

|  |
| --- |
|  |

|  |
| --- |
| **FPT UNIVERSITY** |
| PRJ301 Assignment Document |
| Online Shopping Laptop Store |

|  |  |
| --- | --- |
| **Group Members** | Trương Nguyễn Công Trung – SE151539  Nguyễn Nhật Khang– SE140188  Nguyễn Hữu Phước – SE140866 |

- Buon Ma Thuot, July/2021 –

**Table of contents**

I. Overview

1. User story

2. Use case diagram

II. Database

1. Entity relationship diagram
2. List of tables

III. System design

1. Overview – MVC model
2. Class diagram
3. Sequence diagram

IV. Main screen

1. Overview – sitemap
2. Screens

V. Conclusion and Discussion

**I. Overview**

We will build a web-app to sell laptop online. User must login to the system, and then they can see information of laptops we have. Once they have decided which laptop they want to build, they can add it to their cart on the system. Then users can view all the laptops in their cart, remove or update the quality of laptops and make an order. To operate the system, we may need an admin user who manage all other account in the system, and many employee users who manage laptops ‘s information in the system.

1. User story:

1.1 Admin:

- As an admin, I want to login to the system, so that I can use my admin functions.

- As an admin, I want to be able to view all accounts in the system, so that I can control all user in the system.

- As an admin, I want to be able to edit an account in the system, so that I can manage who will be employee, who will be customer.

- As an admin, I want to be able to delete an account in the system, so that I can prevent any potential user.

- As an admin, I want to search accounts, so that it is easier to do other tasks.

1.2 Employee:

- As an employee, I want to login to the system, so that I can use my employee functions.

- As an employee, I want to be able to create new product, so that the system can have information about a new laptop that we import.

- As an employee, I want to edit information of a laptop, so that if there is any information that is incorrect or need to be update such as quality we have, we do it.

- As an employee, I want to delete a laptop, so that if there is any laptop we no longer sell it, we can delete it.

- As an employee, I want to view all laptops, so that I can manage all laptop in the system.

- As an employee, I want to search laptops in the system based on some information, so that it is more convenient to do other task.

1.3 User:

- As a user, I want to login to the system, so that I can use my customer functions.

- As a user, I want to view all laptops, so that I can select which one I should buy.

- As a user, I want to search laptops in the system based on some information, so that it is more convenient to select laptop.

- As a user, I want to have an cart, so that I can add laptop, update quantity of laptop in the cart but not buy it immediately.

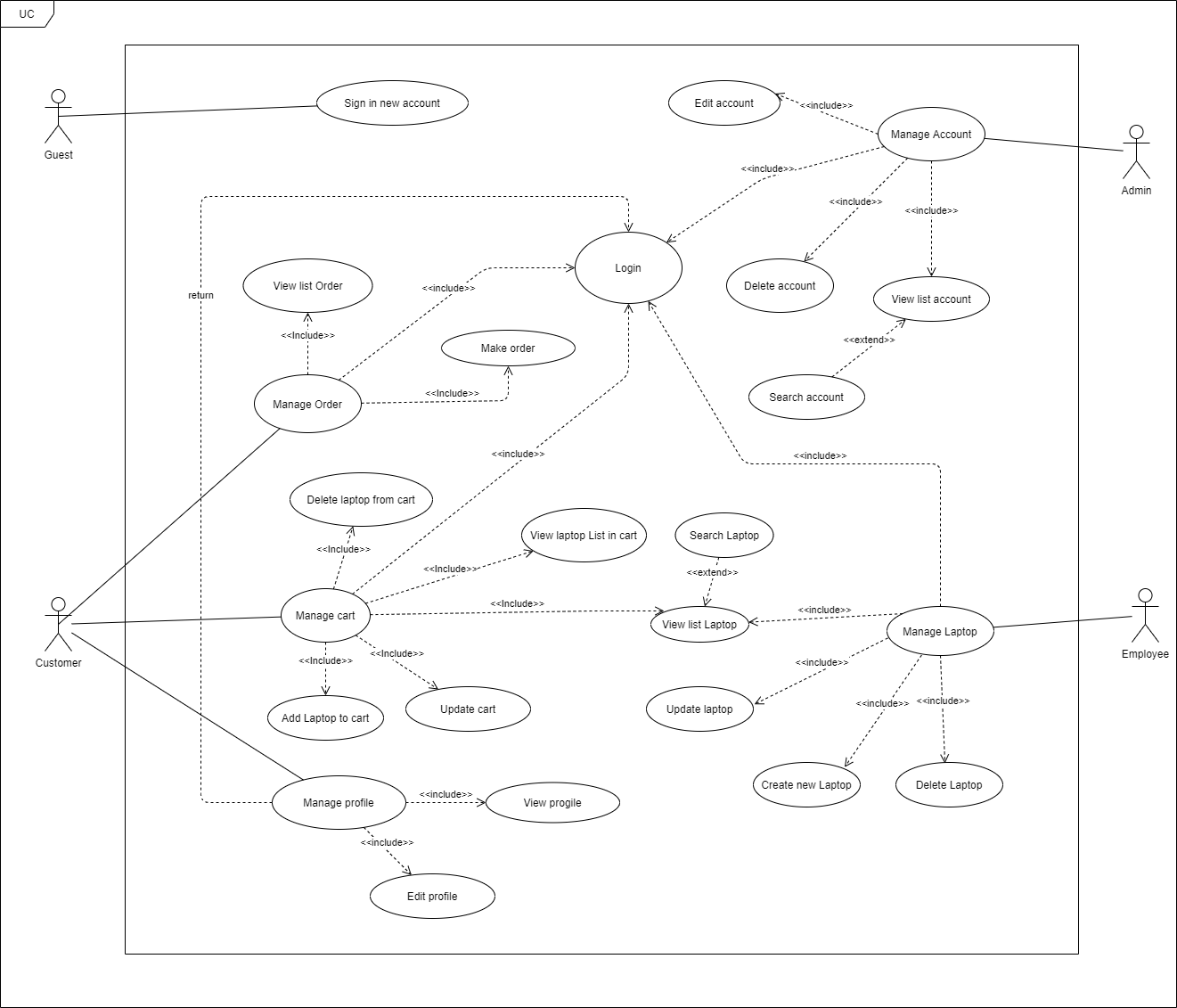
- As a user, I want to make an order, that is I actually buy some laptop and I can also be able to view all the laptop I have bought.

- As a user, I want to view my profile and update it, so I can manage my information on the system.

1.4 Guest (Unauthenticated user):

- As a guest, I want to register a new account, so that I can be a customer and buy laptop.

2. Use case diagram:



*Diagram 1: Use case diagram*

3. Use case detail:

3.1. System actors

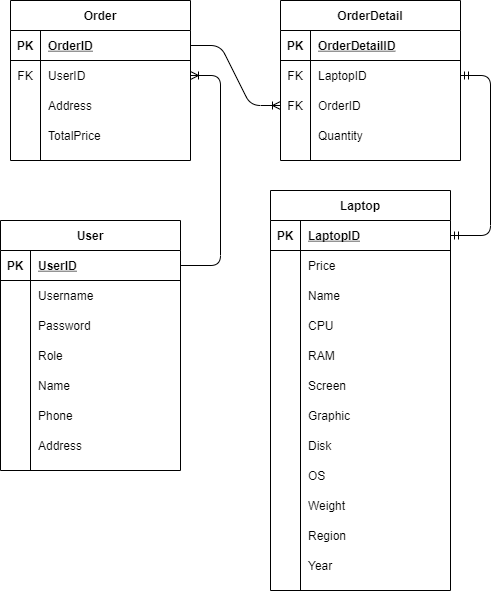
|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Admin | Manage user accounts |
| 2 | Employee | Manage laptop |
| 3 | User | Manage their own cart, profile and make order. |
| 4 | Guest | Unauthenticated user |

3.2. Use case list

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use case** | **Primary Actors** | **Secondary Actors** |
| 1 | Register new account | Guest | N/A |
| 2 | View profile | User | N/A |
| 3 | Update profile | User | N/A |
| 4 | Add laptop to cart | User | N/A |
| 5 | Delete laptop in cart | User | N/A |
| 6 | Update quantity of laptop in cart | User | N/A |
| 7 | View cart | User | N/A |
| 8 | Make order from cart | User | N/A |
| 9 | View list orders | User | N/A |
| 10 | View list laptops | User, Employee | N/A |
| 11 | Search laptops | User, Employee | N/A |
| 12 | Update laptop | Employee | N/A |
| 13 | Delete laptop | Employee | N/A |
| 14 | Create laptop | Employee | N/A |
| 15 | View list accounts | Admin | N/A |
| 16 | Update account | Admin | N/A |
| 17 | Delete account | Admin | N/A |
| 18 | Search accounts | Admin | N/A |

**II. Database**

1. Entity relationship diagram:



**Figure 2.1:** *Entity relation diagram*

2. List of tables

2.1 Users

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| 1 | UserID | int | 4 | Y | Y | PK |  |
| 2 | Username | nvarchar(150) | 250 | Y | Y |  |  |
| 3 | Password | nvarchar(1000) | 2000 | N | Y |  |  |
| 4 | Role | int | 4 | N | Y |  | 1: Admin  2: Customer  3: Employee |
| 5 | Name | Nvarchar(50) | 100 | N | Y |  |  |
| 6 | Phone | Nvarchar(50) | 100 | N | Y |  |  |
| 7 | Address | Nvarchar(150) | 250 | N | Y |  |  |

2.2 Laptop

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| 1 | LaptopID | int | 4 | Y | Y | PK |  |
| 2 | Price | Float | 4 | N | Y |  |  |
| 3 | Name | Nvarchar(50) | 100 | N | Y |  |  |
| 4 | CPU | Nvarchar(50) | 100 | N | N |  |  |
| 5 | RAM | Nvarchar(50) | 100 | N | N |  |  |
| 6 | Screen | Nvarchar(50) | 100 | N | N |  |  |
| 7 | Graphic | Nvarchar(50) | 100 | N | N |  |  |
| 8 | Disk | Nvarchar(50) | 100 | N | N |  |  |
| 9 | OS | Nvarchar(50) | 100 | N | N |  |  |
| 10 | Weight | Float | 4 | N | N |  |  |
| 11 | Region | Nvarchar(50) | 100 | N | N |  |  |
| 12 | Year | int | 4 | N | N |  |  |

2.3 Order

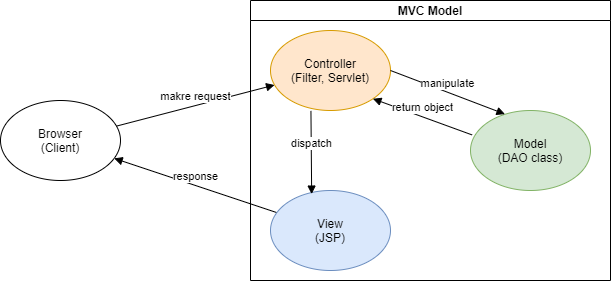
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| 1 | OrderID | int | 4 | Y | Y | PK |  |
| 2 | UserID | int | 4 | N | Y | FK |  |
| 3 | Address | Nvarchar(150) | 250 | N | Y |  | Shipment address |
| 4 | TotalPrice | float | 4 | N | Y |  |  |

2.4 OrderDetail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Field Name | Type | Size | Unique | Not Null | PK/FK | Notes |
| 1 | OrderDetailID | int | 4 | Y | Y | PK |  |
| 2 | LaptopID | int | 4 | N | Y | FK |  |
| 3 | OrderID | Int | 4 | N | Y | FK |  |
| 4 | Quantity | Int | 4 | N | Y |  |  |

**III. System design**

1. Overview – MVC model



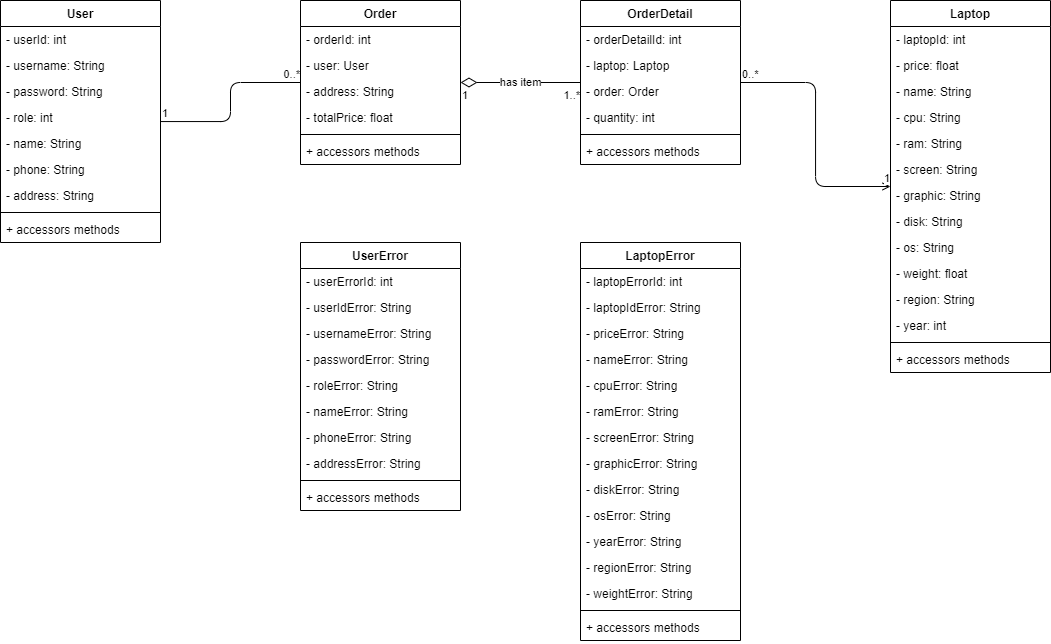
*Figure 3.1: MVC Model*

MVC Model consist of 3 main components: controller, model, view.

* Controller: accept request from client
* Model: manipulate database
* View: render UI and response to client

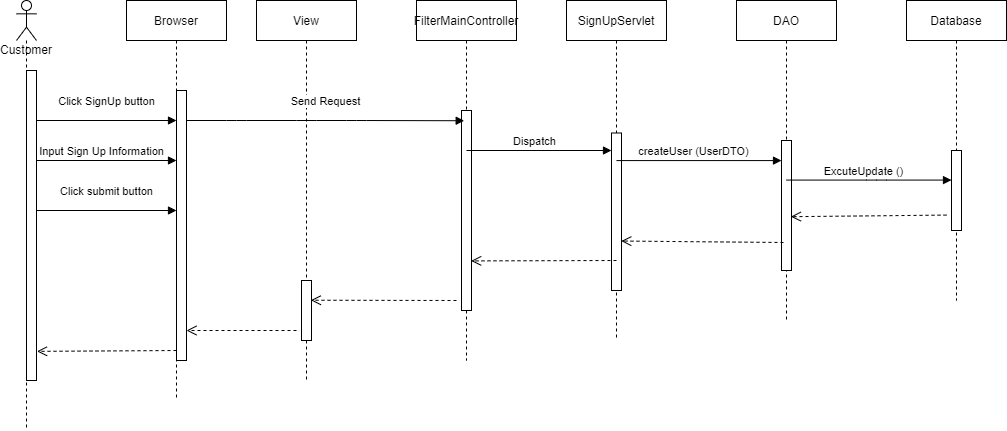
Additional technologies:

* Bootstrap: an CSS framework, design and customise front-end web quickly
* Git and Github: an source code and version control, it help us manage our code better.

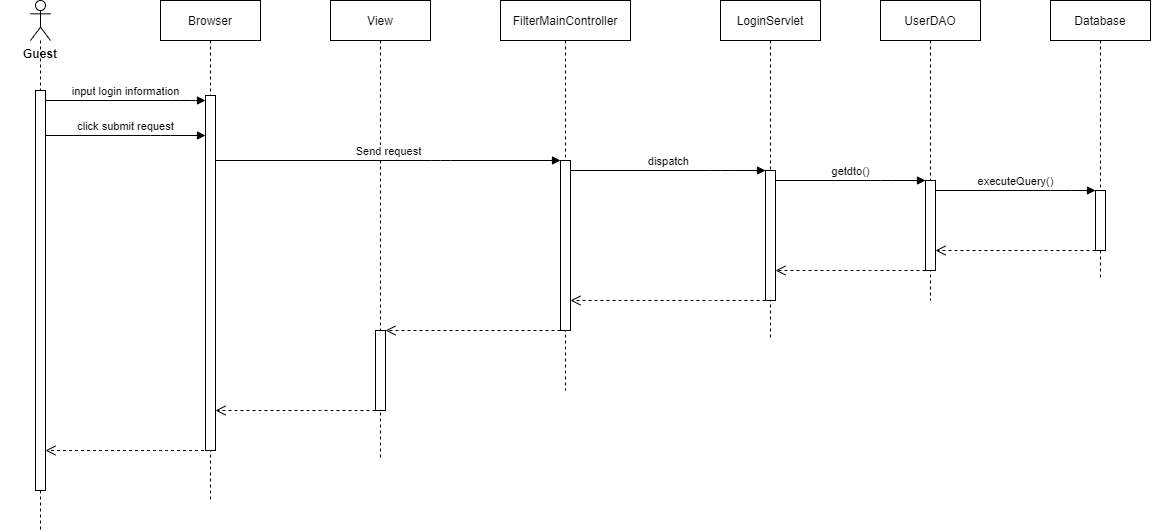
1. Class diagram

*Figure 3.2: Entity Class Diagram*

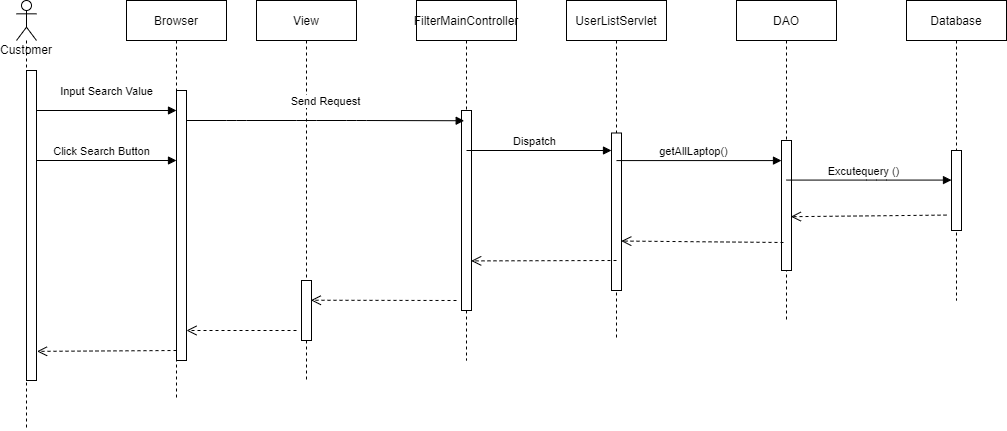
1. Sequence diagram
   1. Guest
      1. Register



