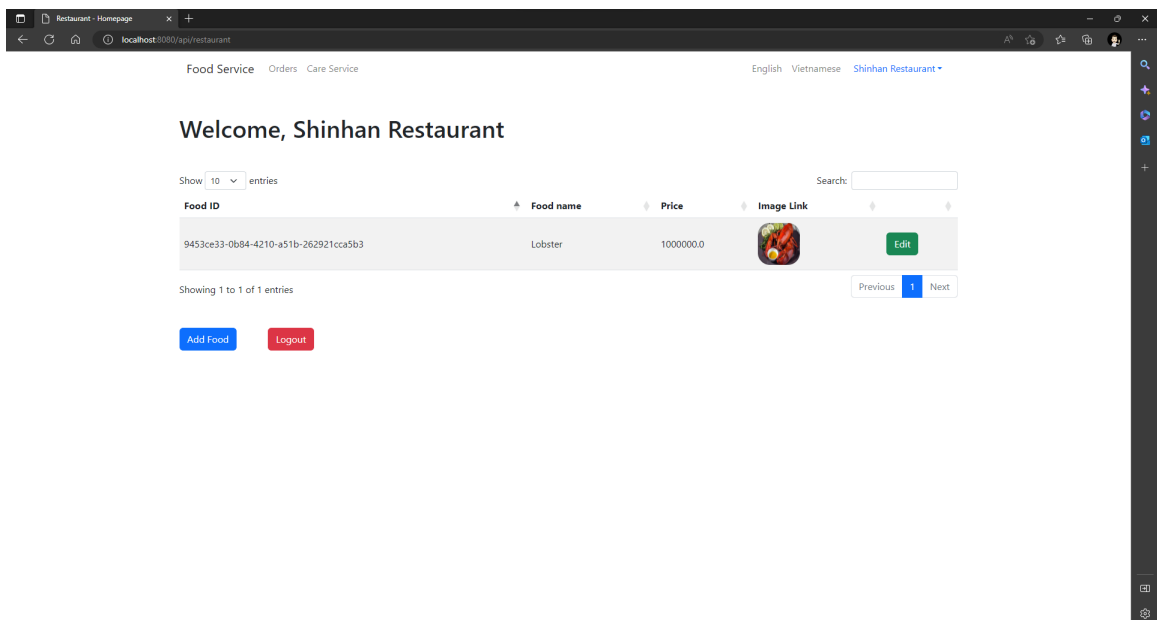
Here is the interface screen of the Restaurant's food list when food has been added.
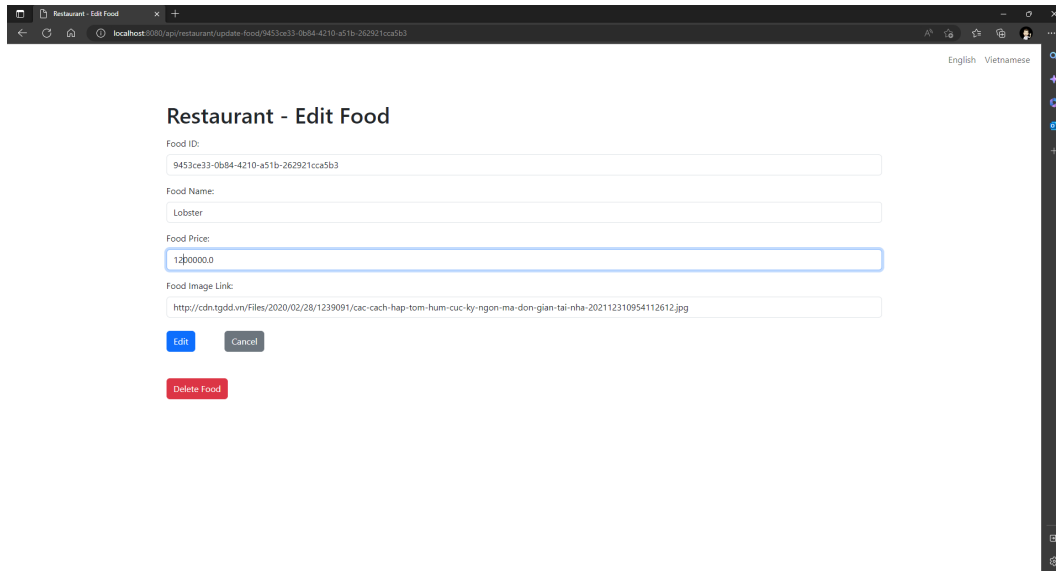
We can edit food with the button "Edit Food". Here, we can Edit or Delete Food. We need to fill in the formation if we want to edit food. Click the button "Cancel" to return to the restaurant's food list.