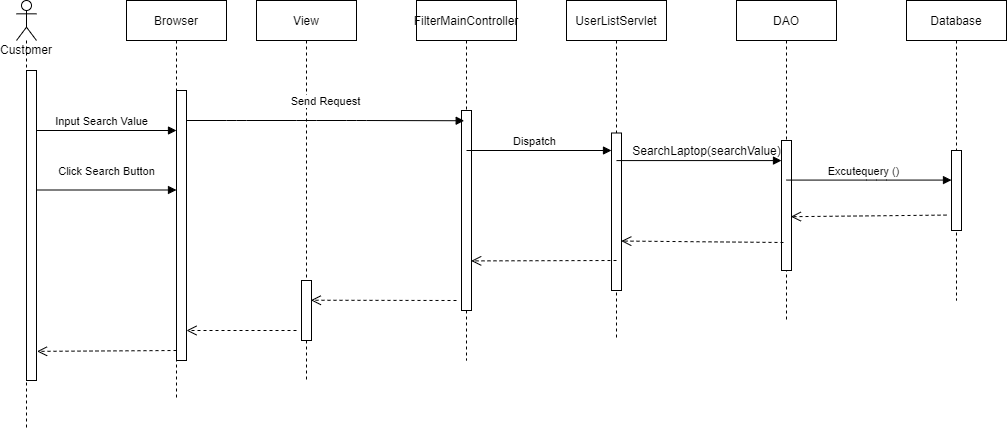
* 1. User
     1. Login



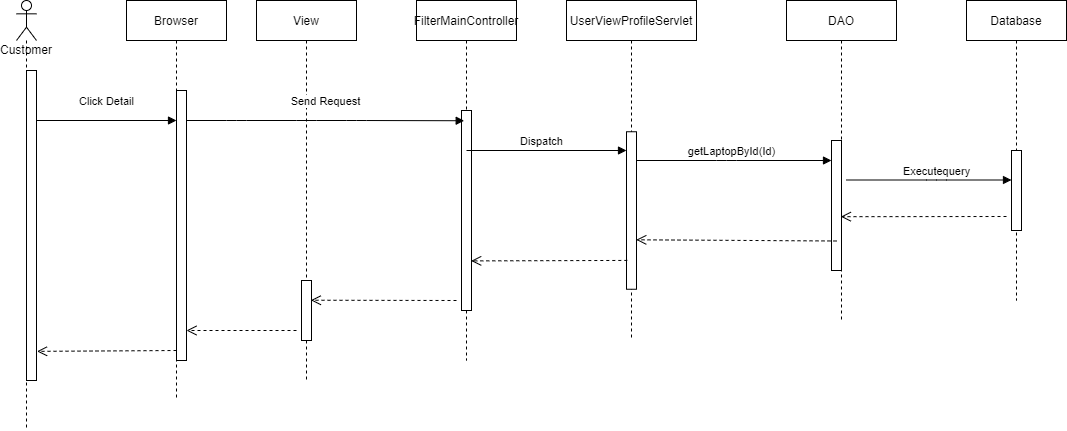
* + 1. View laptop list



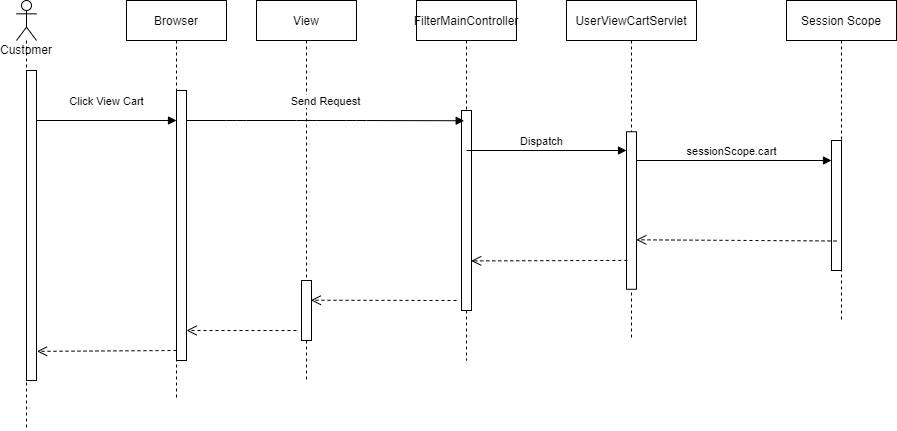
* + 1. Search laptop list



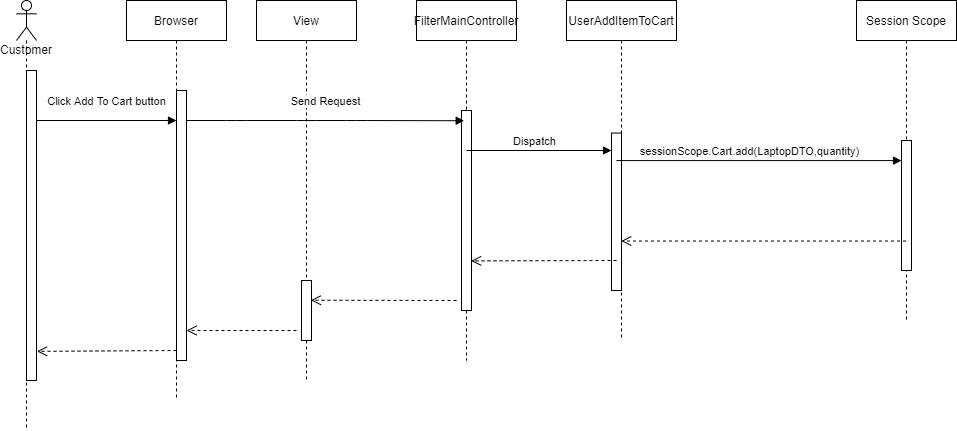
* + 1. View detail laptop



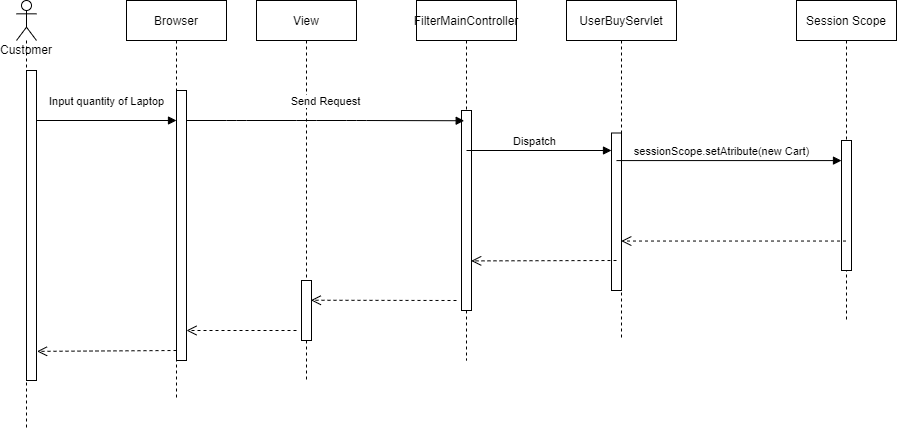
* + 1. View list laptop in cart



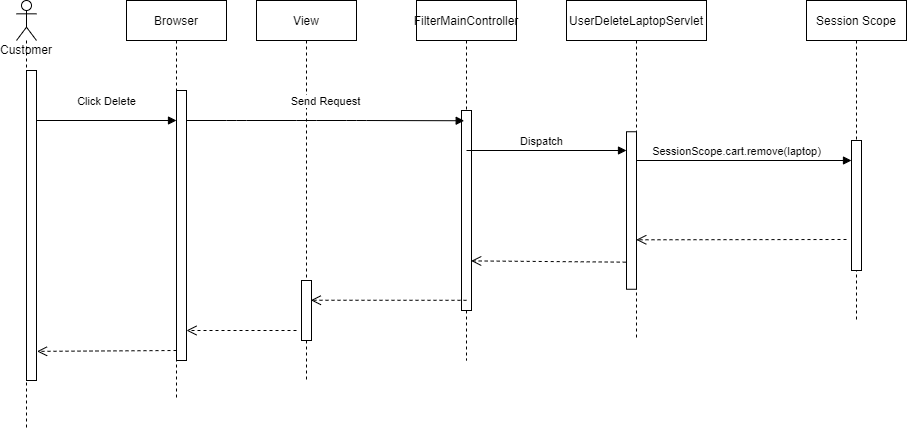
* + 1. Add laptop in cart



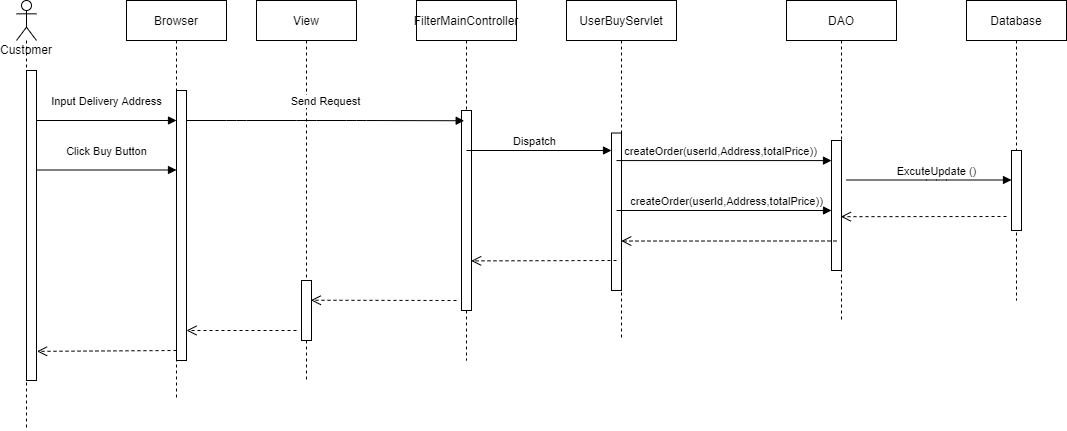
* + 1. Update laptop in cart



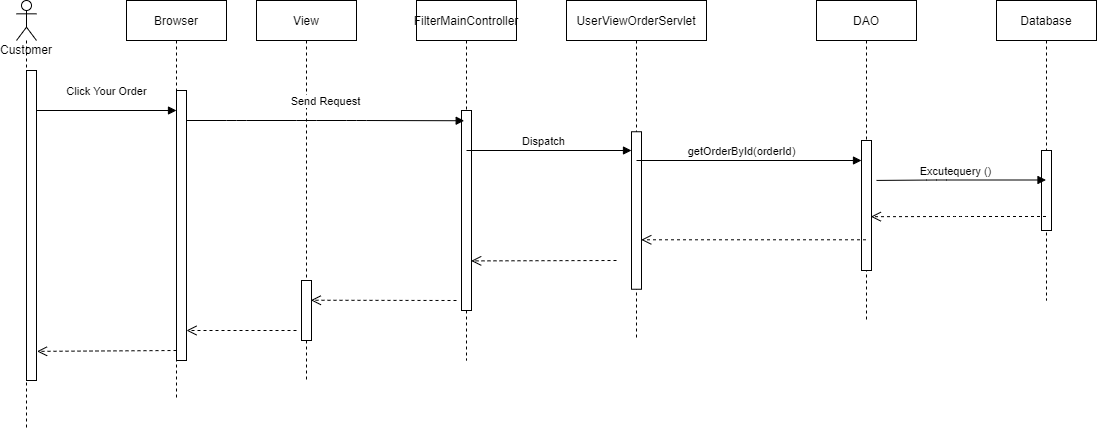
* + 1. Delete laptop in cart



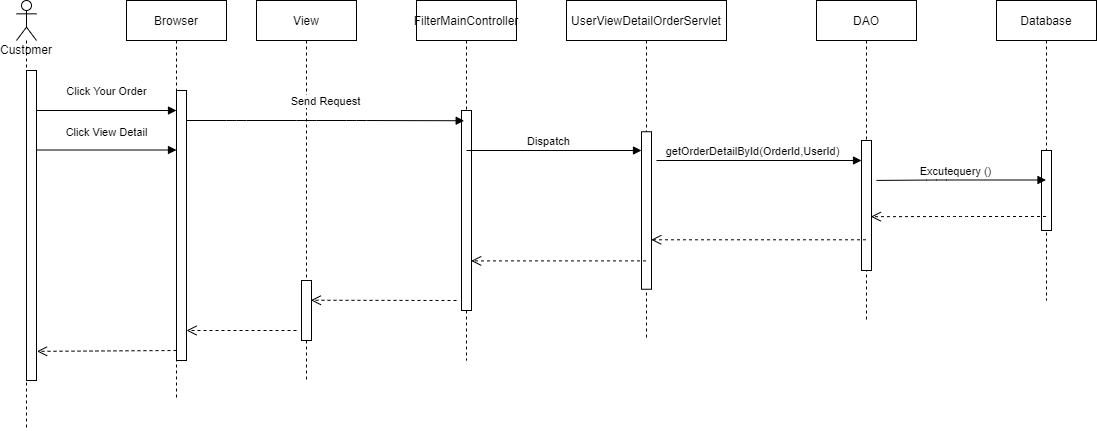
* + 1. Make order



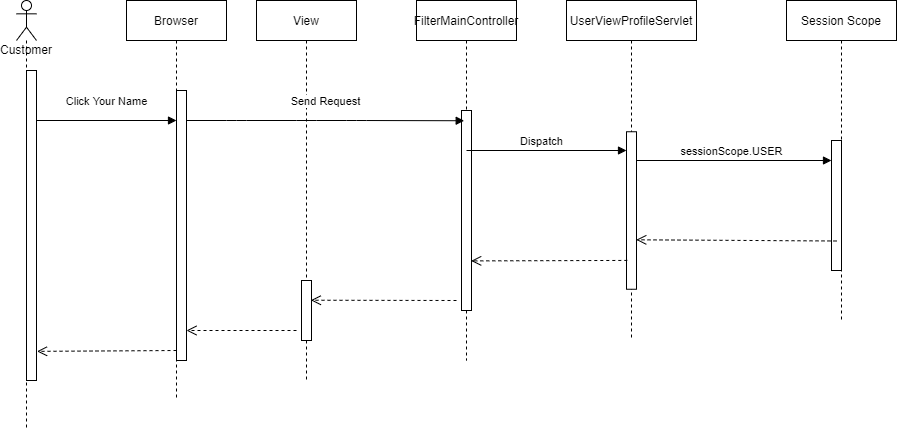
* + 1. View order list



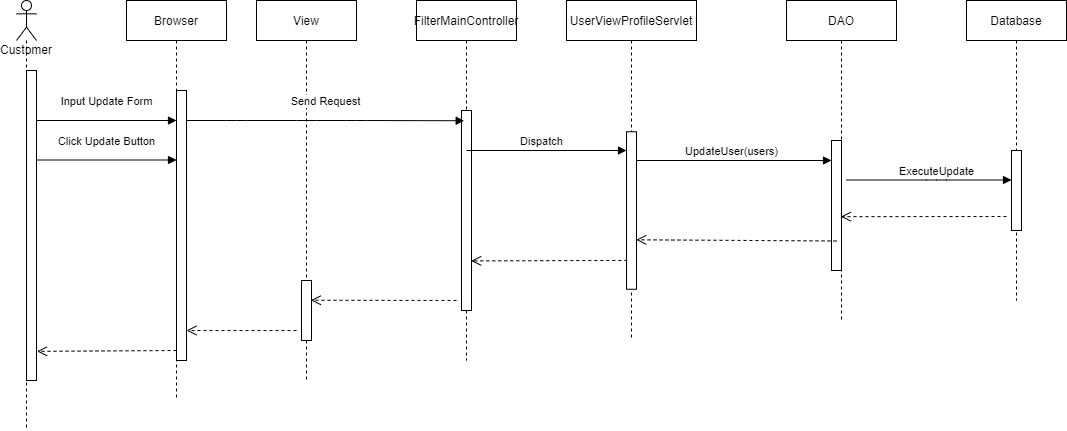
* + 1. View order detail



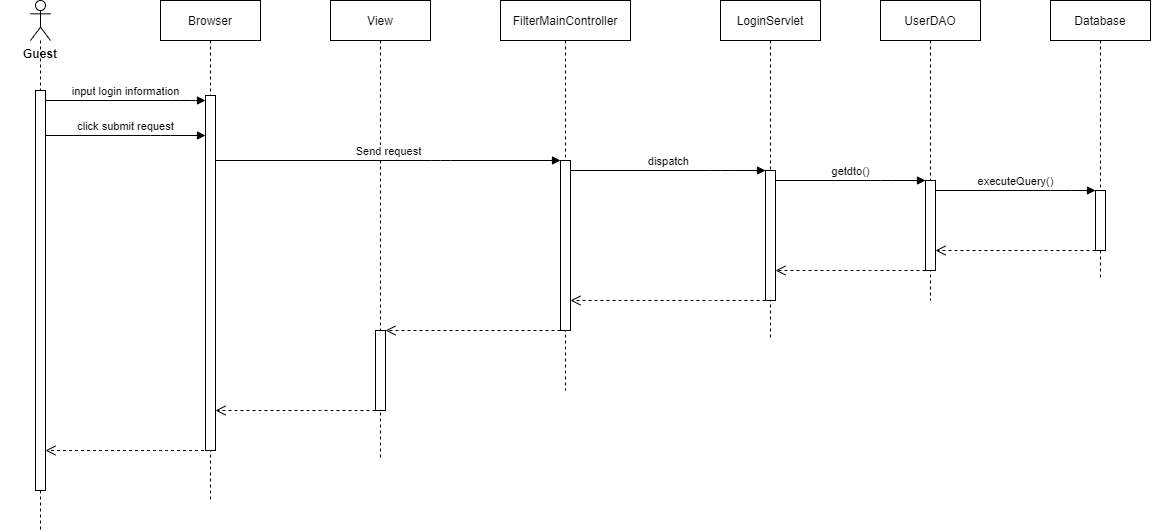
* + 1. View profile



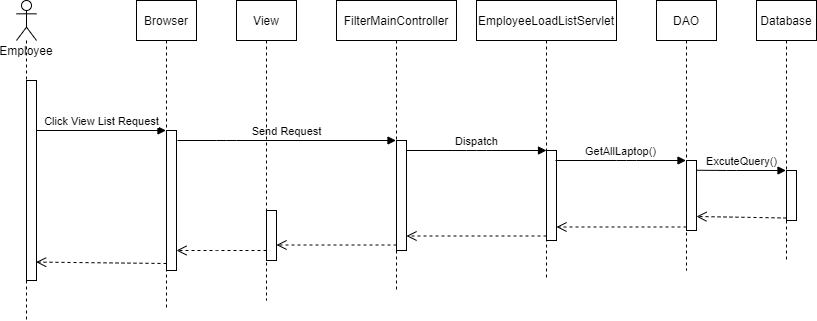
* + 1. Update profile



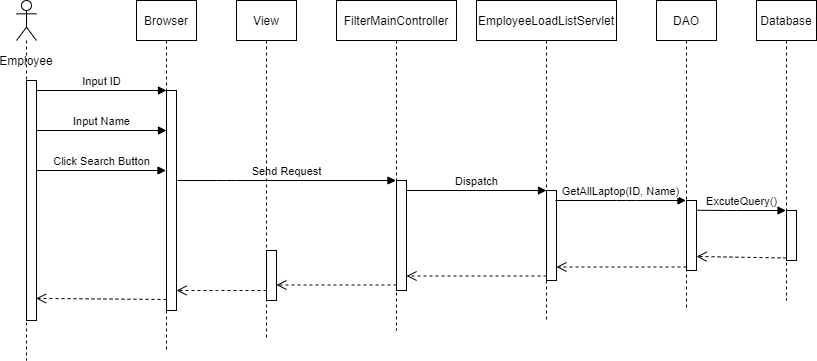
* 1. Employee
     1. Login



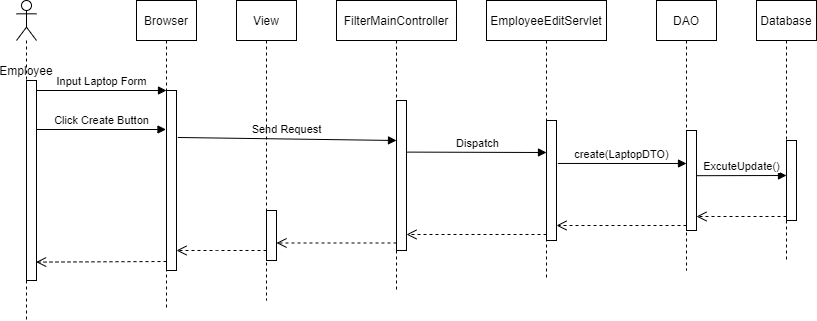
* + 1. View laptop list