This is what it looks like after editing the lobster food price from 1.000.000 to 1.200.000. We can search by food name with Search bar or show the number of entries you want to see or move between pages corresponding to the buttons displayed on the interface screen.
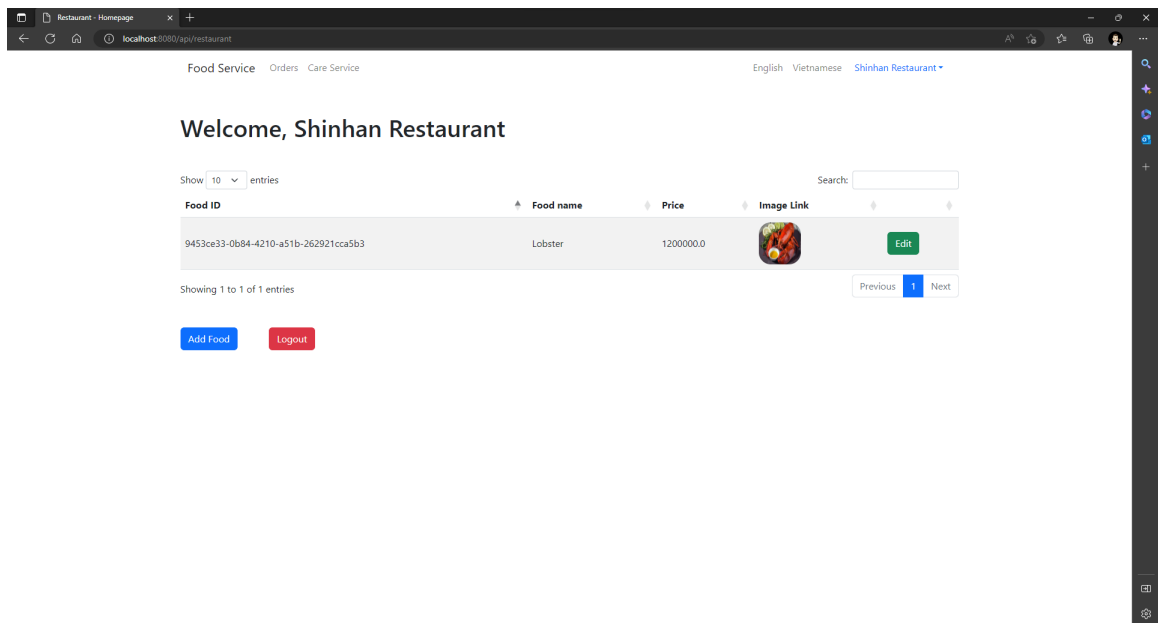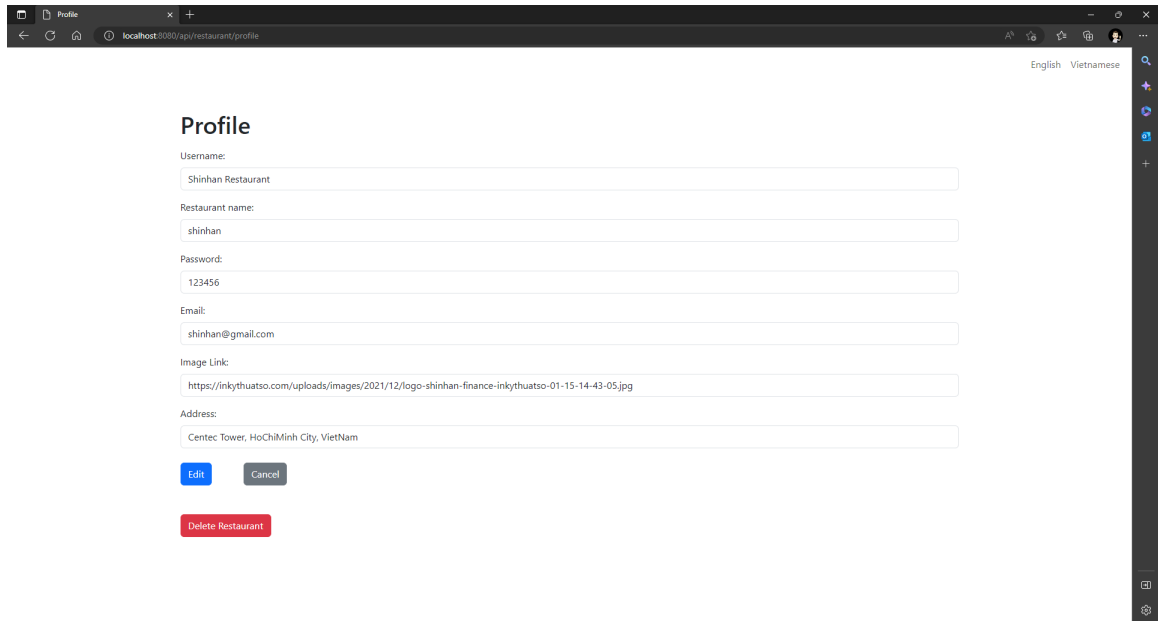
In the top right corner, we click on the Restaurant's profile, it will show the Profile page. Here, we can "Edit" or "Delete Restaurant". We need to fill in the form if we want to edit our profile. Click the button "Cancel" to return to the food list of Restaurant.

Still in the top right corner, we can click "Logout" to return to the Restaurant's Login page.
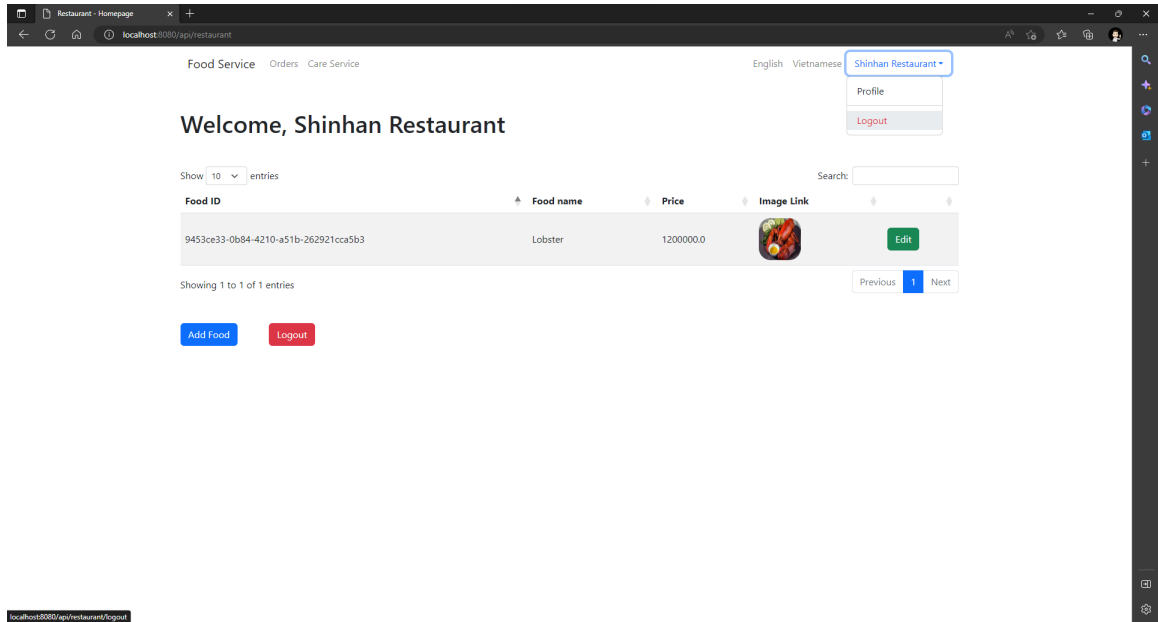
We can change language between English or Vietnamese in the top right corner. We can check the order of the restaurant's food list in the top left corner.