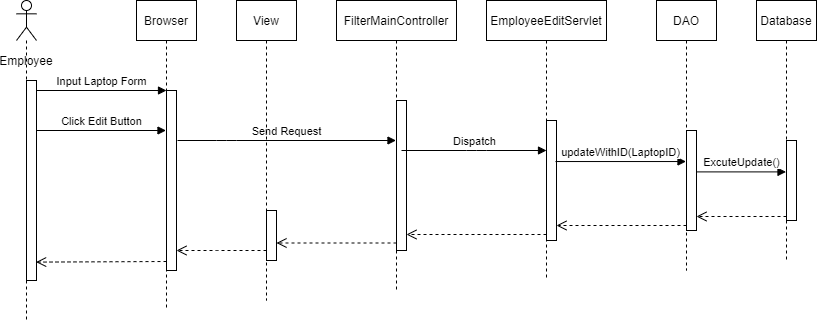
* + 1. Search laptop list



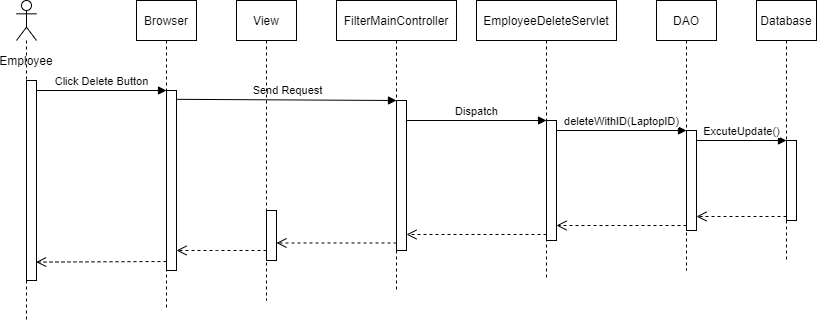
* + 1. Create new laptop list



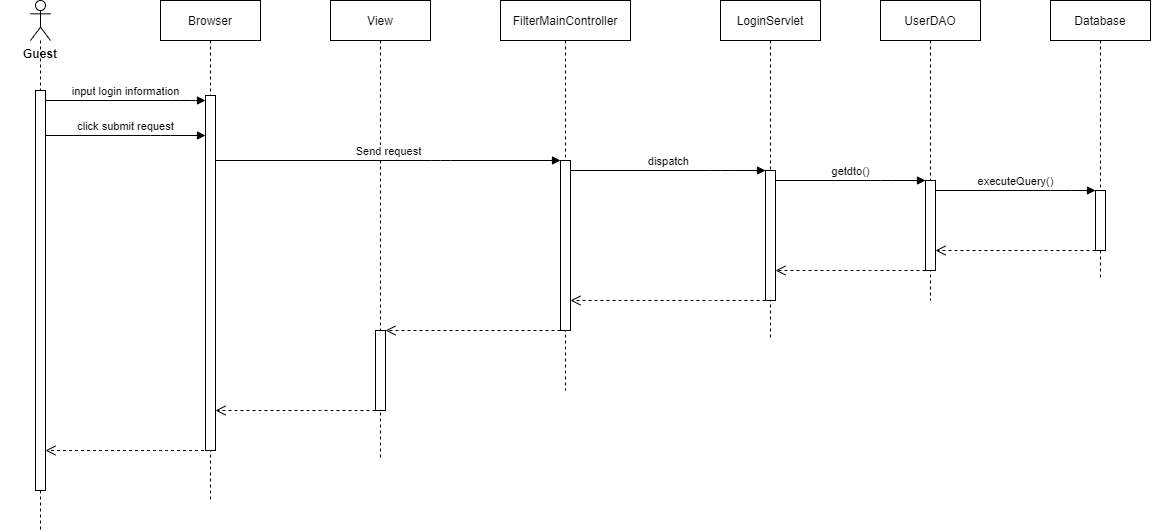
* + 1. Update laptop



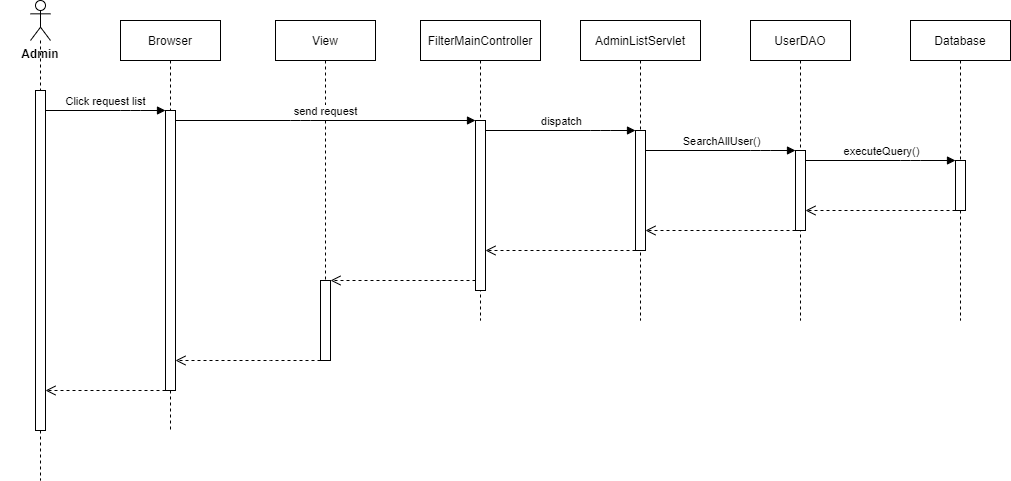
* + 1. Delete laptop



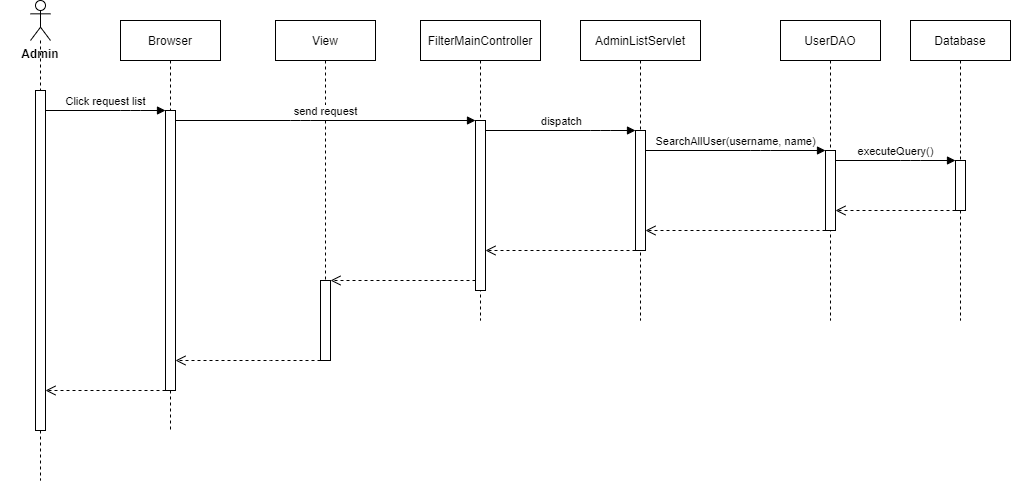
* 1. Admin
     1. Login



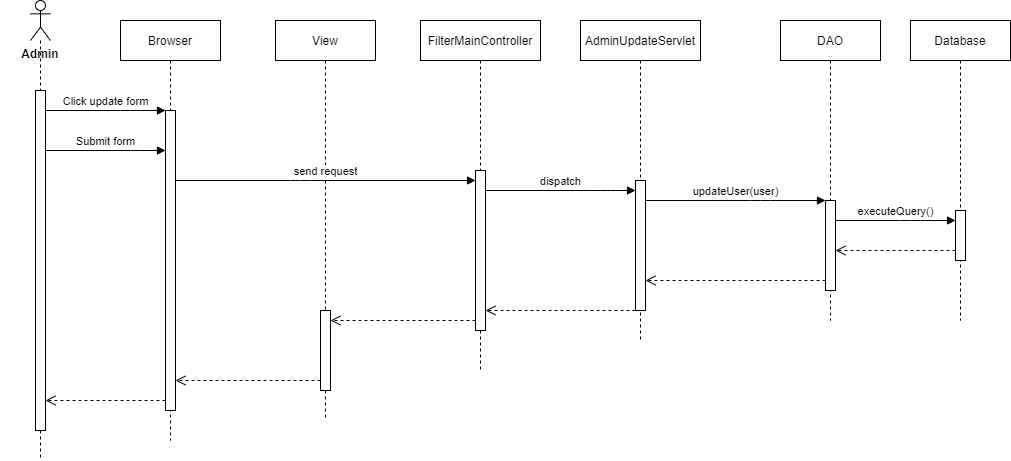
* + 1. View account list



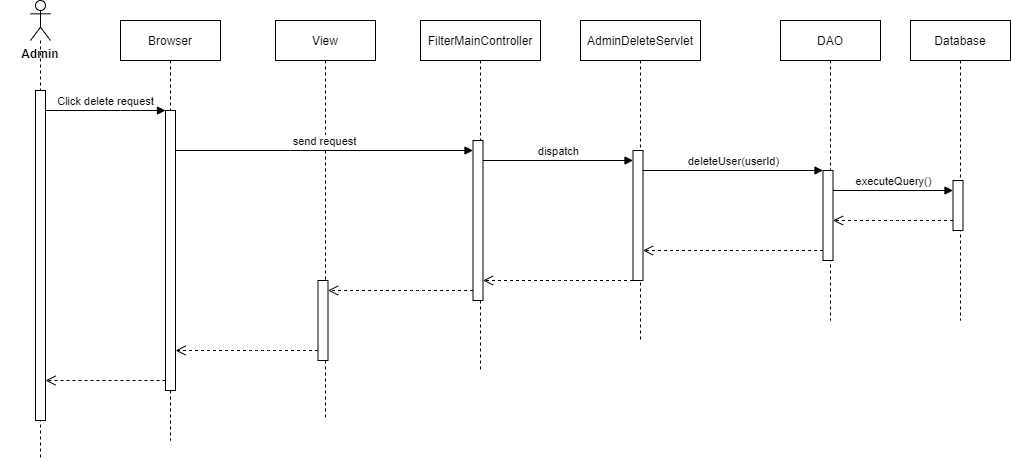
* + 1. Search account list



* + 1. Update account



* + 1. Delete account



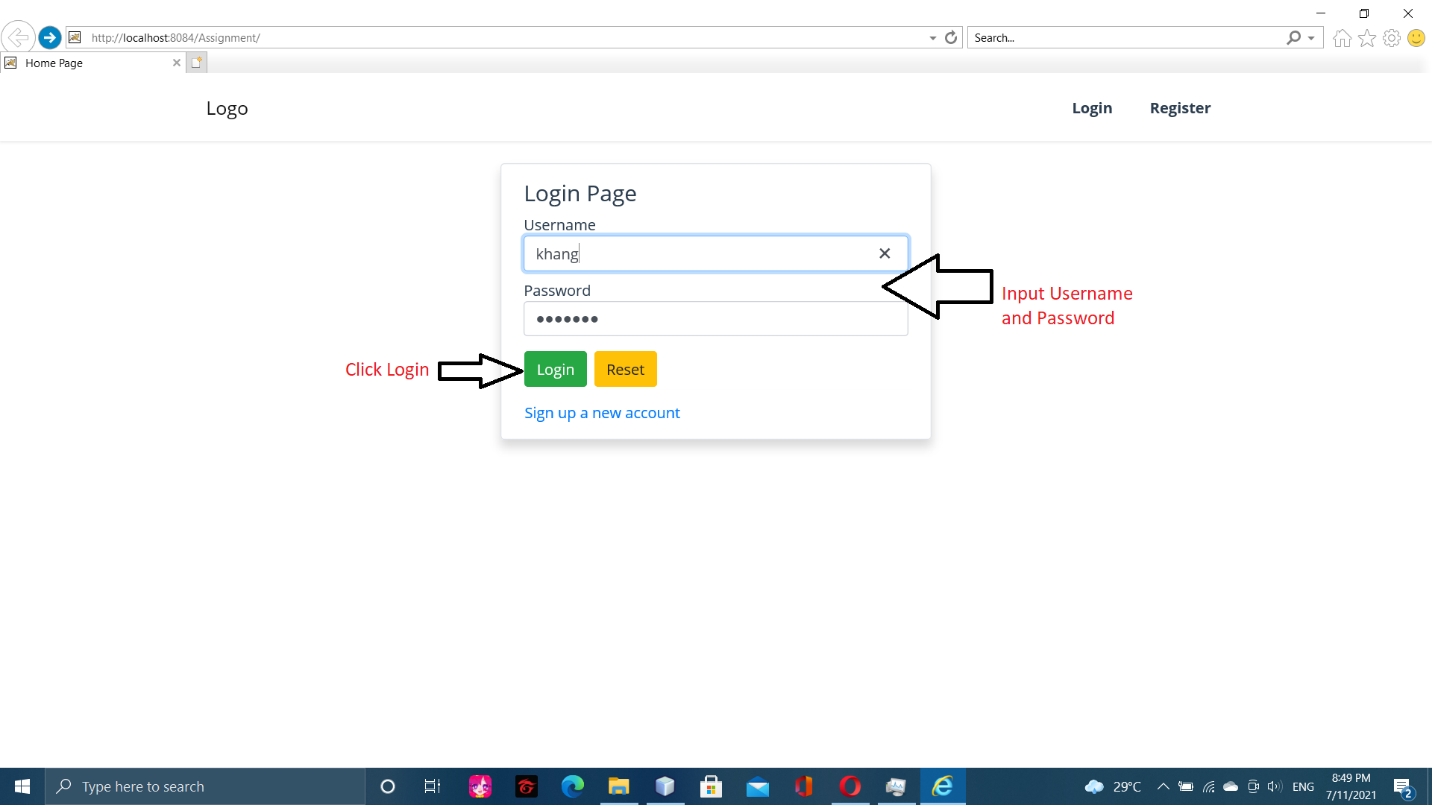
**IV. Main screen**

1. Overview – sitemap

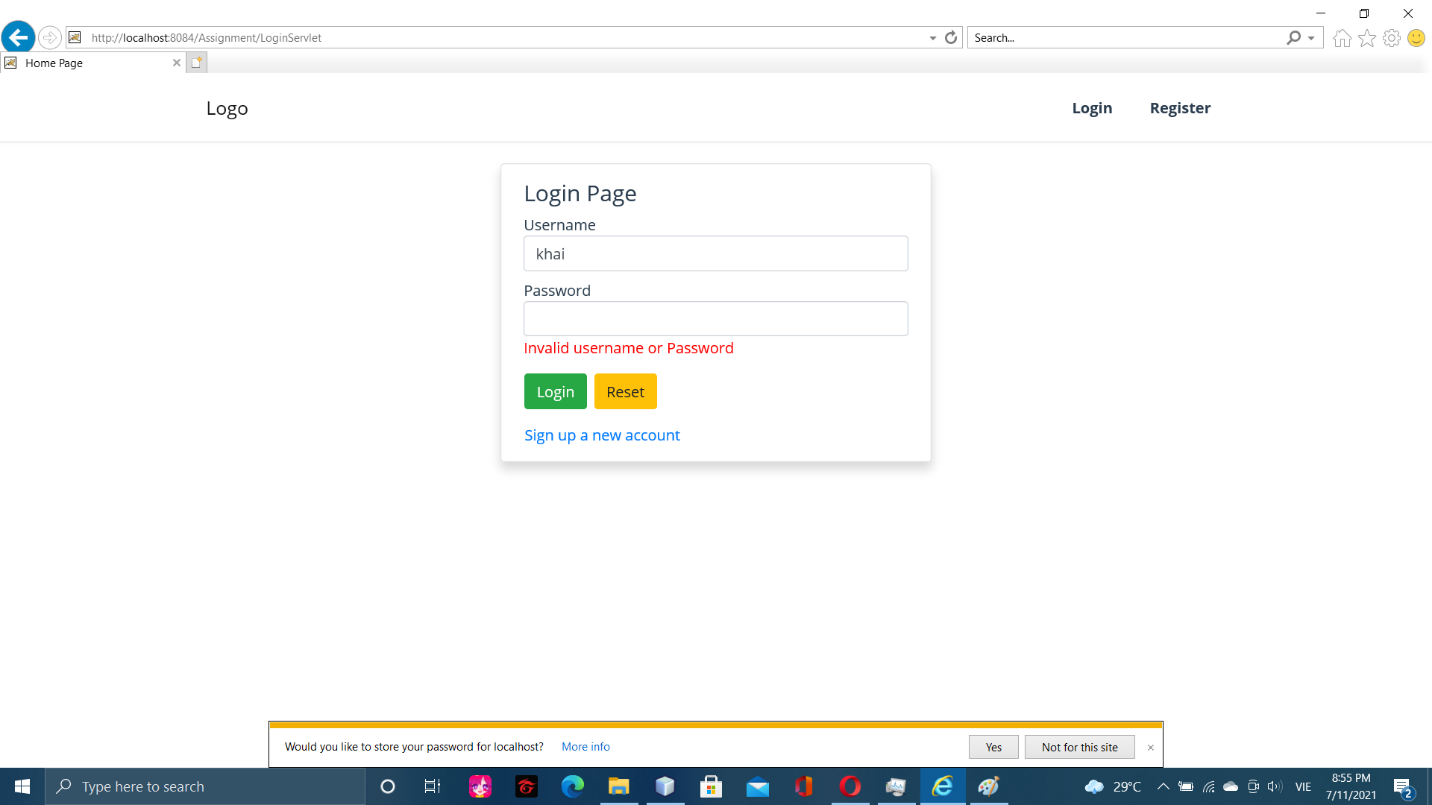


*Figure 4.1: Sitemap*

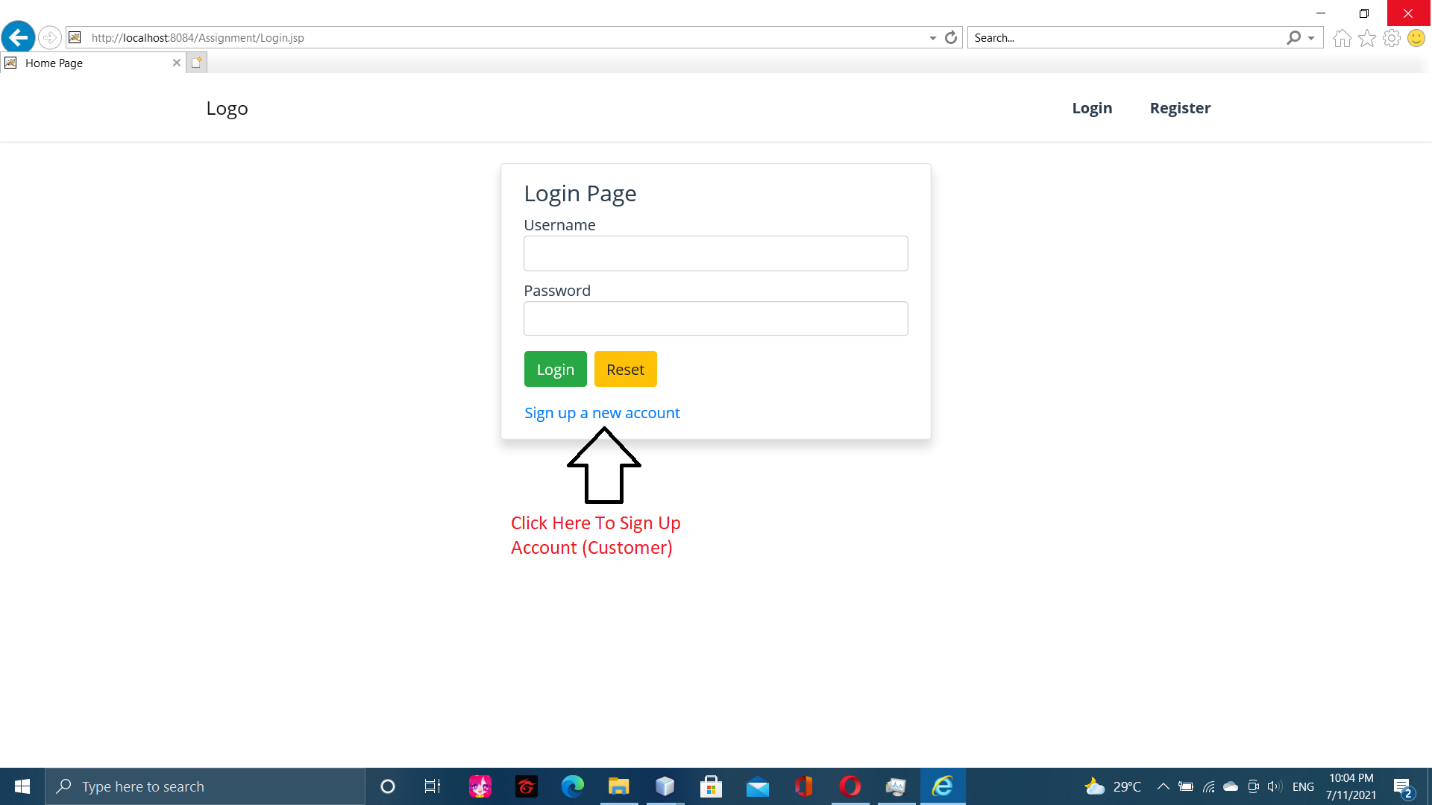
1. Screens



Login screen



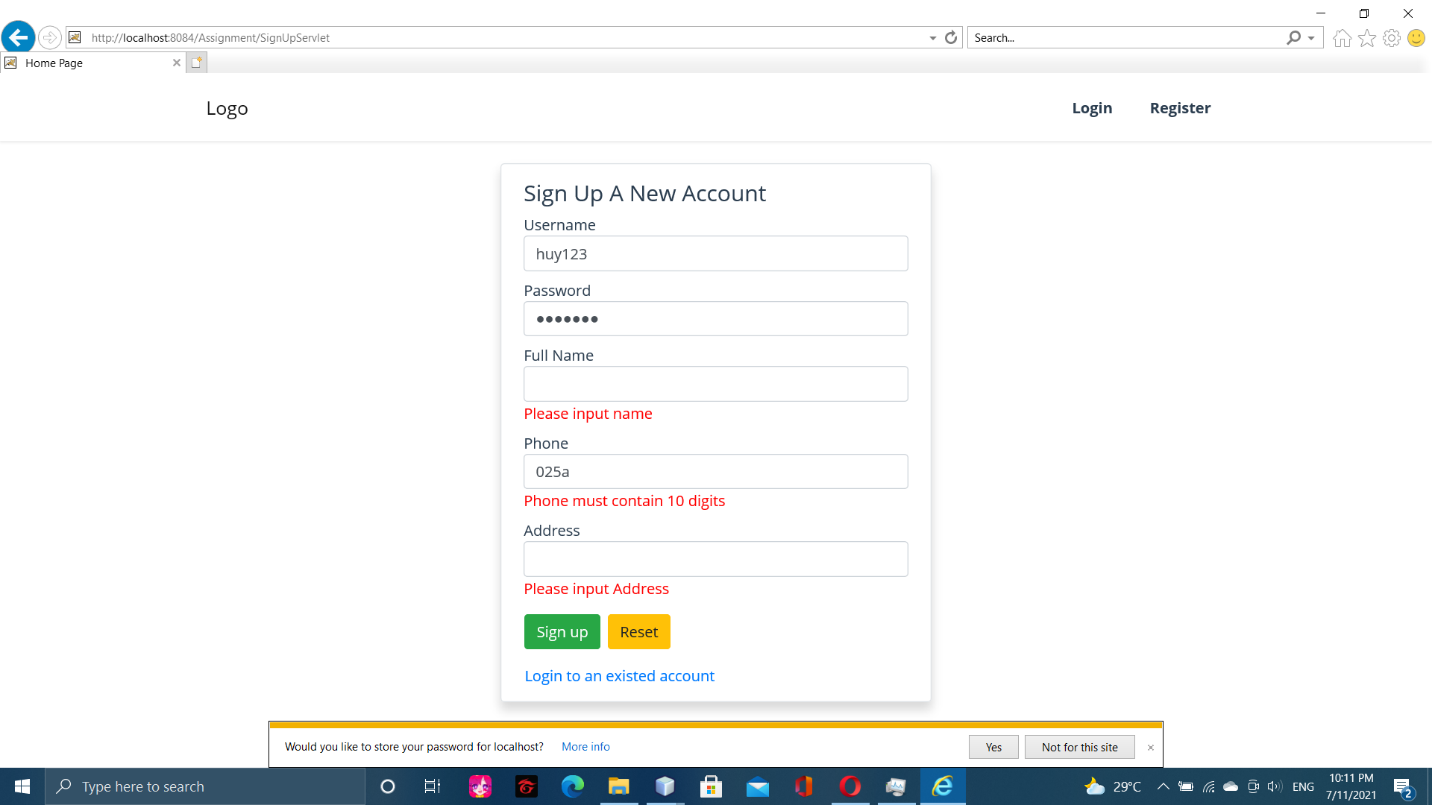
Login error



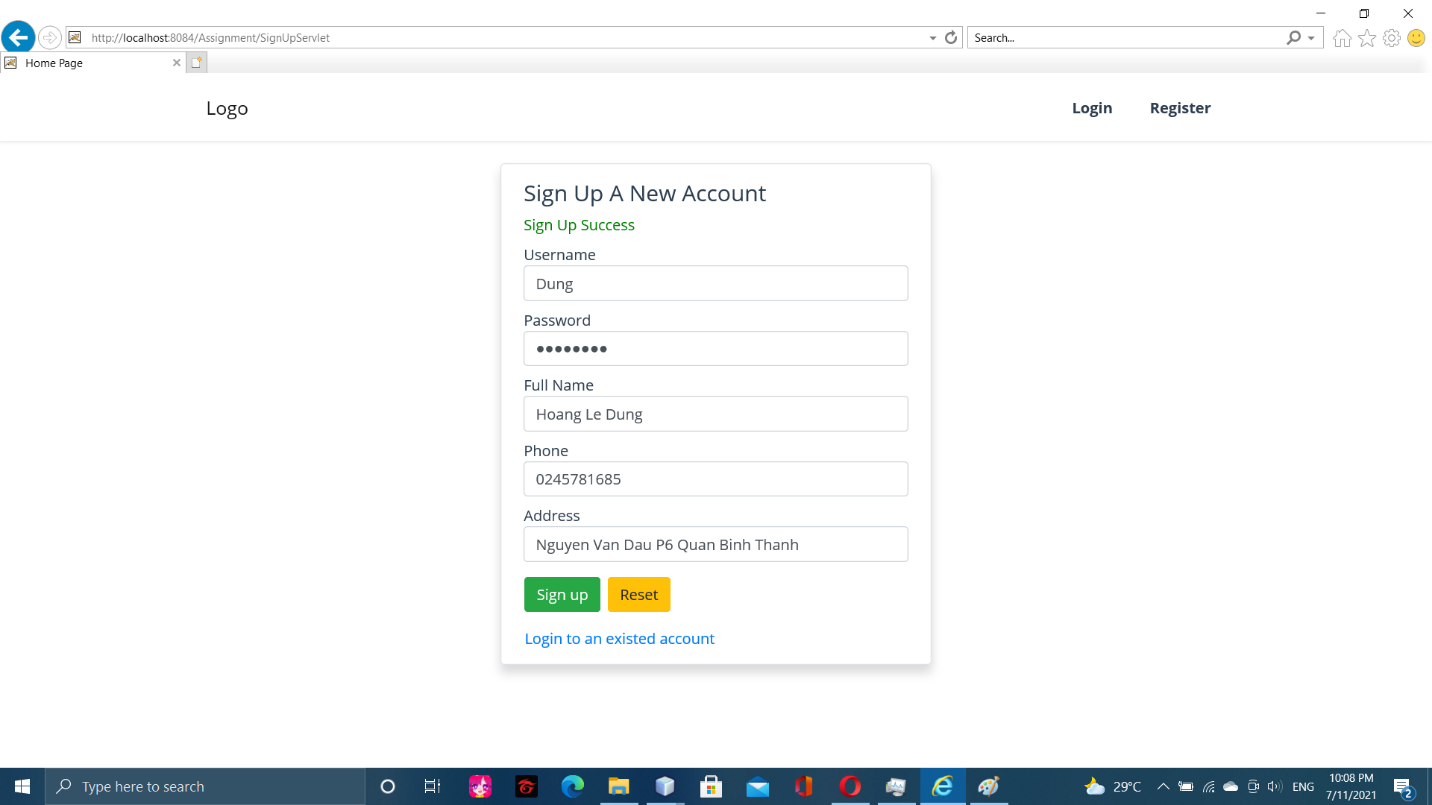
Sign up 1



Sign up 2

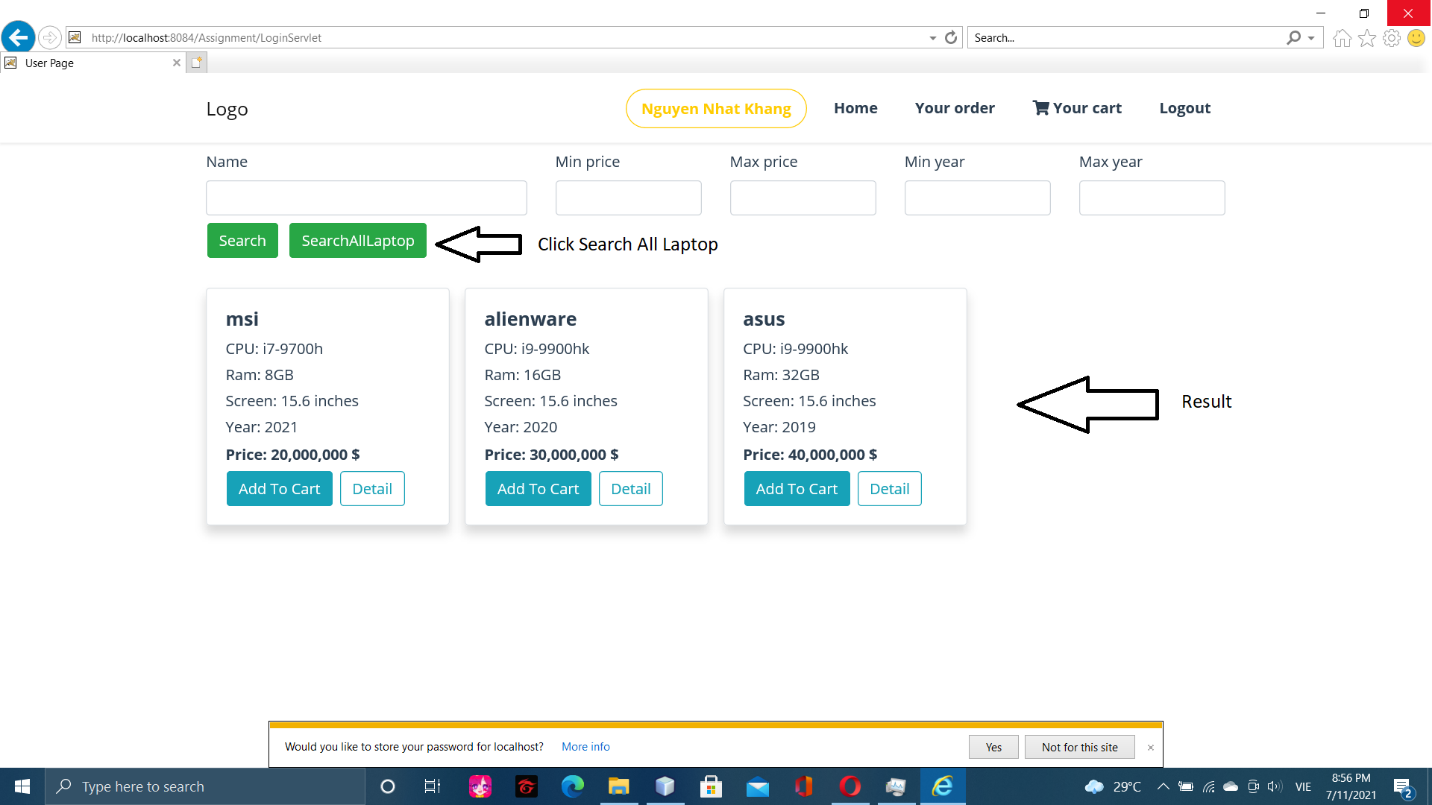


Sign up fail

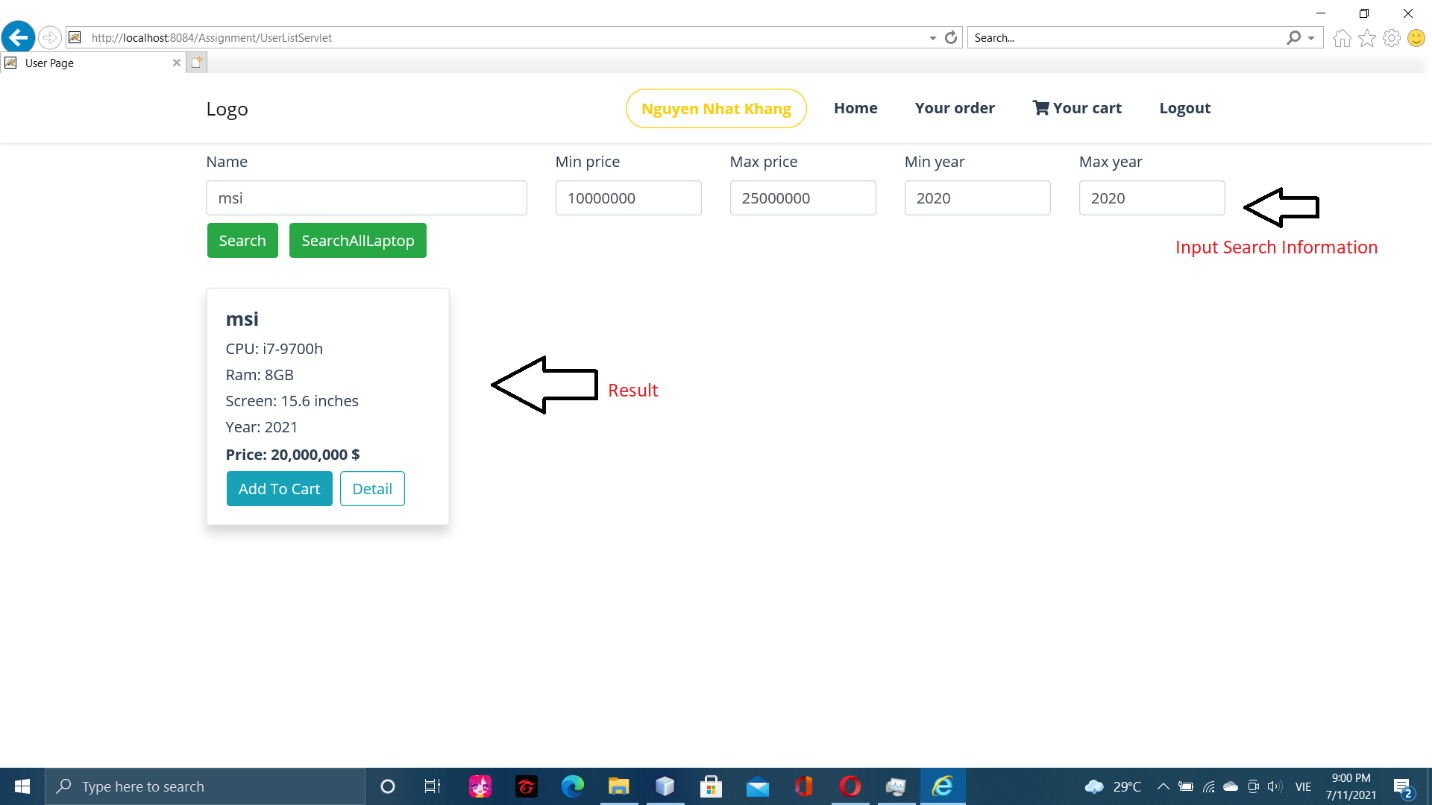


Sign up success

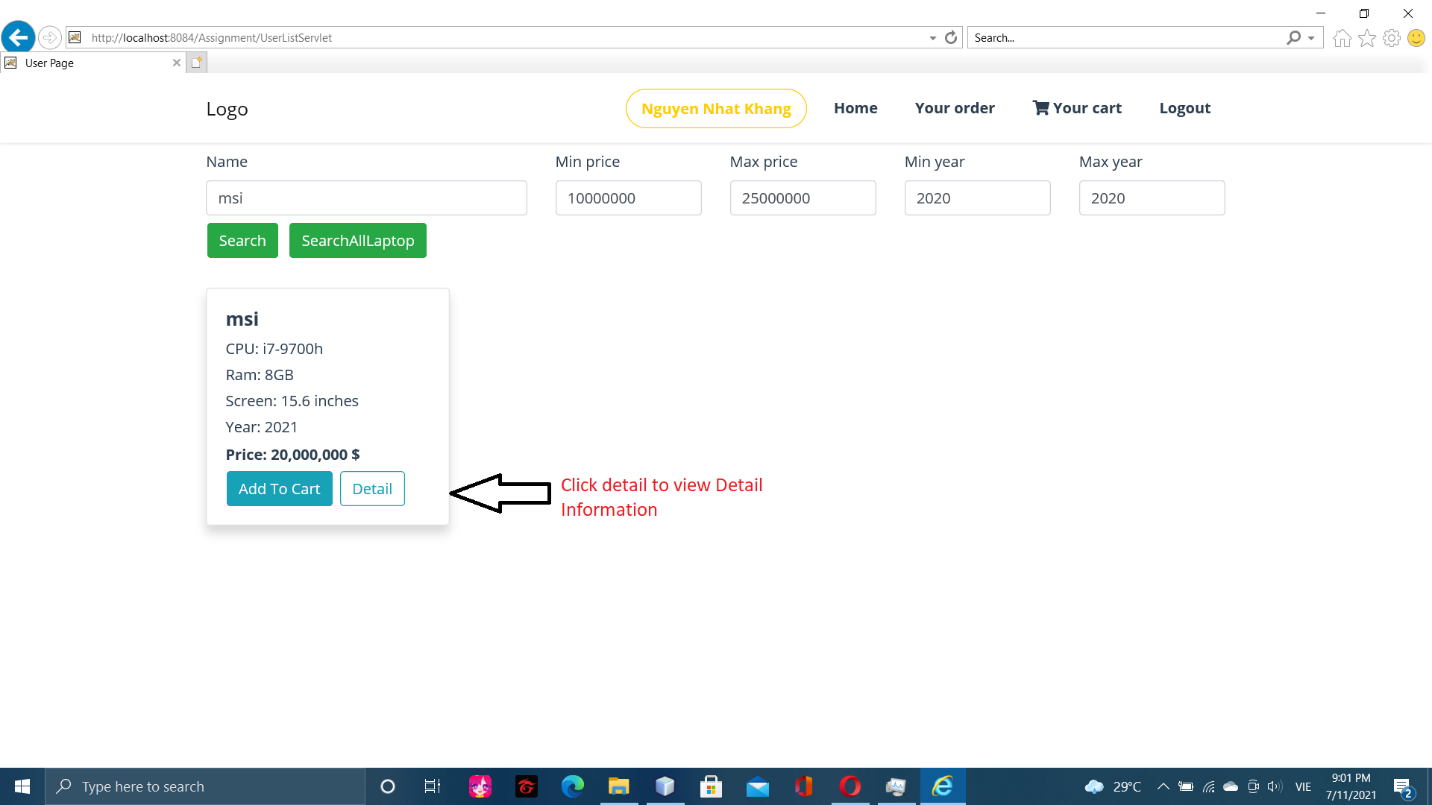
* 1. User:



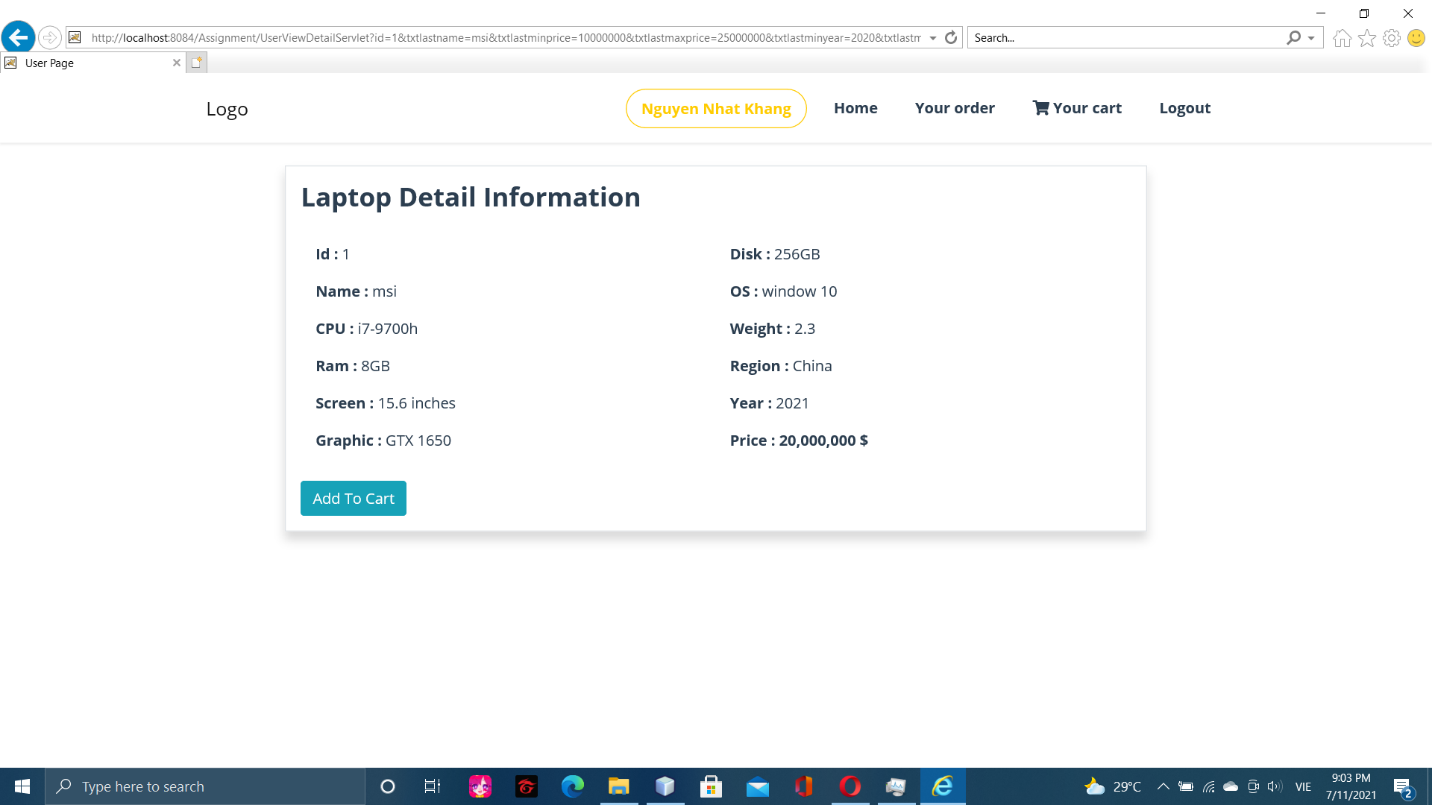
Search all laptop



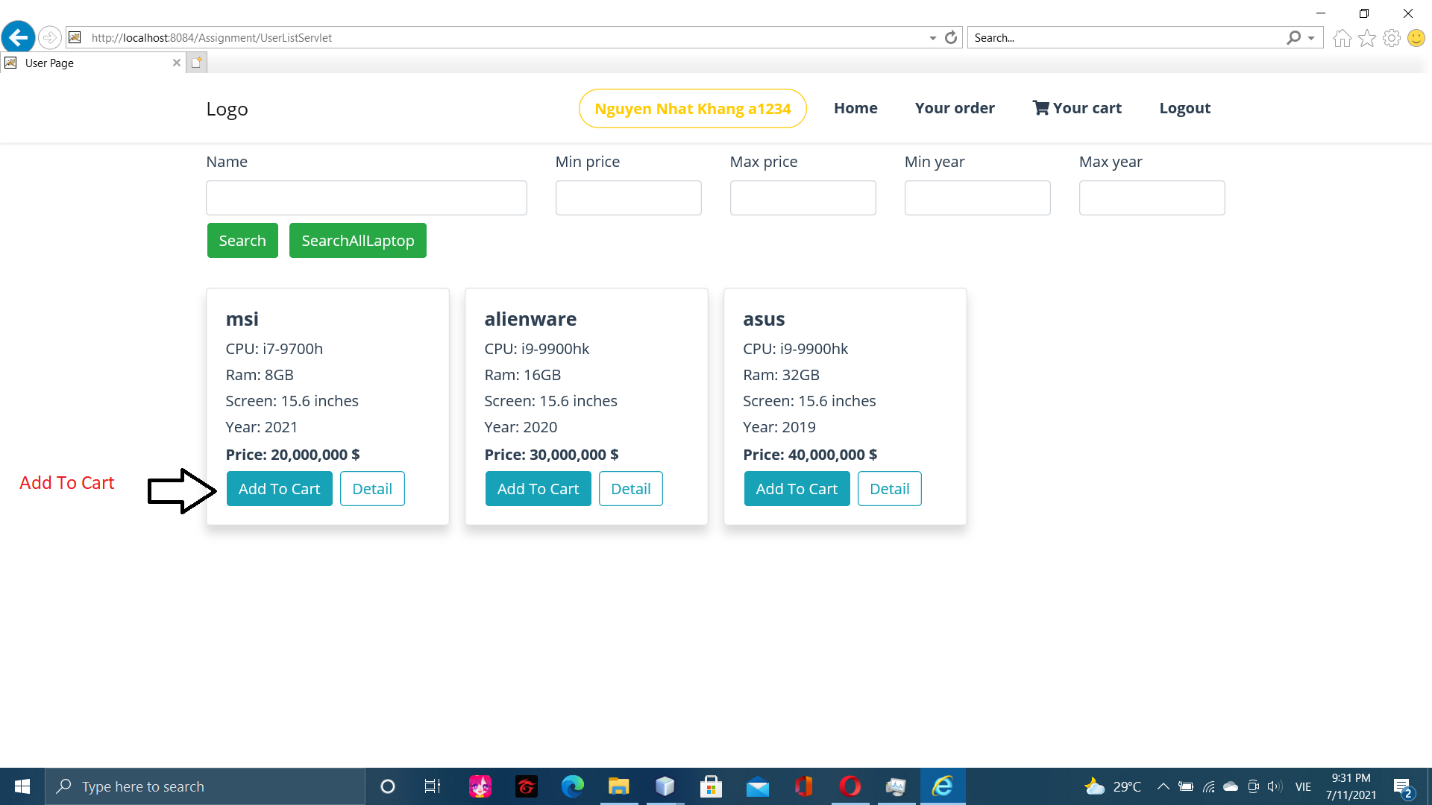
Search laptop



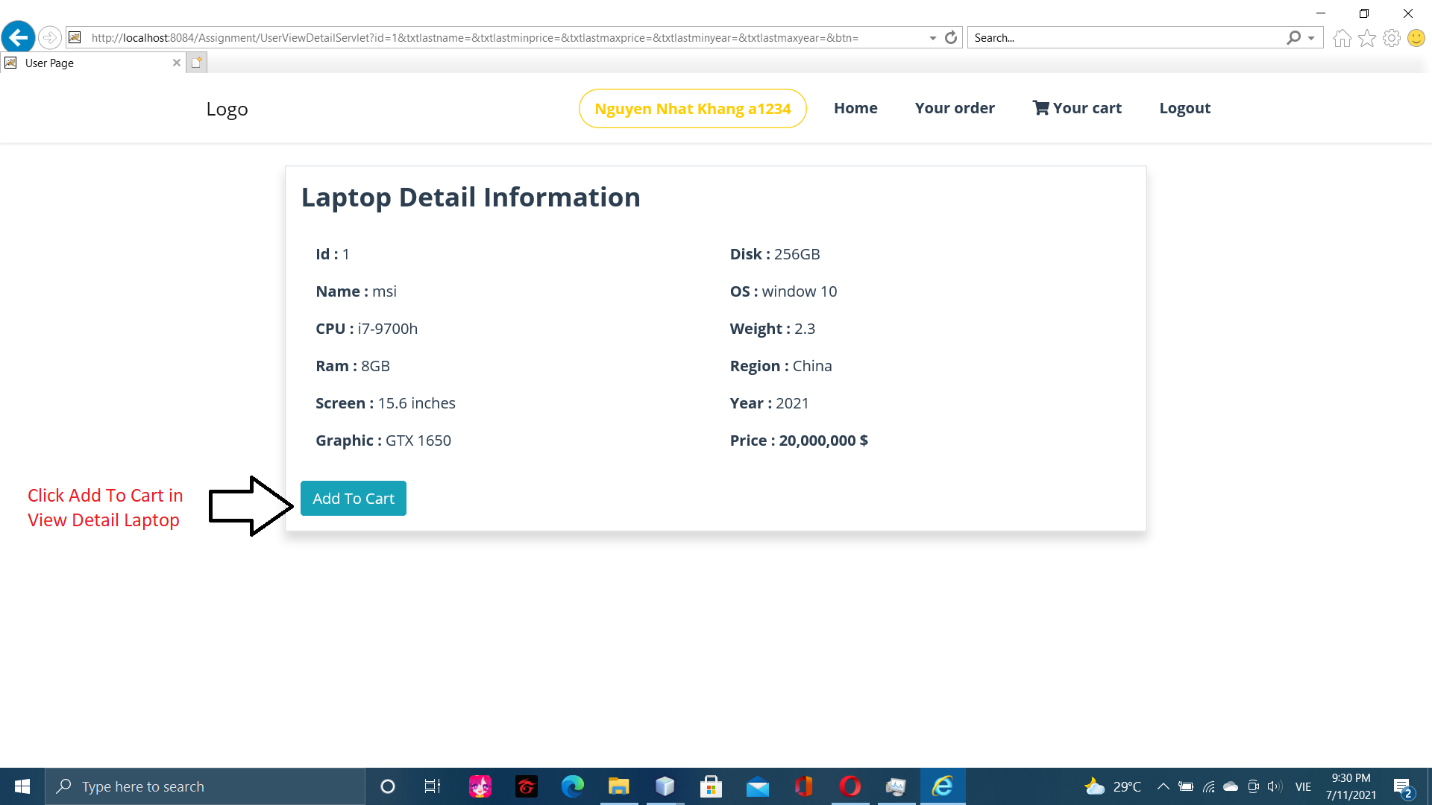
View detail laptop 1



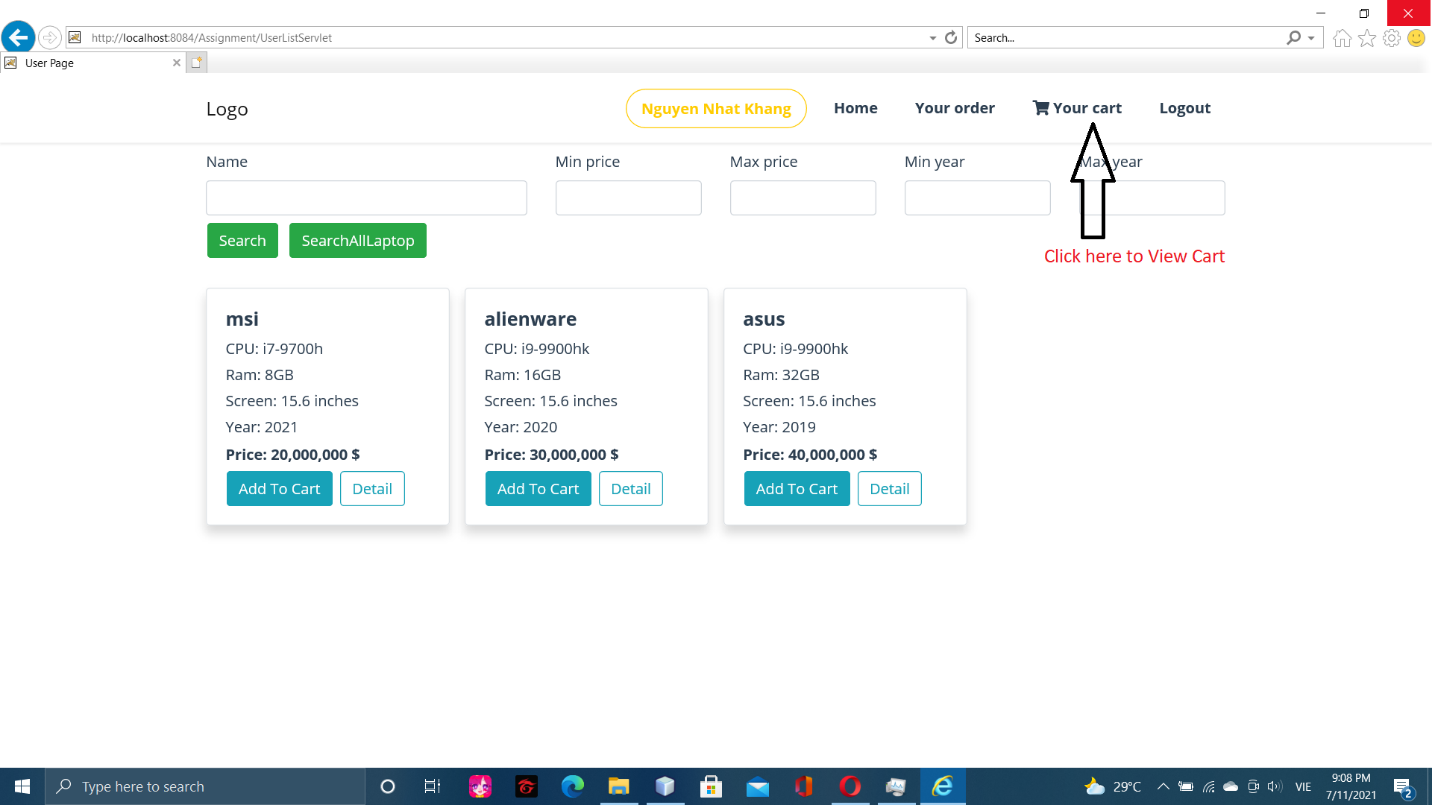
View detail laptop 2



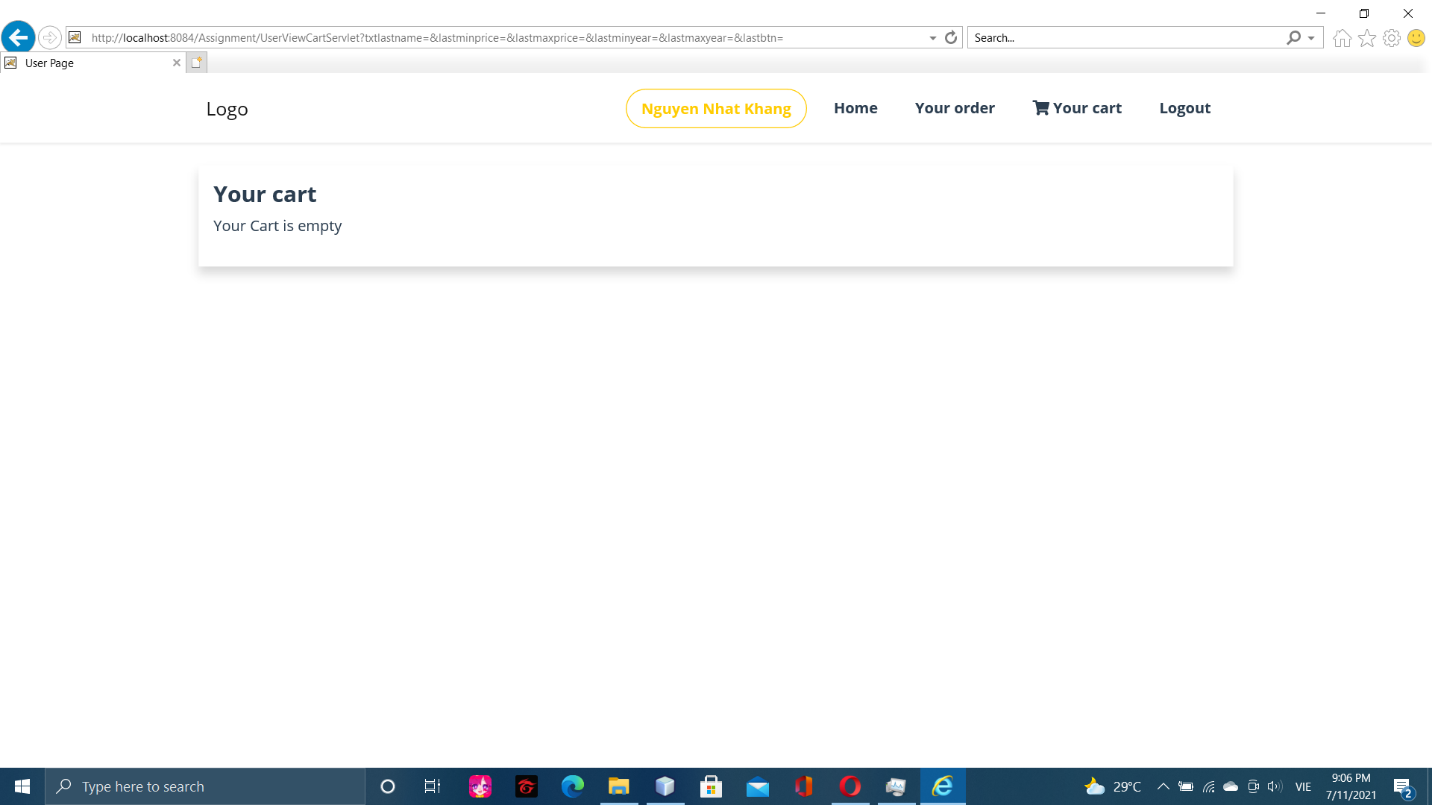
Add laptop to cart in Search page



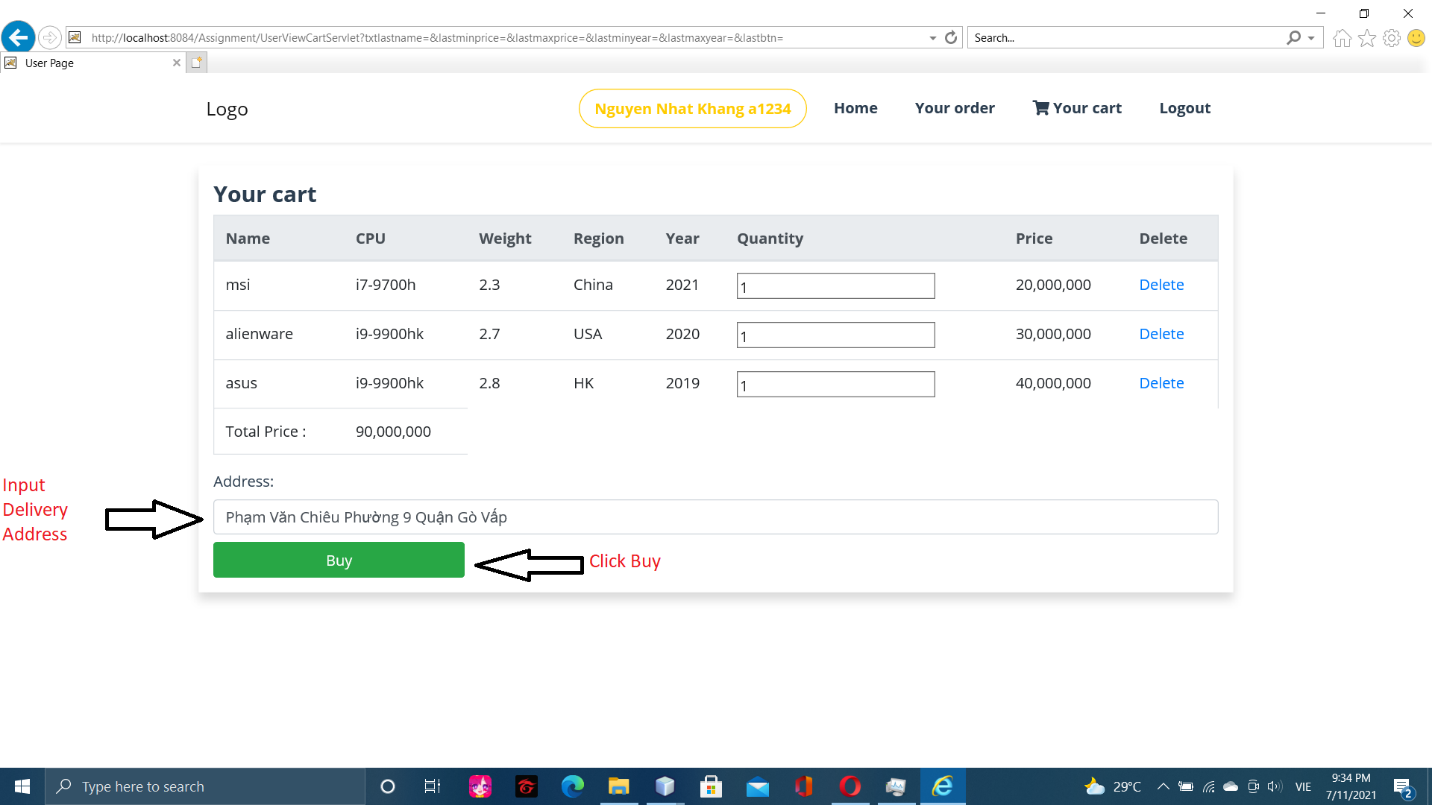
Add to cart in View detail page