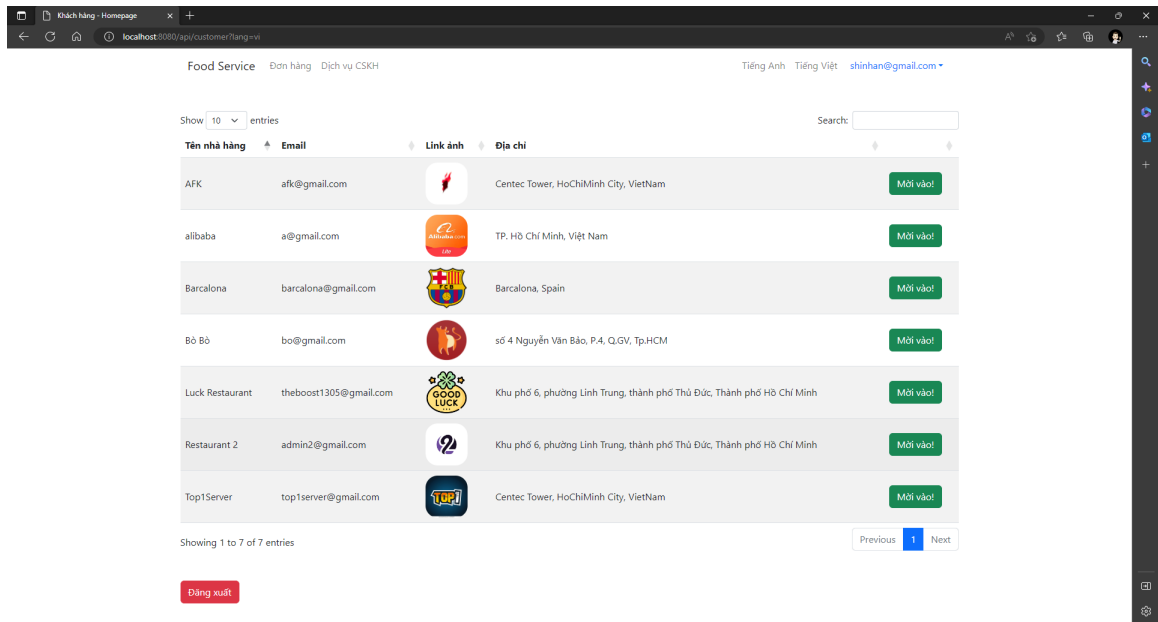
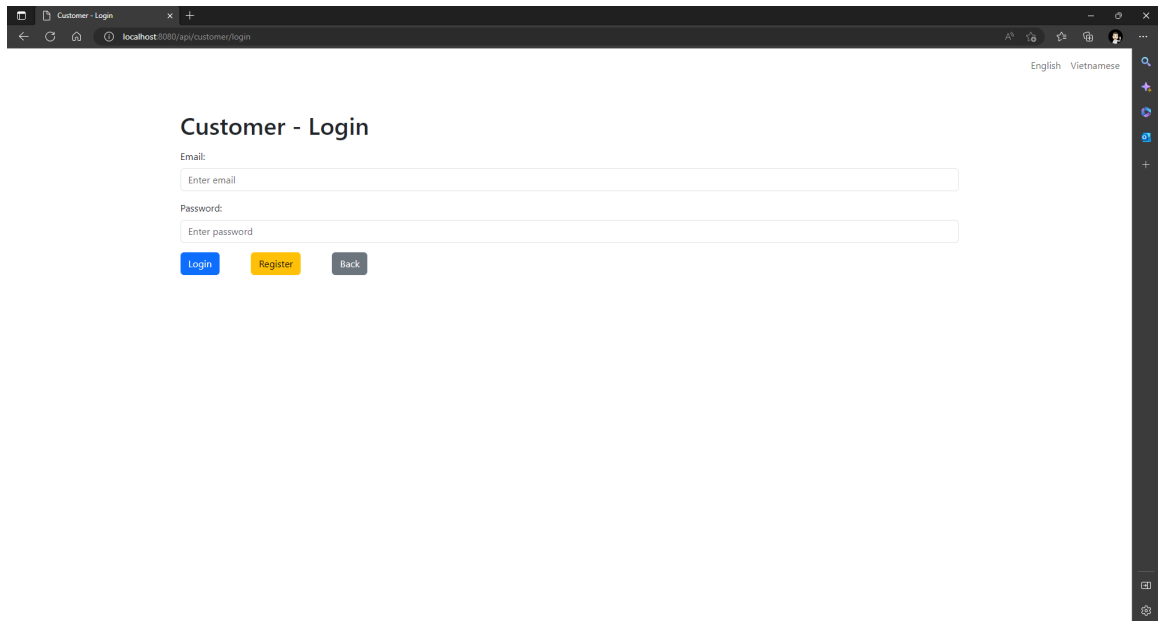Select the "Care Service" for chatting with AI Chatbot. Here, we can ask anything in the chat bar.
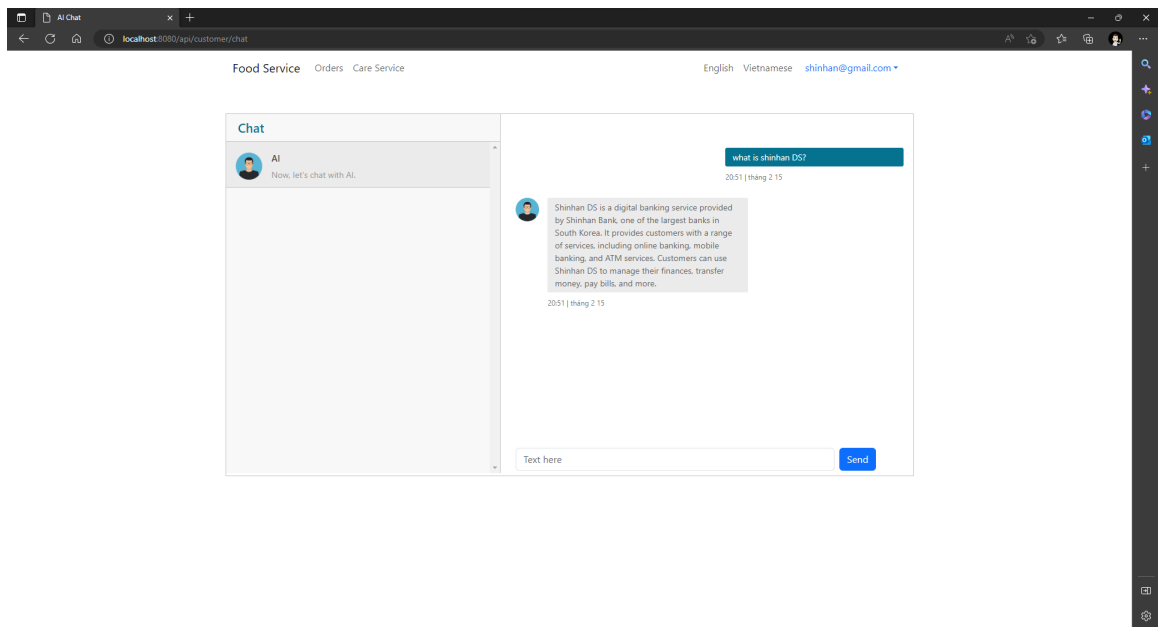
Customer site