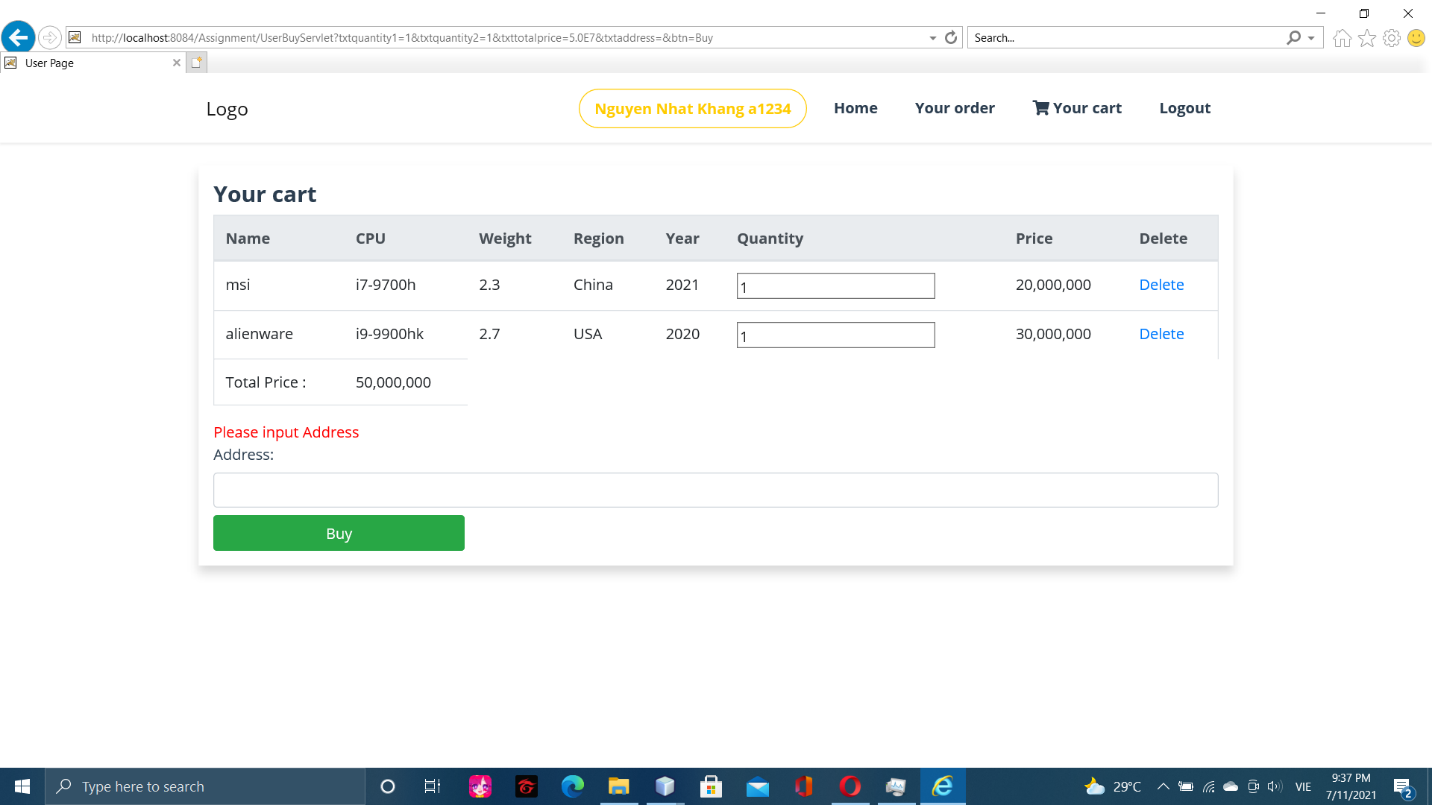
View cart



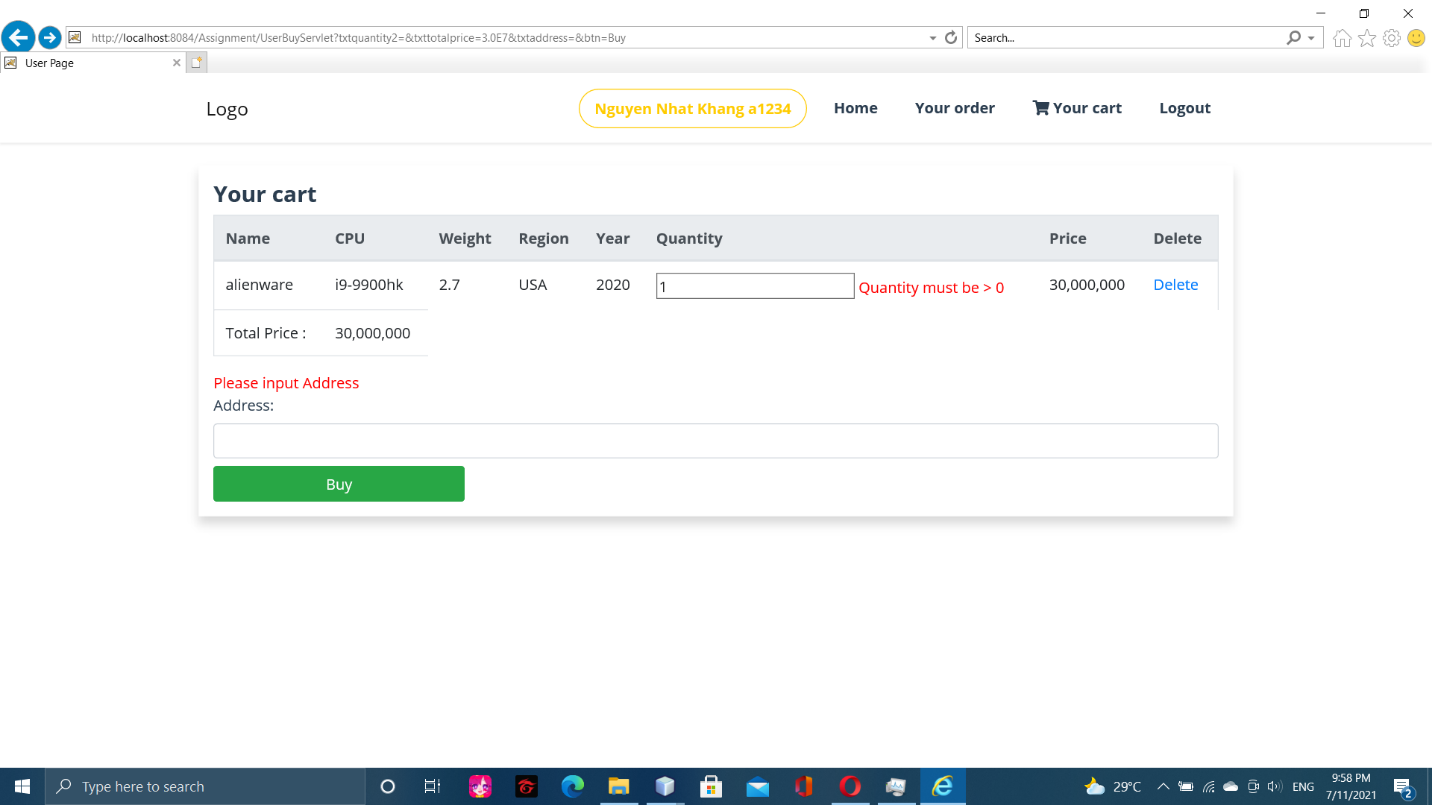
Empty cart



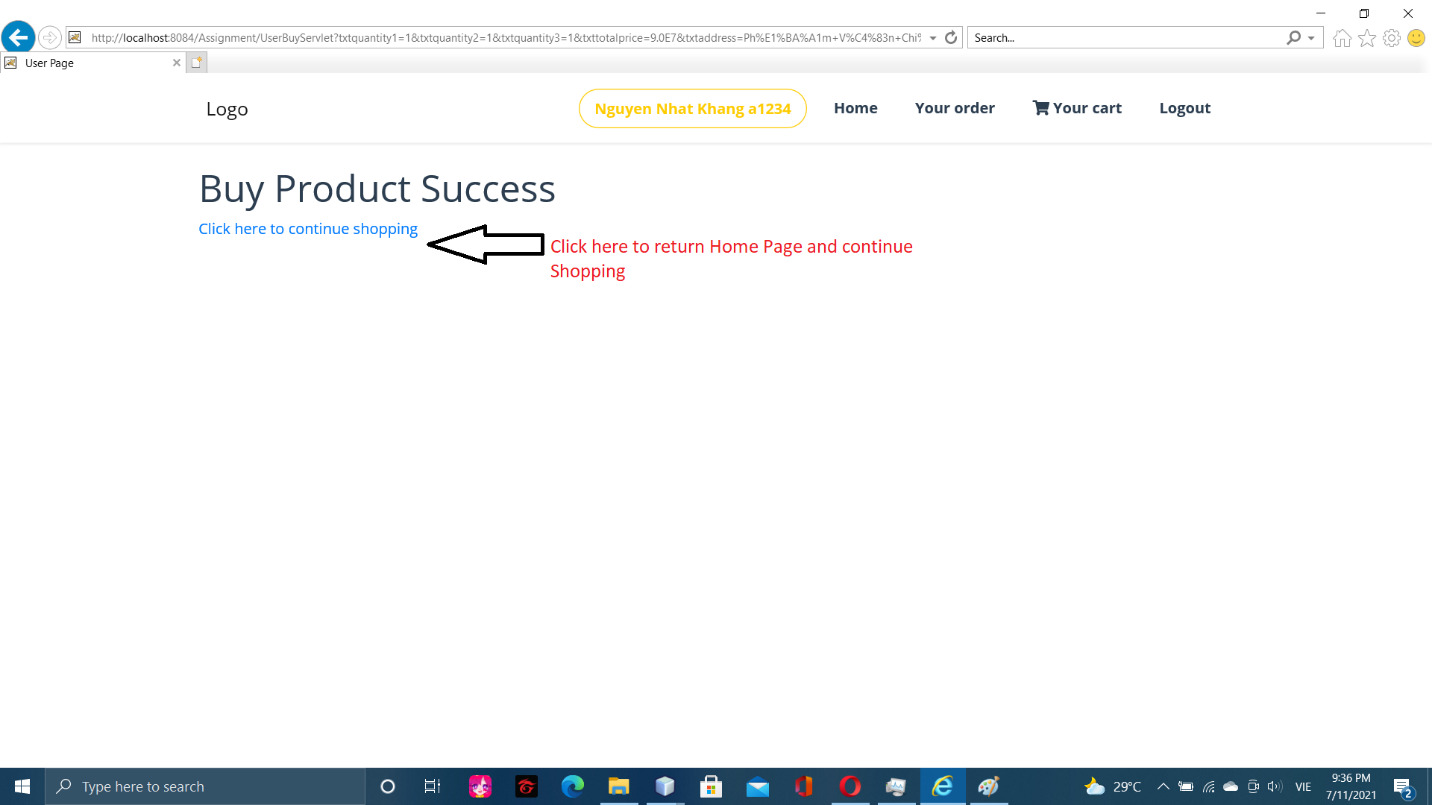
Buy laptop



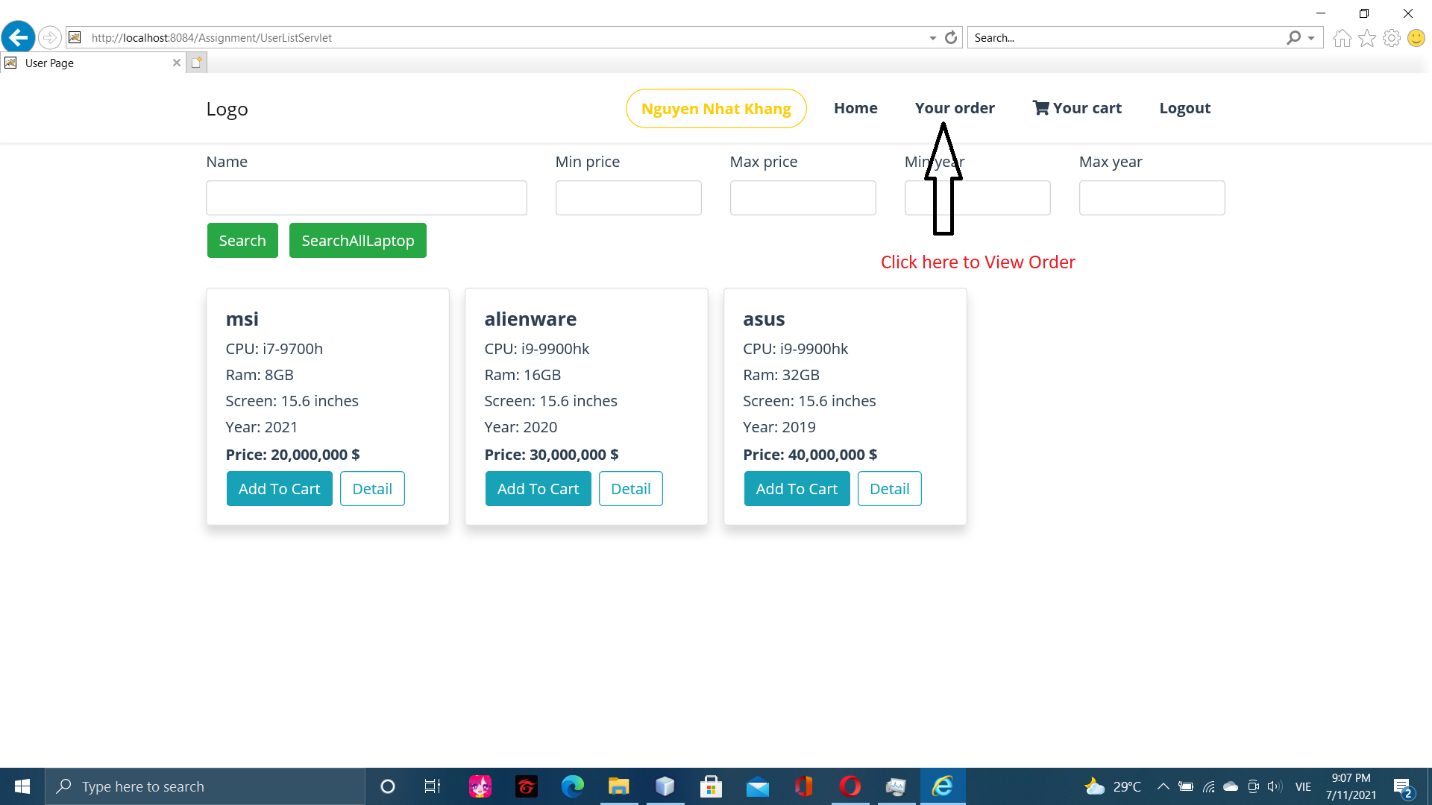
Address null error



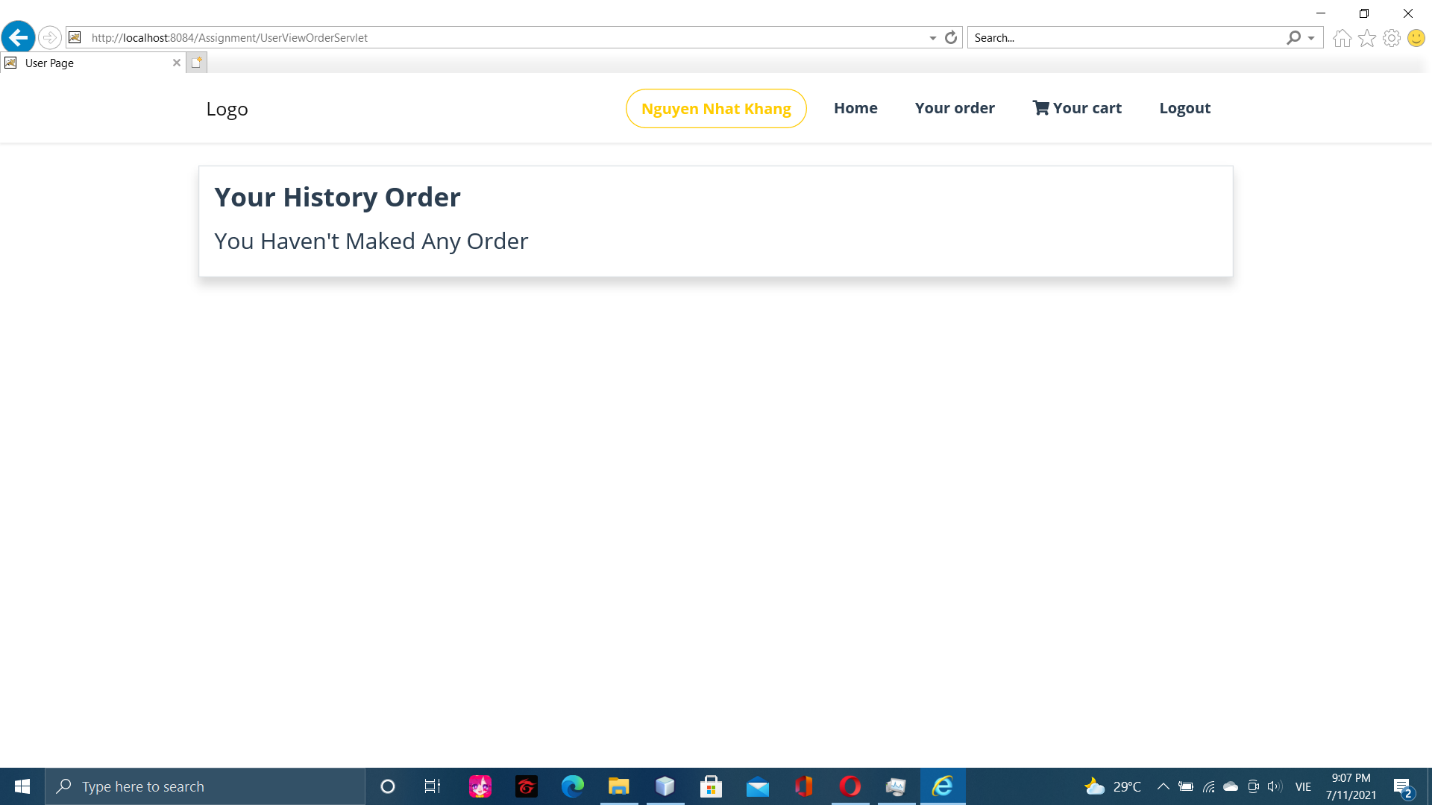
Buy Fail



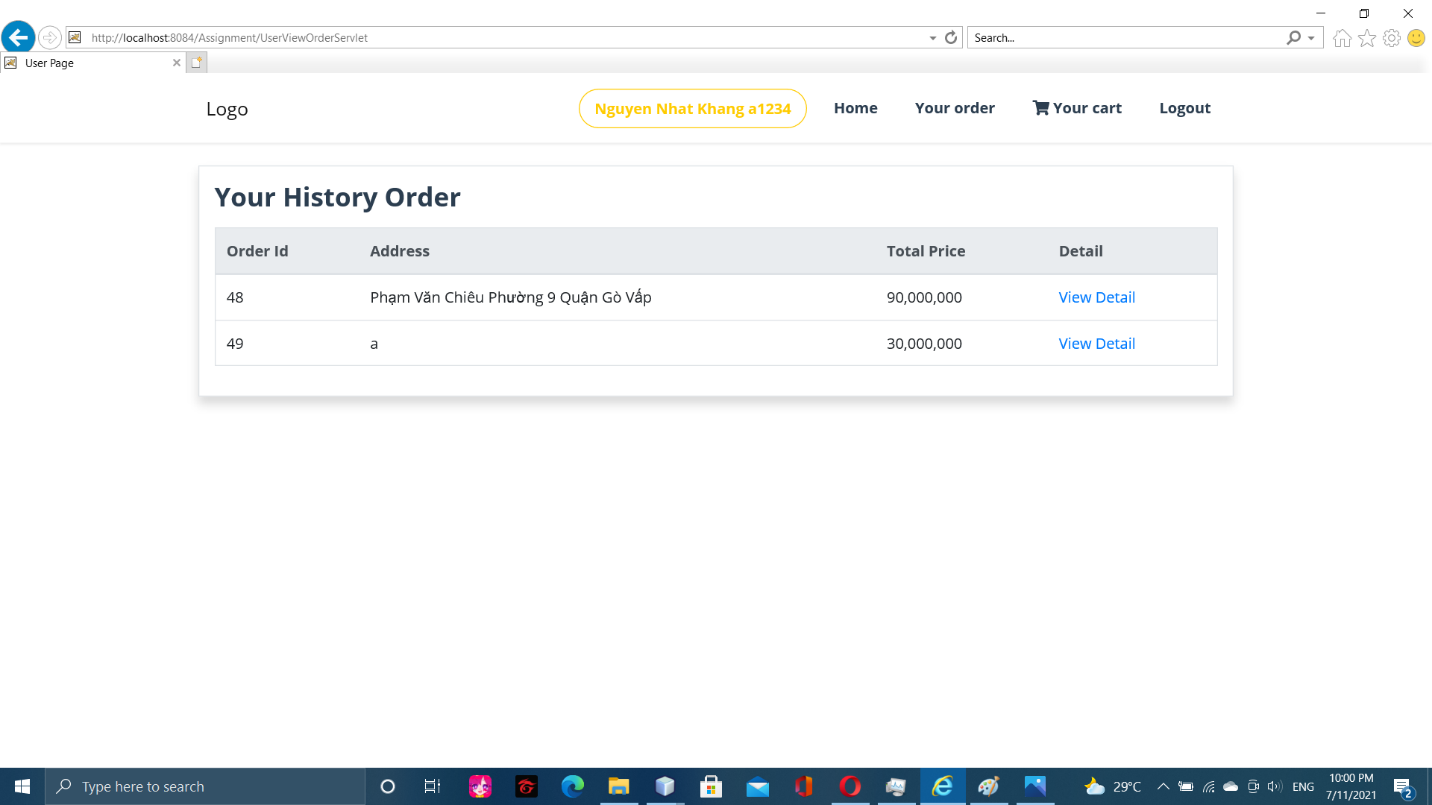
Buy Success



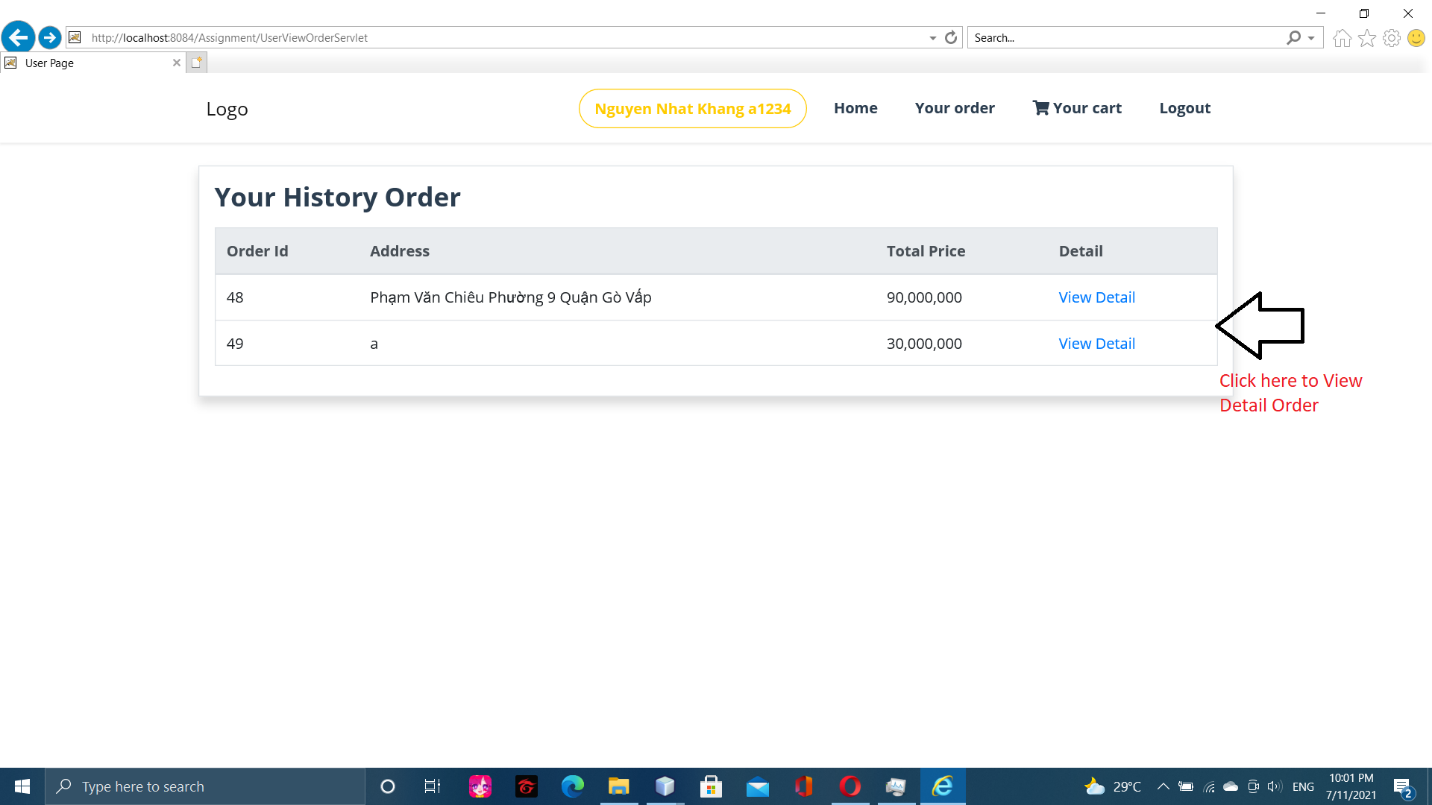
View history order



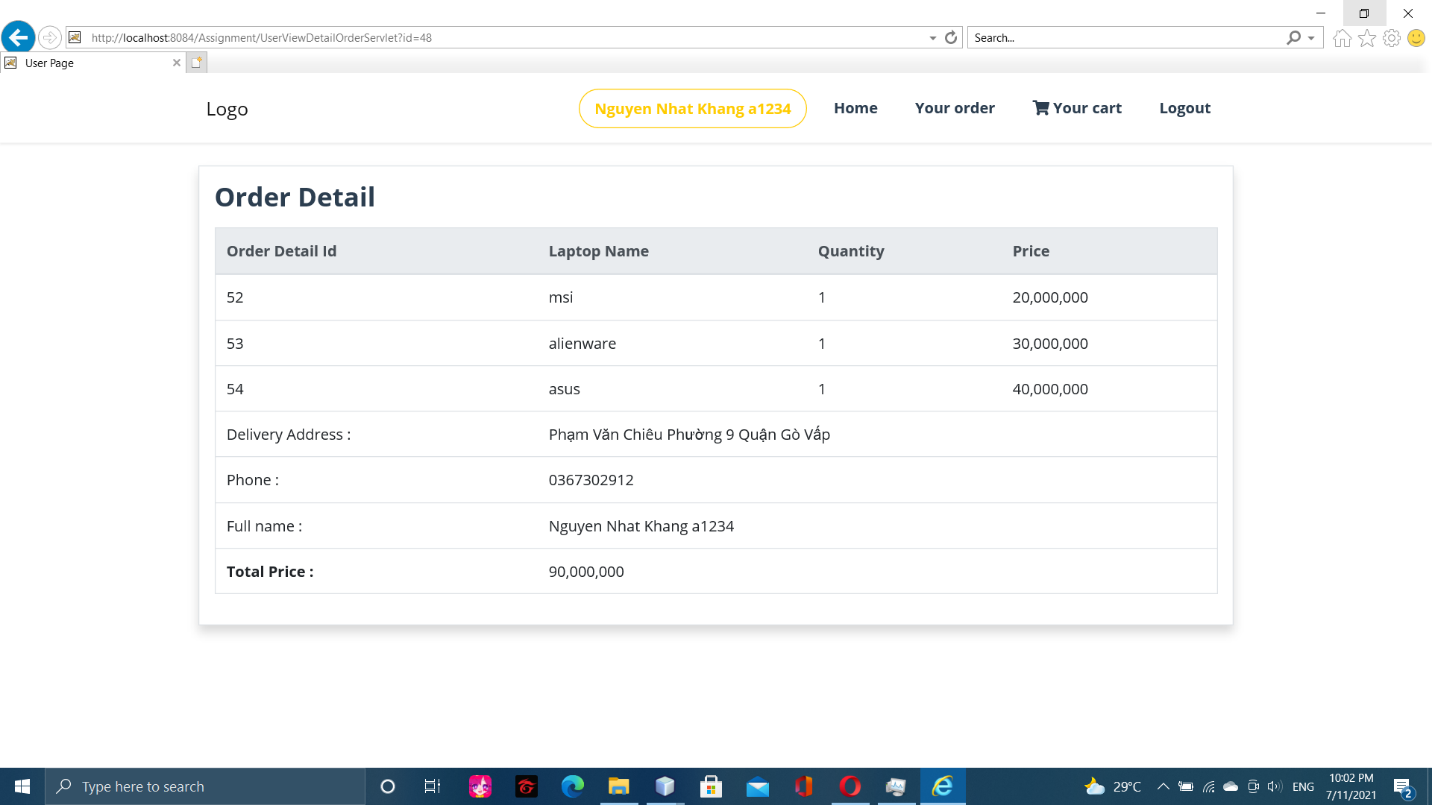
Empty history order



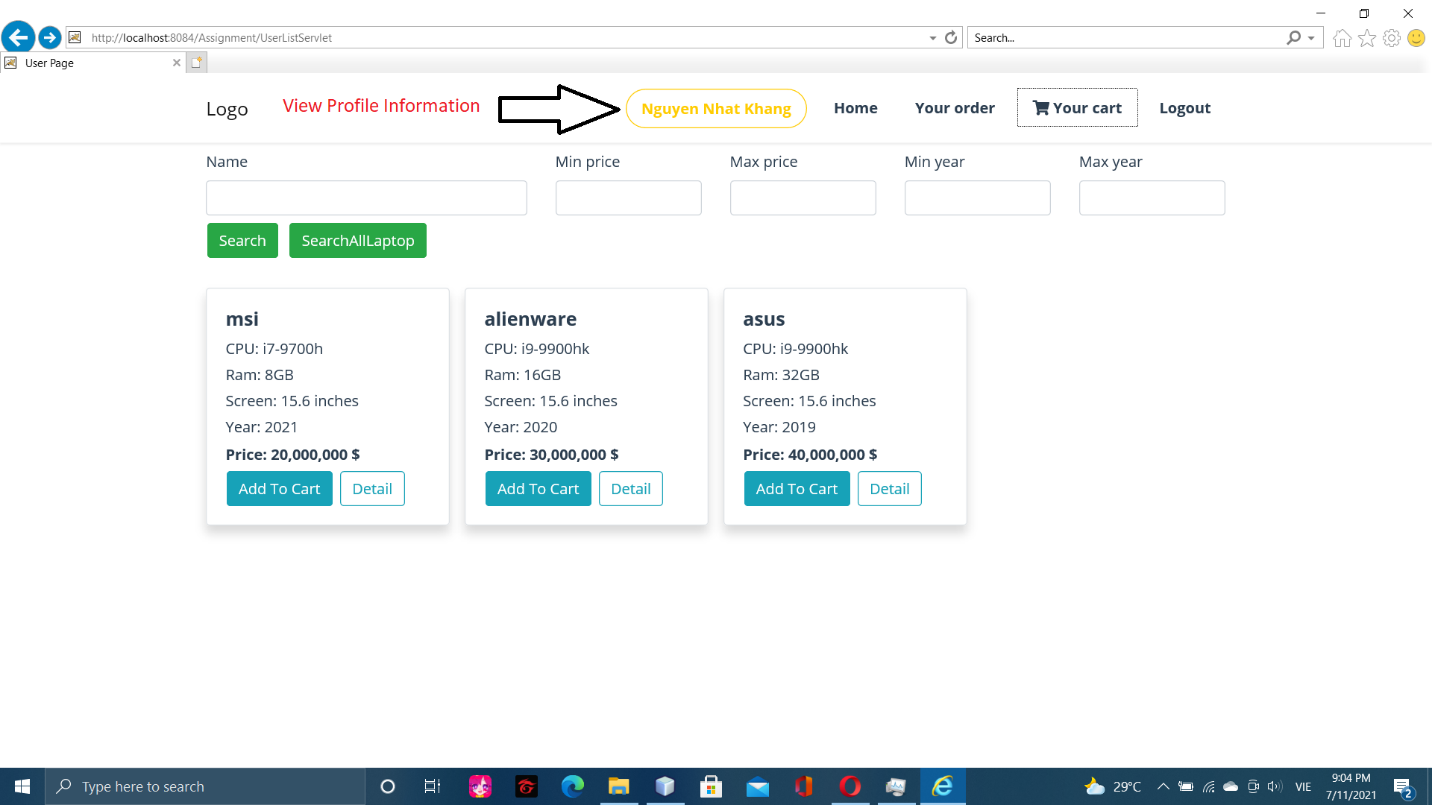
History order



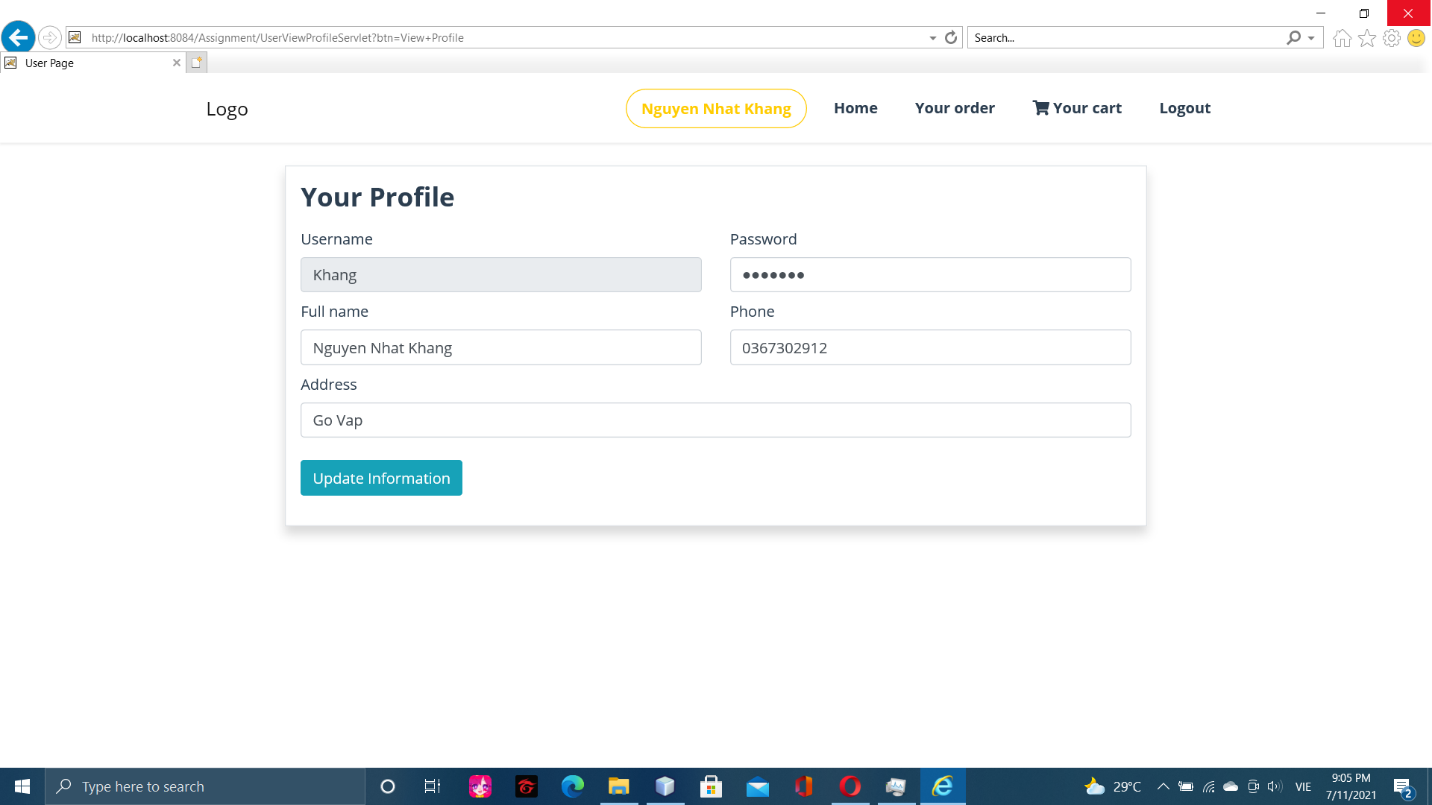
View detail order 1



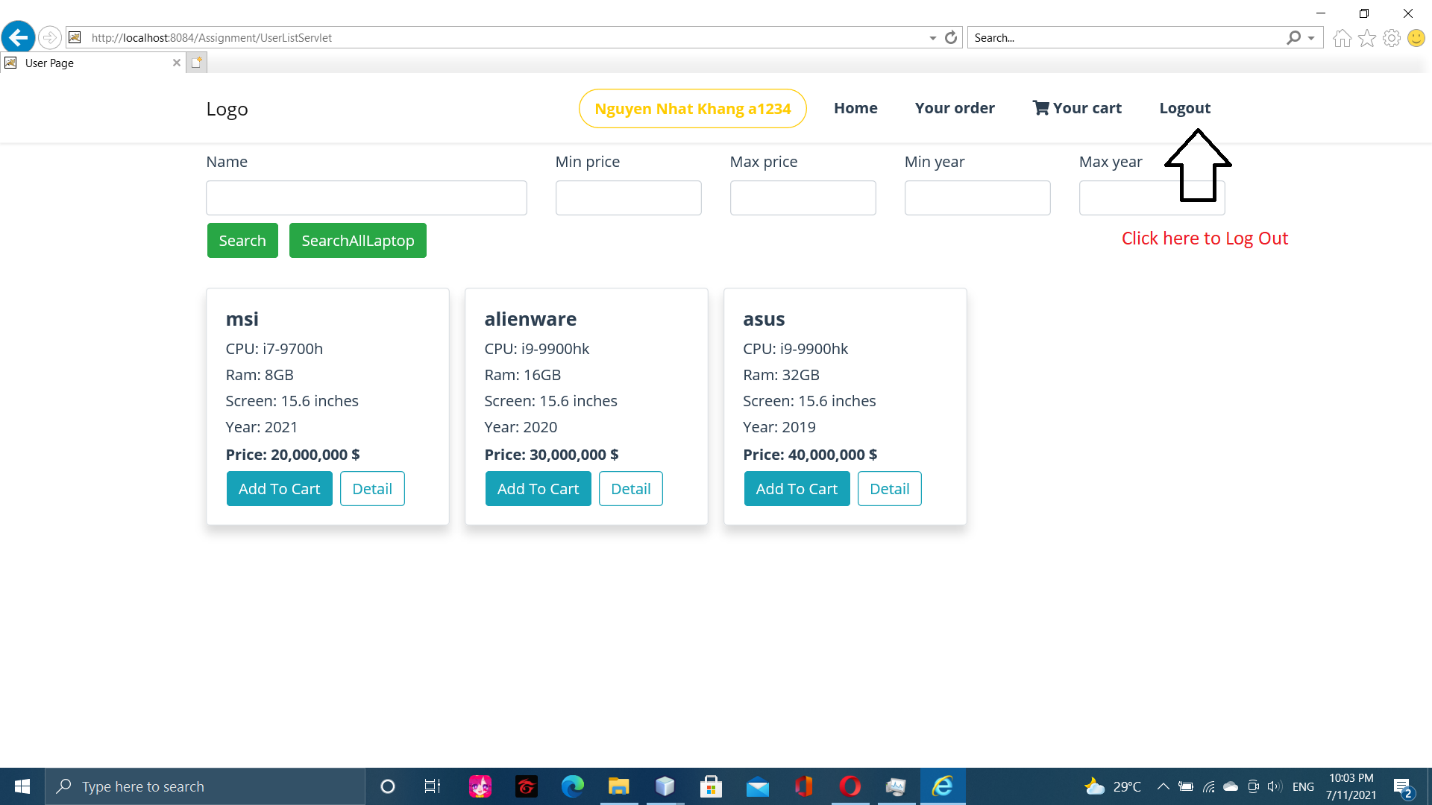
View detail order 2



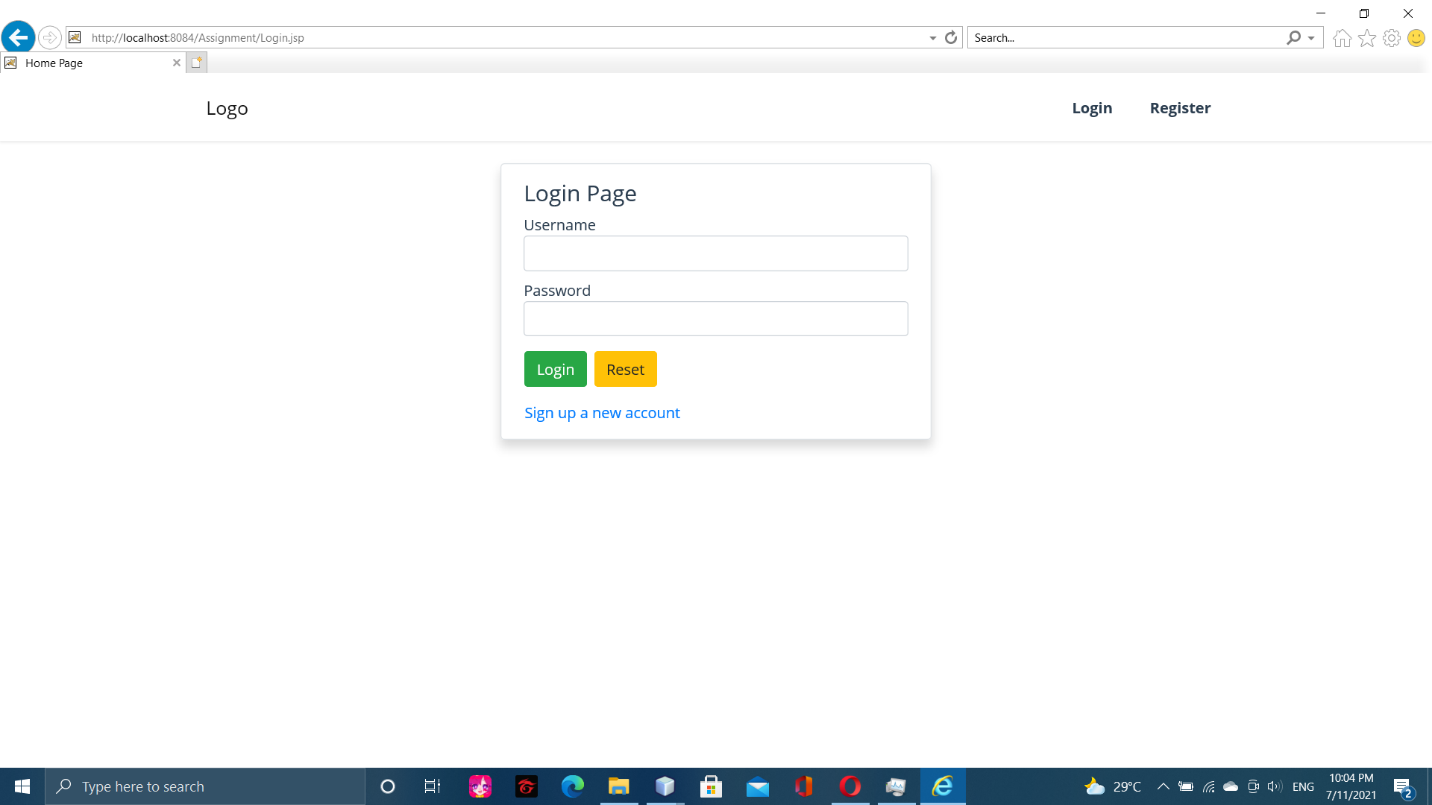
View profile



View profile 2

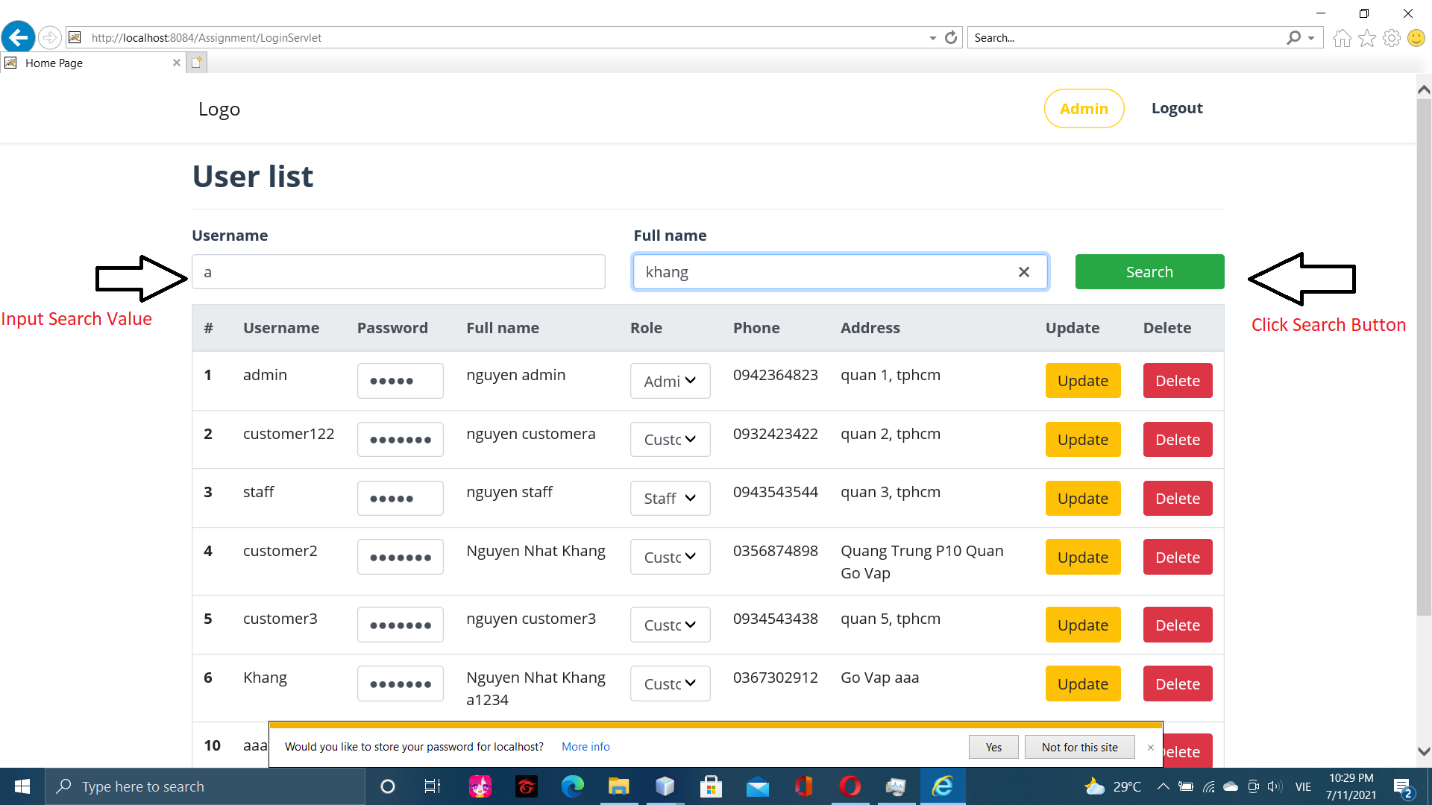


Logout

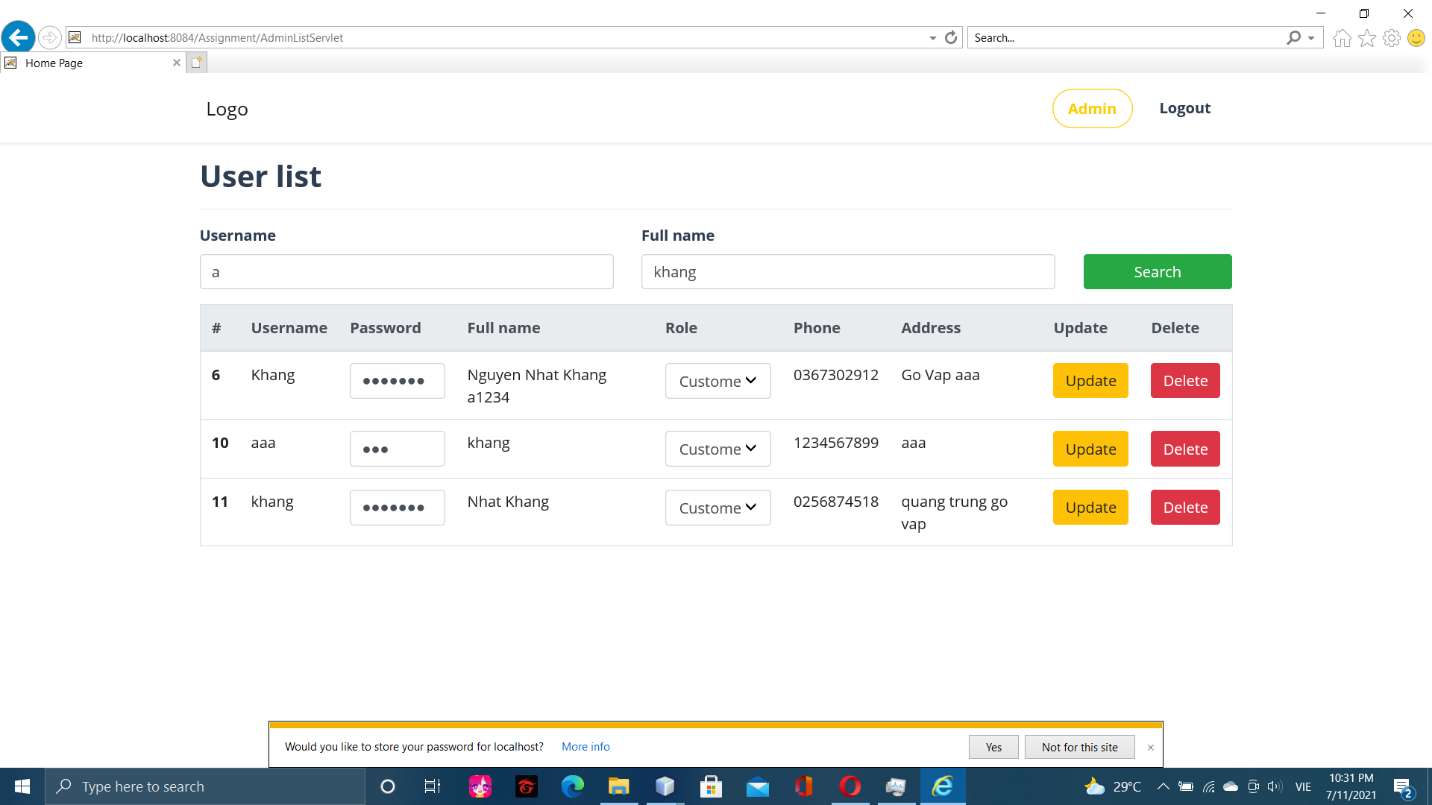