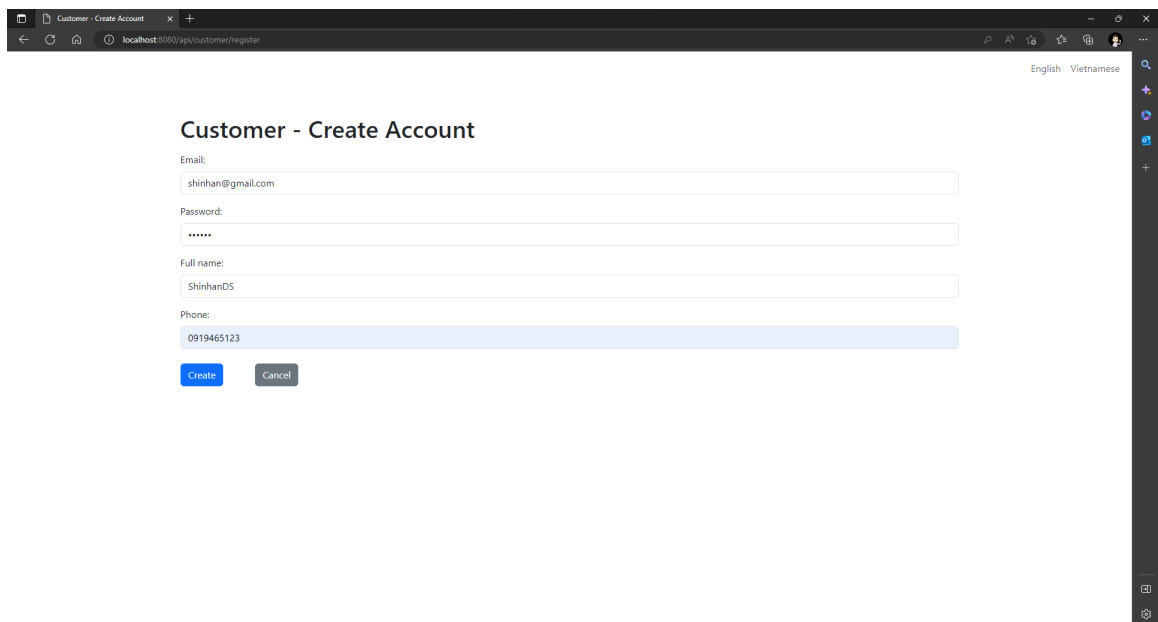First, we see the home page. The home page includes 3 roles: Admin, Restaurant, Customer.
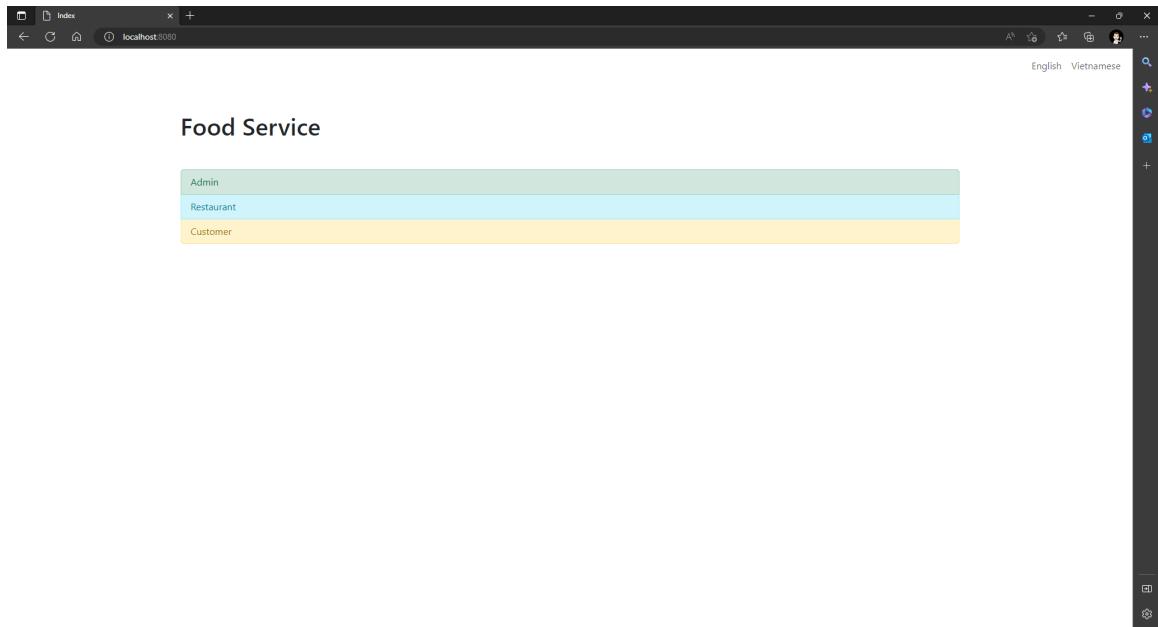
Clicking on Customer will display Customer's Login page.