After click logout

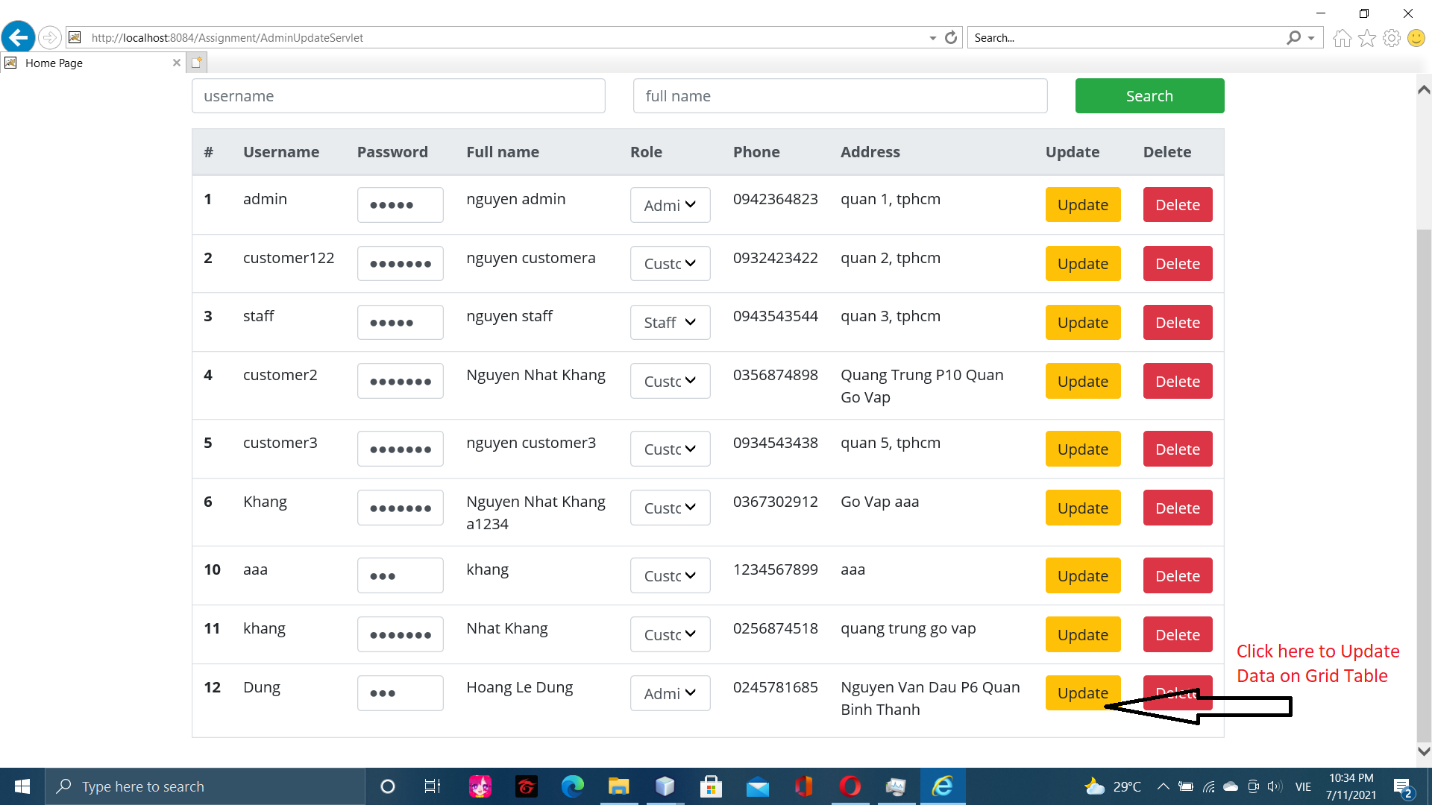
* 1. Admin



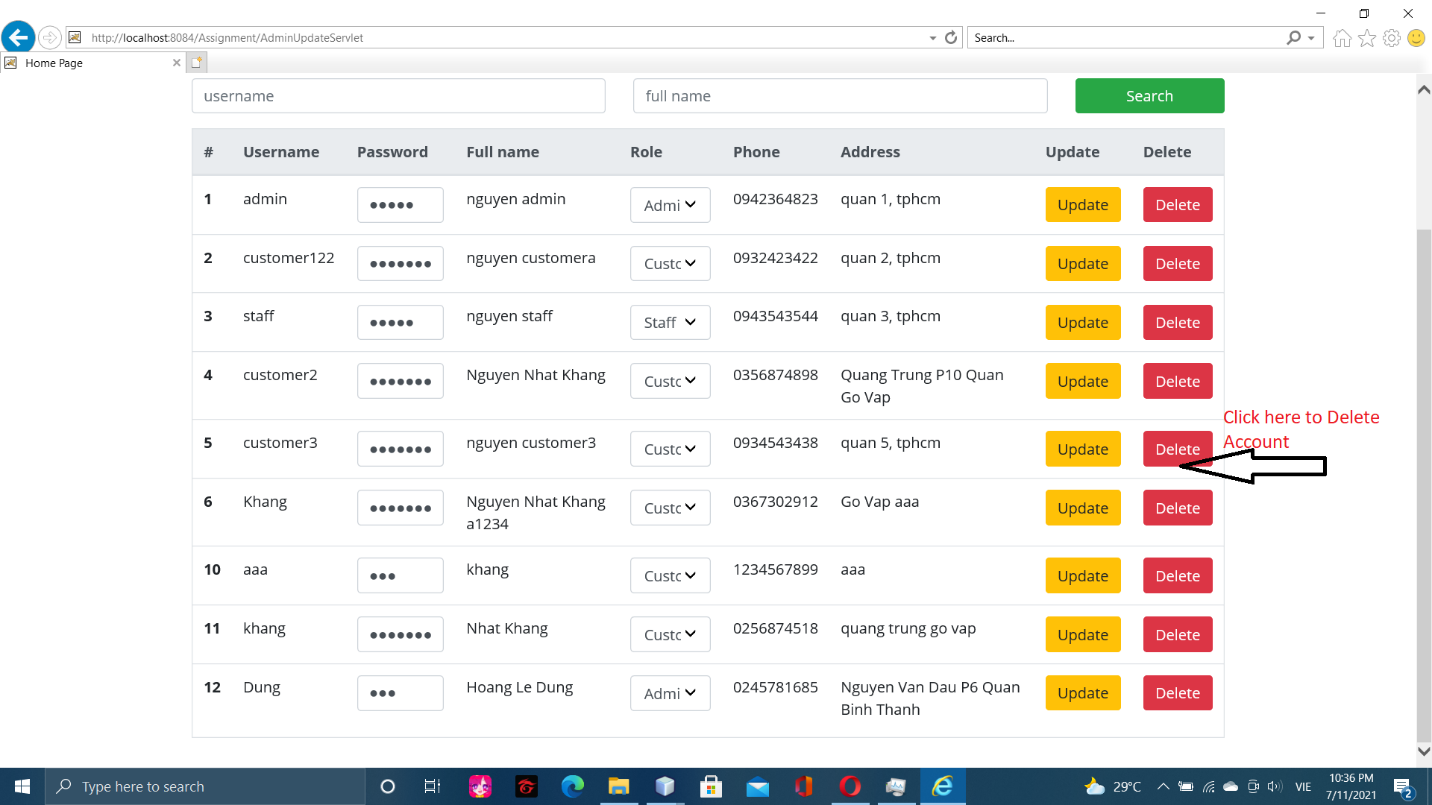
Search account list 1



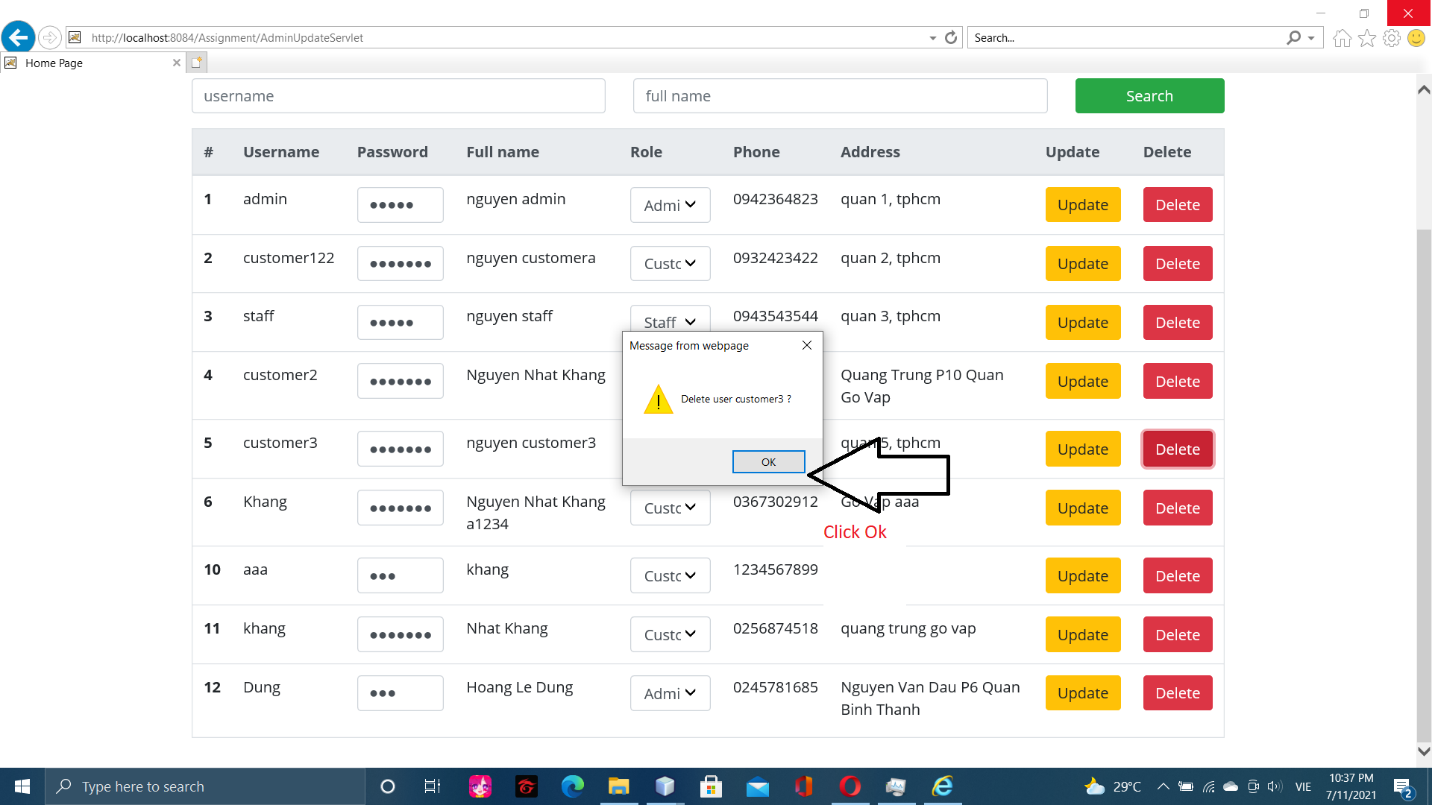
Search account list 2



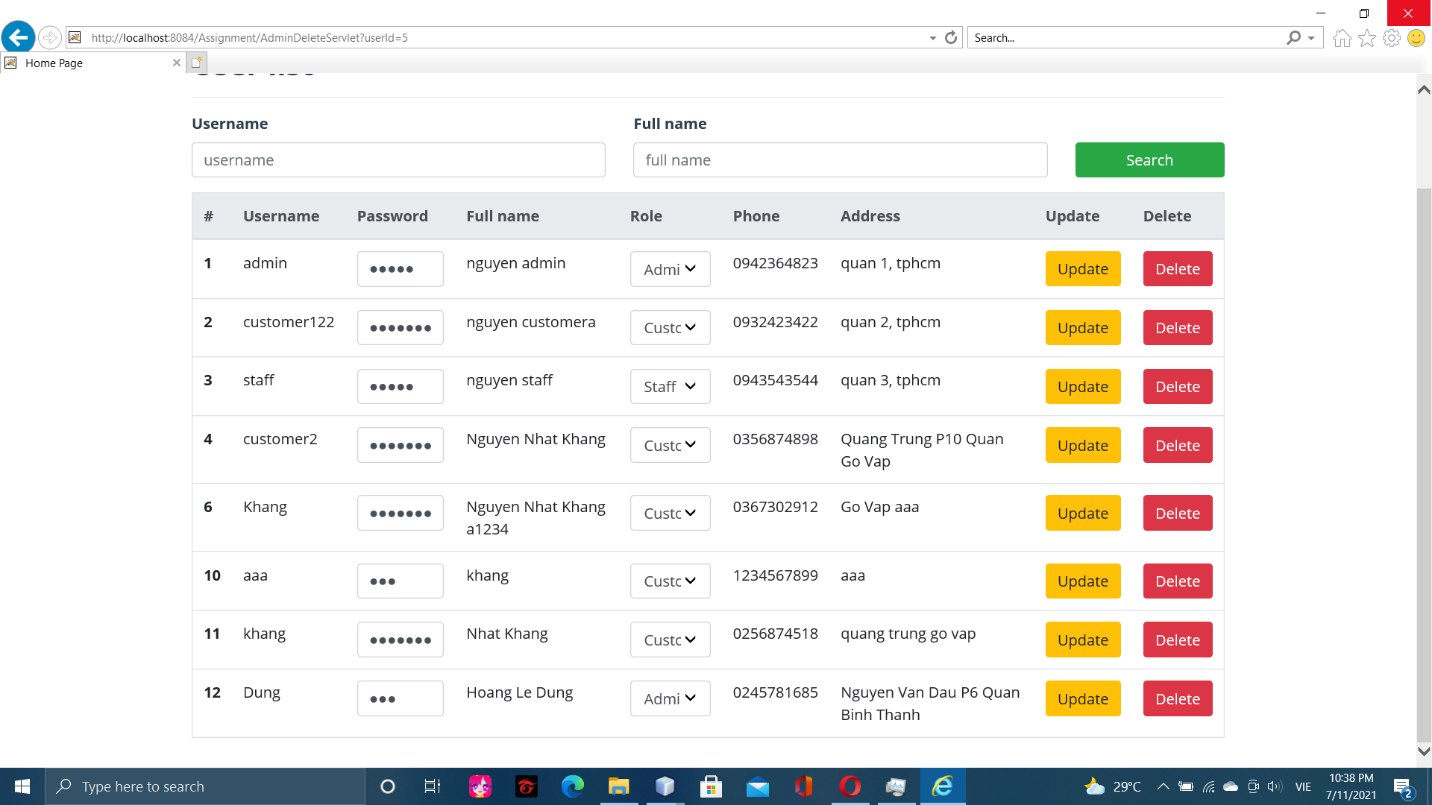
Update account



Delete account 1

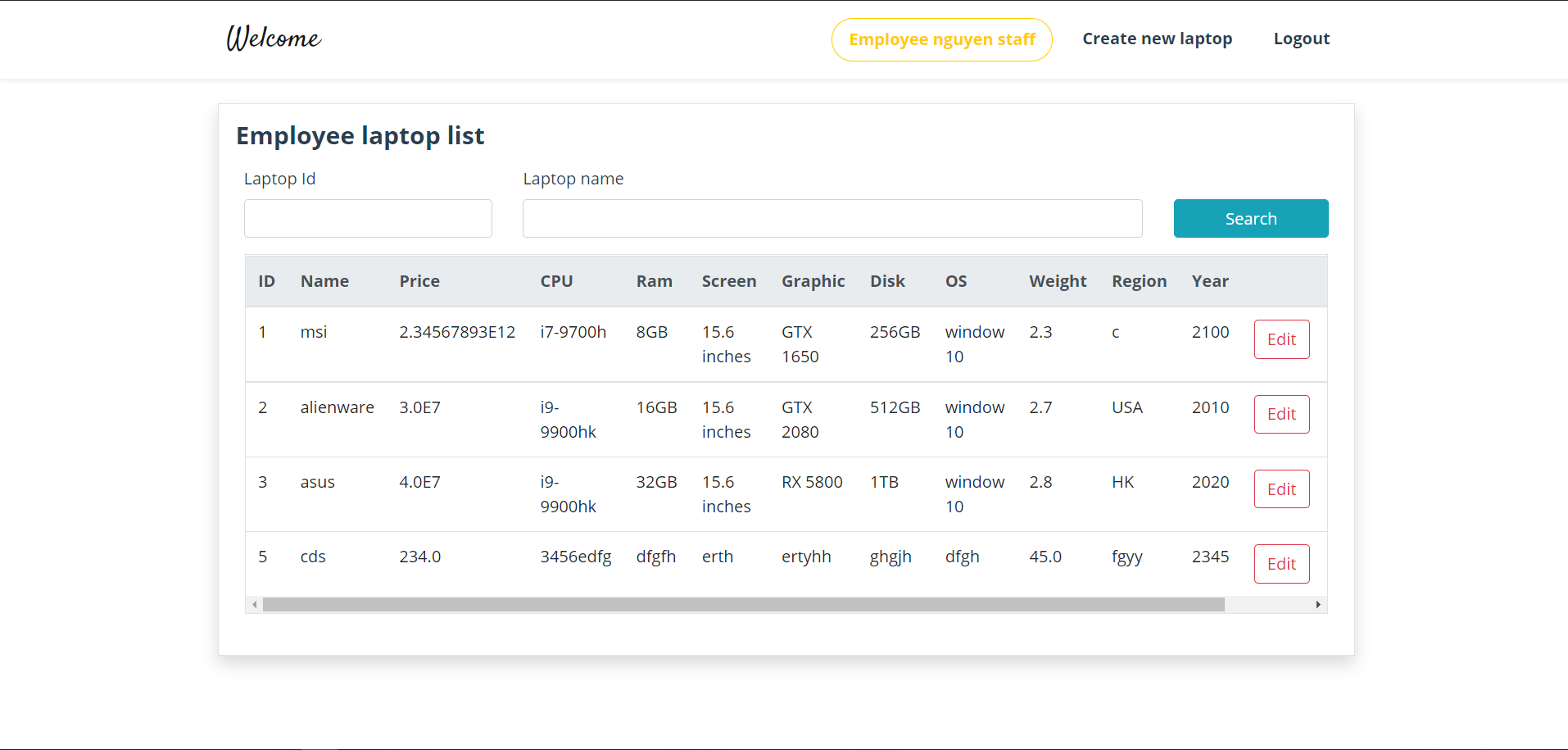


Delete account 2

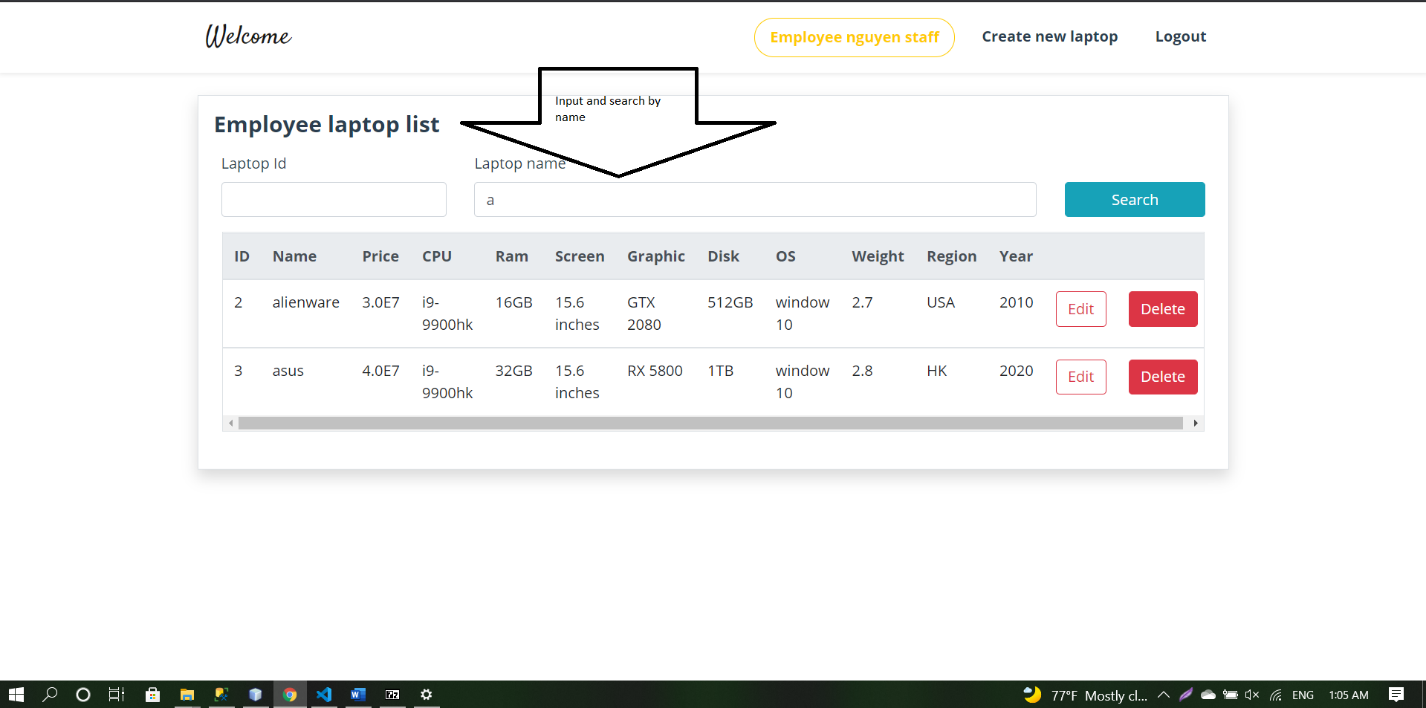