Next, we choose "Register" to create an account. Fill in the fields according to the correct title and click "Create".

Back to the Customer's Login page, we enter the account we created earlier and click "Login".

Then, the Website shows us the list of restaurants already in the data.

Click "Visit" to see the restaurant's dishes.
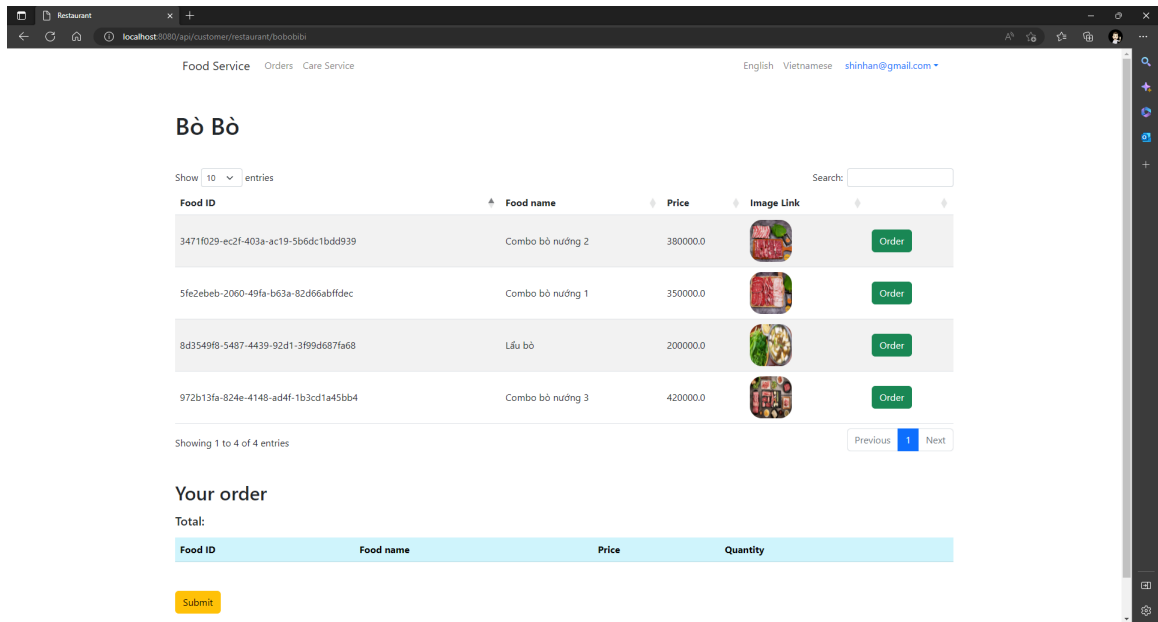
Then, we click "Order" to order the food, it will be put into Your Order.
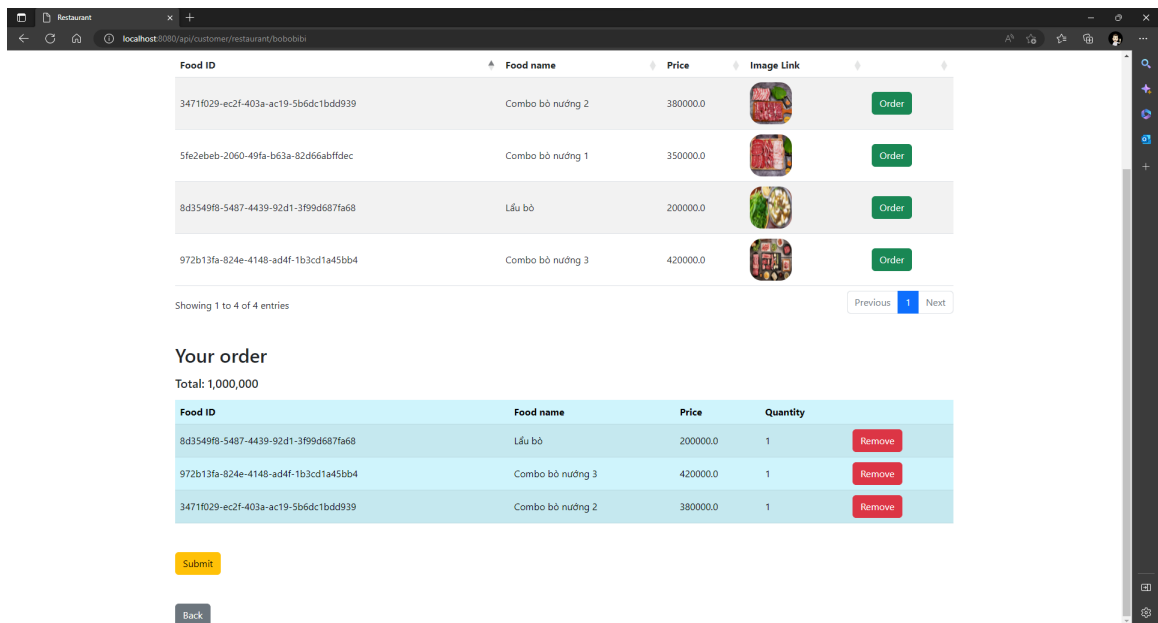
After clicking "Submit", the Website will display the Order Detail page, the dishes that you have ordered.
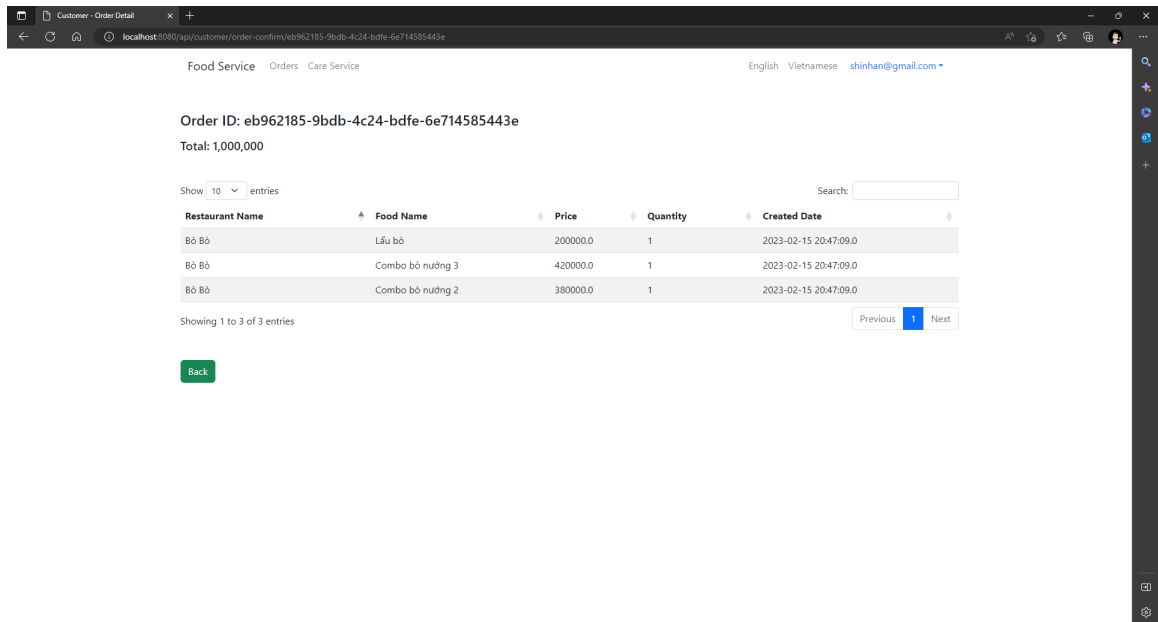
Next, the feature of viewing a Customer's profile.
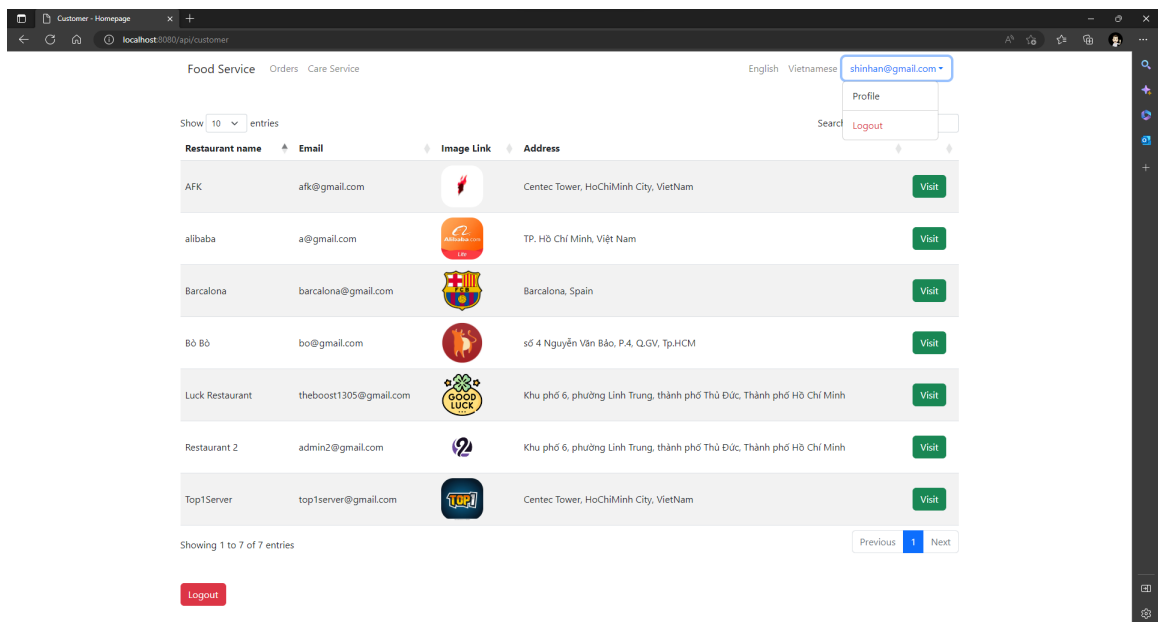
After clicking "Profile", Website displays Customer's profile page. Includes Customer information. You can edit those fields by clicking "Edit".

Finally, the Customer's Logout feature, when clicking "Logout". The Account will be logged out and the Website will return to the Login page.

In addition, other features such as multi-language
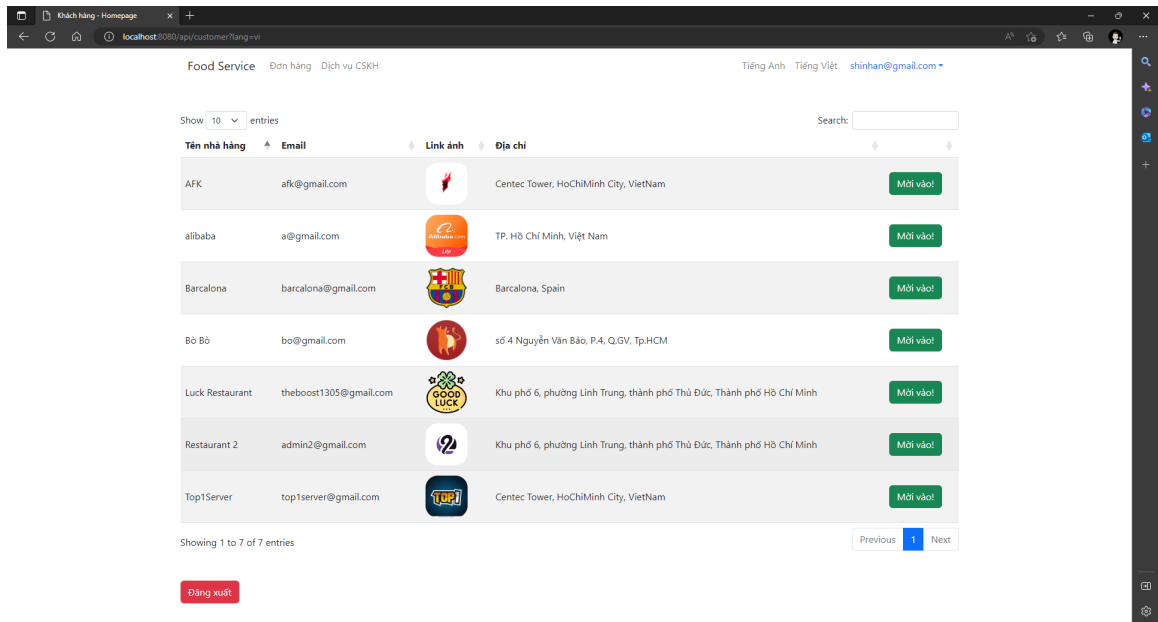
And AI Chatbot.
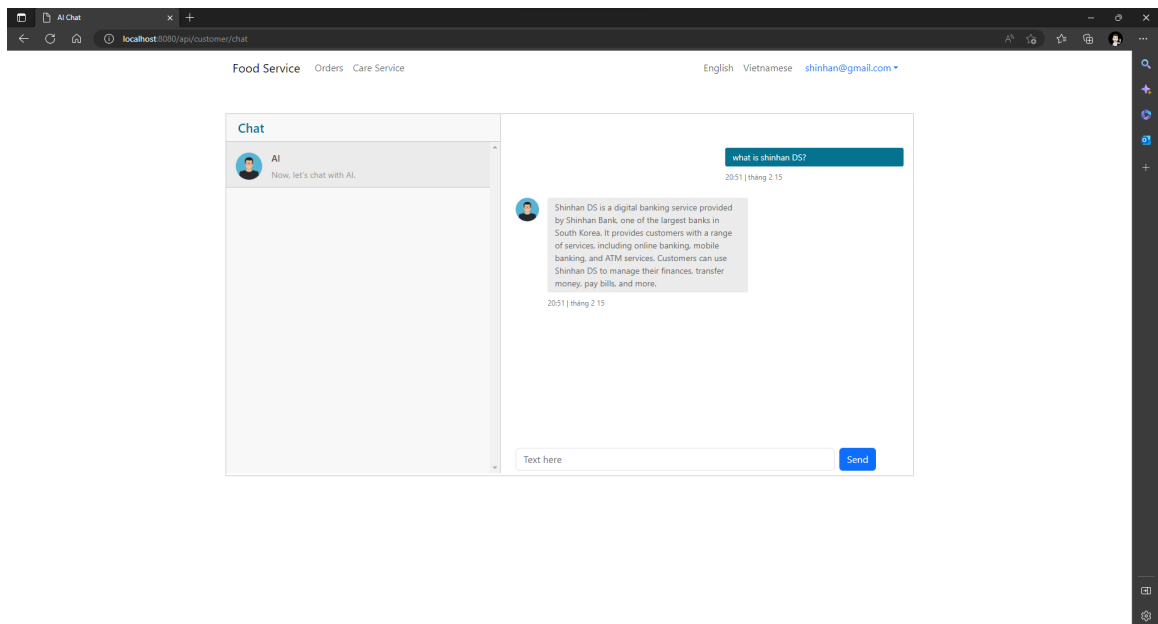
IV.   **Skills and Experiences**

Clean code: We put a lot of time and attention into learning how to write clean code and how to utilize code. Although the process initially took a lengthy time, it was tremendously beneficial for the project's upkeep and future expansion.

Communication: We had the chance to speak with mentors and other teammates face-to-face about issues when tasks were assigned during working on the project. We've learned a lot from listening to people describe how to process and identify information. We've also improved our communication and discussion skills.

Teamwork: Working successfully as a team is essential in the workplace. Although school projects have helped us strengthen our collaborative skills, the course still affords us more opportunity to do so. We particularly know how to listen and communicate with coworkers. Additionally, it is in charge of finishing the assignment on time in order to avoid interfering with the team's work. This will serve as a solid foundation for our future success.

Note Taking: It is challenging to recall all the details of a meeting that lasts for at least 30 minutes while working. As a result, taking good notes can help you retain the information and concentrate more throughout the meeting.

Task Estimation: While some jobs appear to be straightforward, they are not. In order for tasks to be approximated as accurately as possible, workers must have a

thorough understanding of their duties through working on numerous other tasks. We've

completed a moderate number of activities while working on the project, so we can

anticipate that our estimation of the tasks will be more accurate than it was previously.

V. Conclusion

We had the chance to learn and use our skills in a joyful and studyful environment throughout our training at ShinhanDS with Likelion Vietnam. Likelion Vietnam has given us a wealth of support and practical knowledge that is hard to acquire while still in school. We've learned how to arrange our work professionally and carefully as a result of working on the project with the mentor of Likelion Vietnam. It was also a chance for us to get to know the team members and work with each other. The training program was not only a chance for us to learn new skills and complete requirements for our academic programs; it also provided us with time to hang out and enjoy ourselves.