Delete account 3

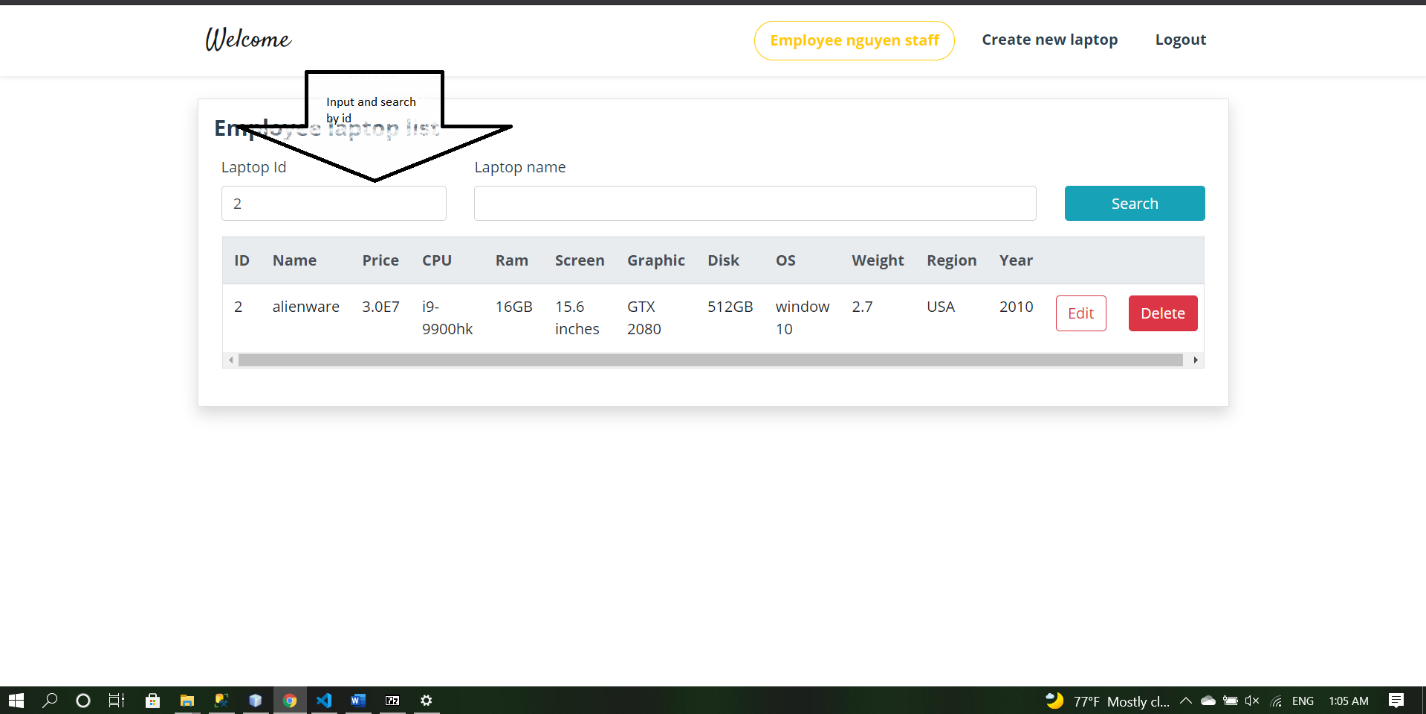
* 1. Employee



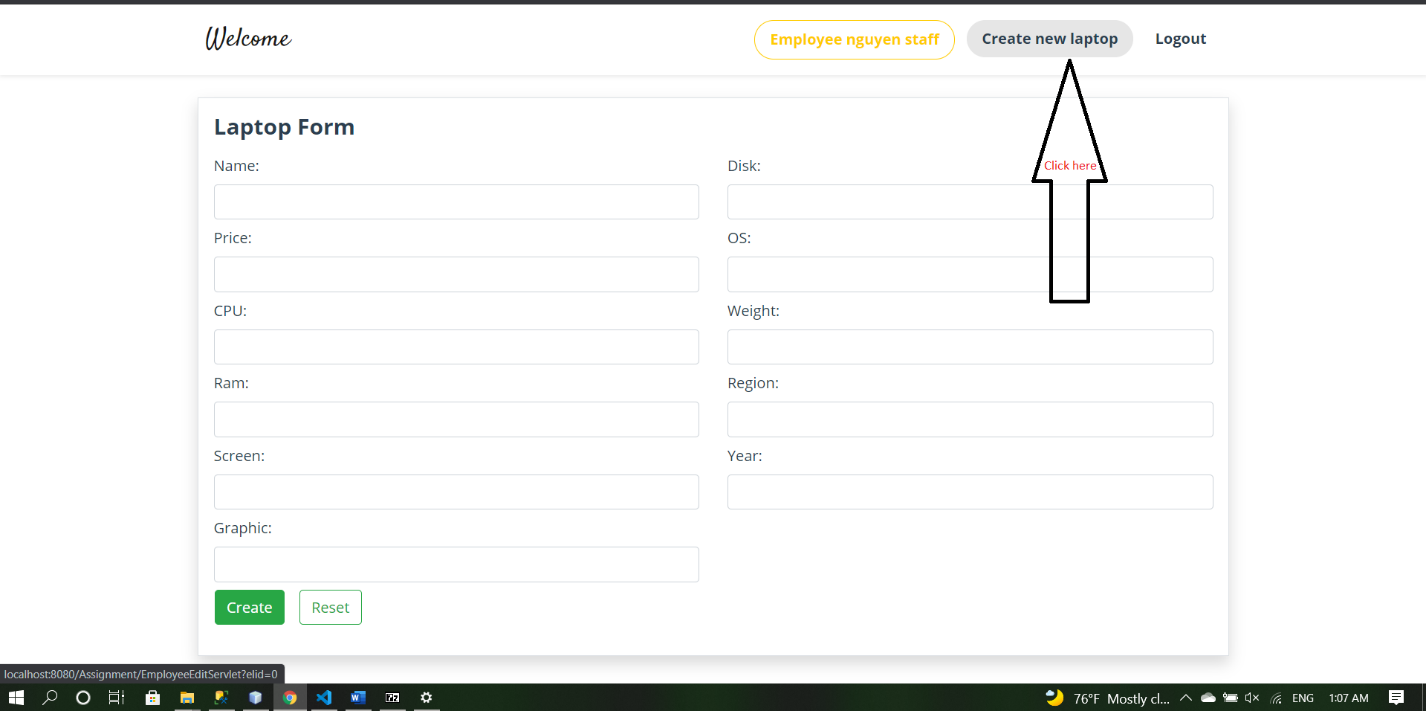
View all laptop



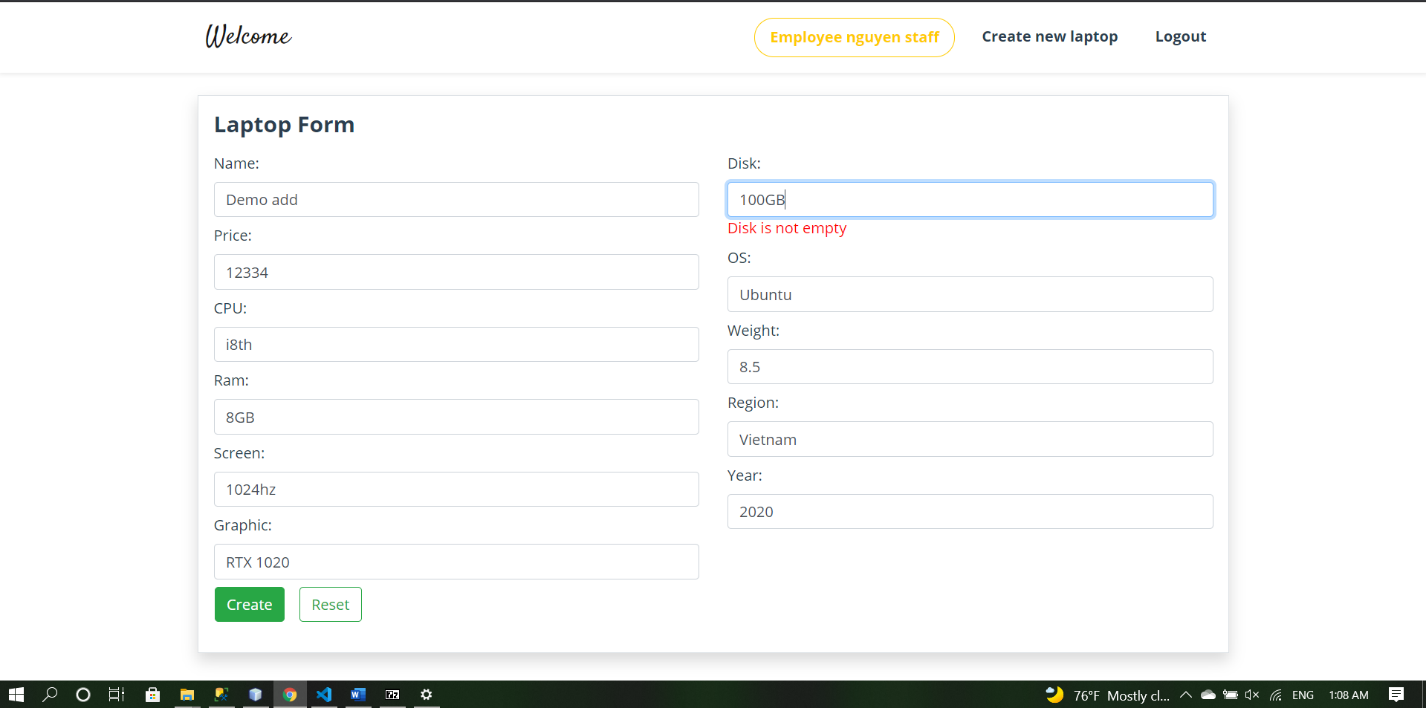
Search laptop list by name



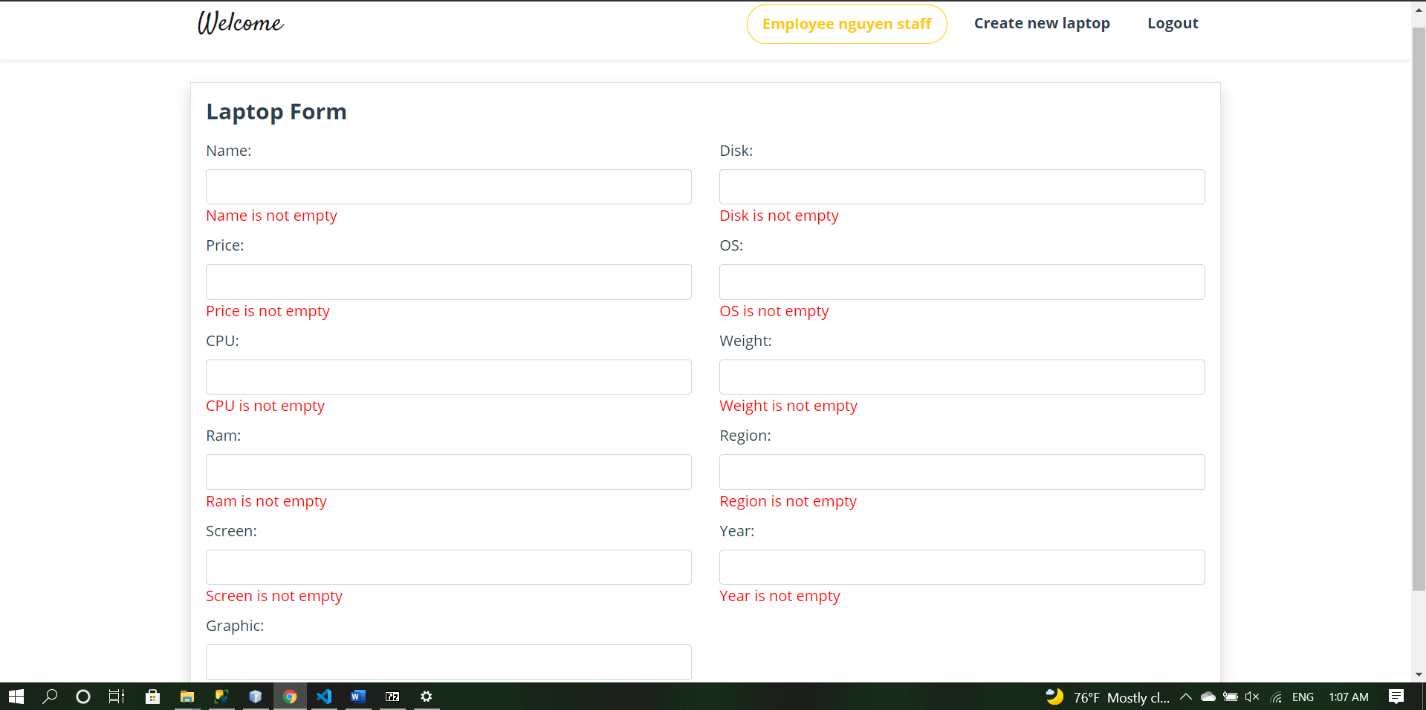
Search laptop list by id



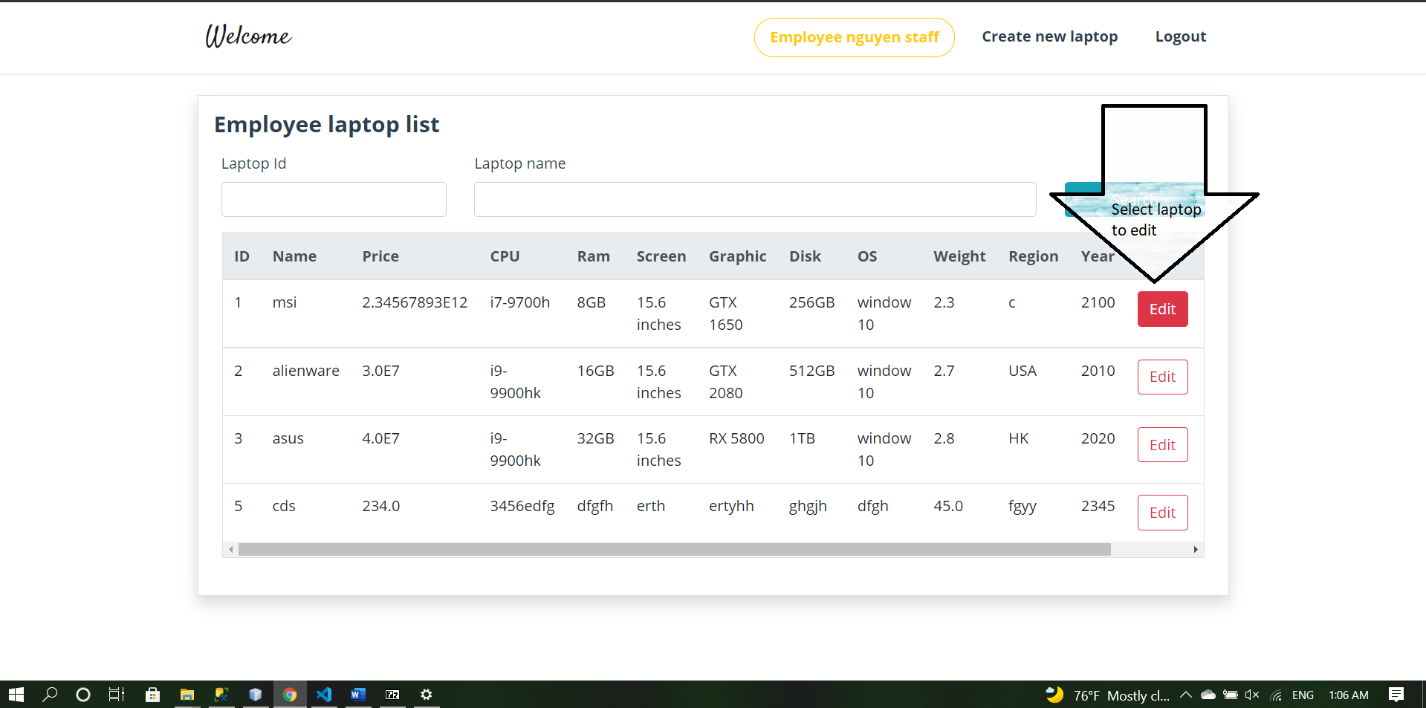
Create new laptop



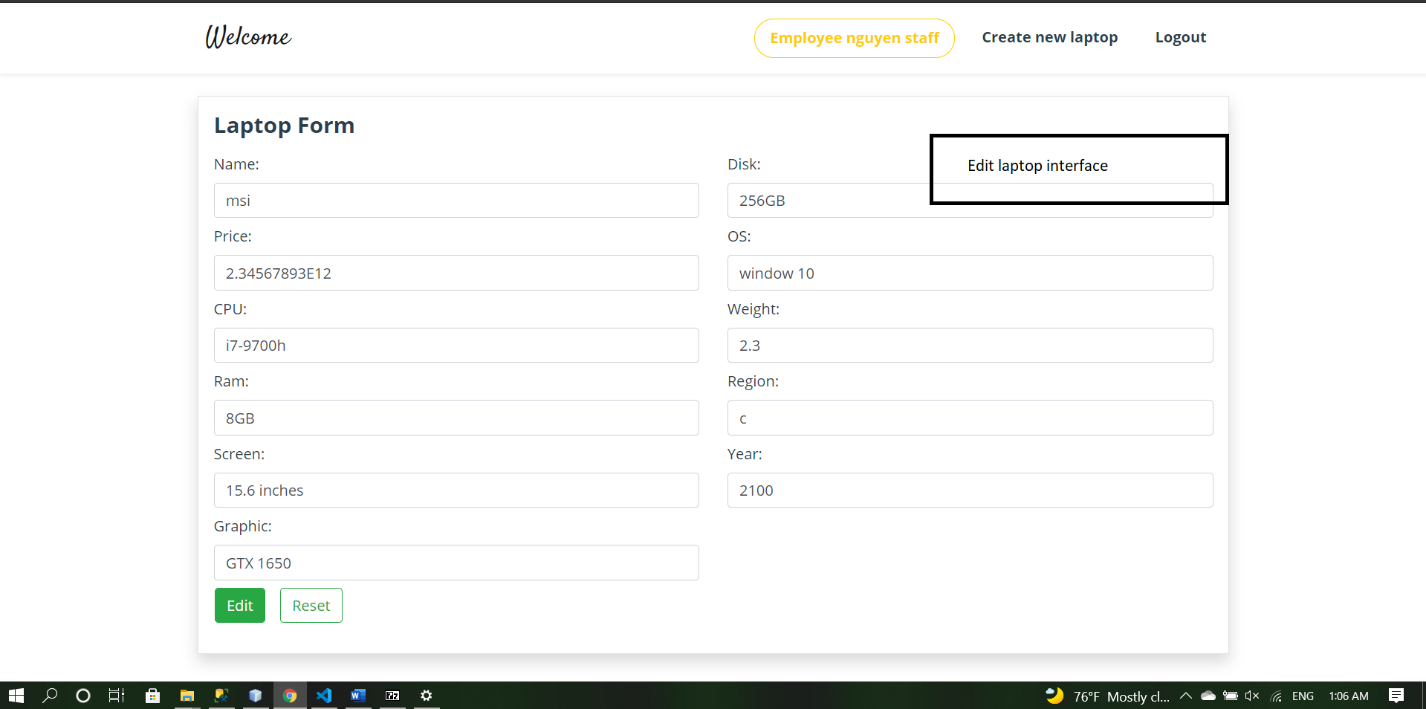
Input new laptop data



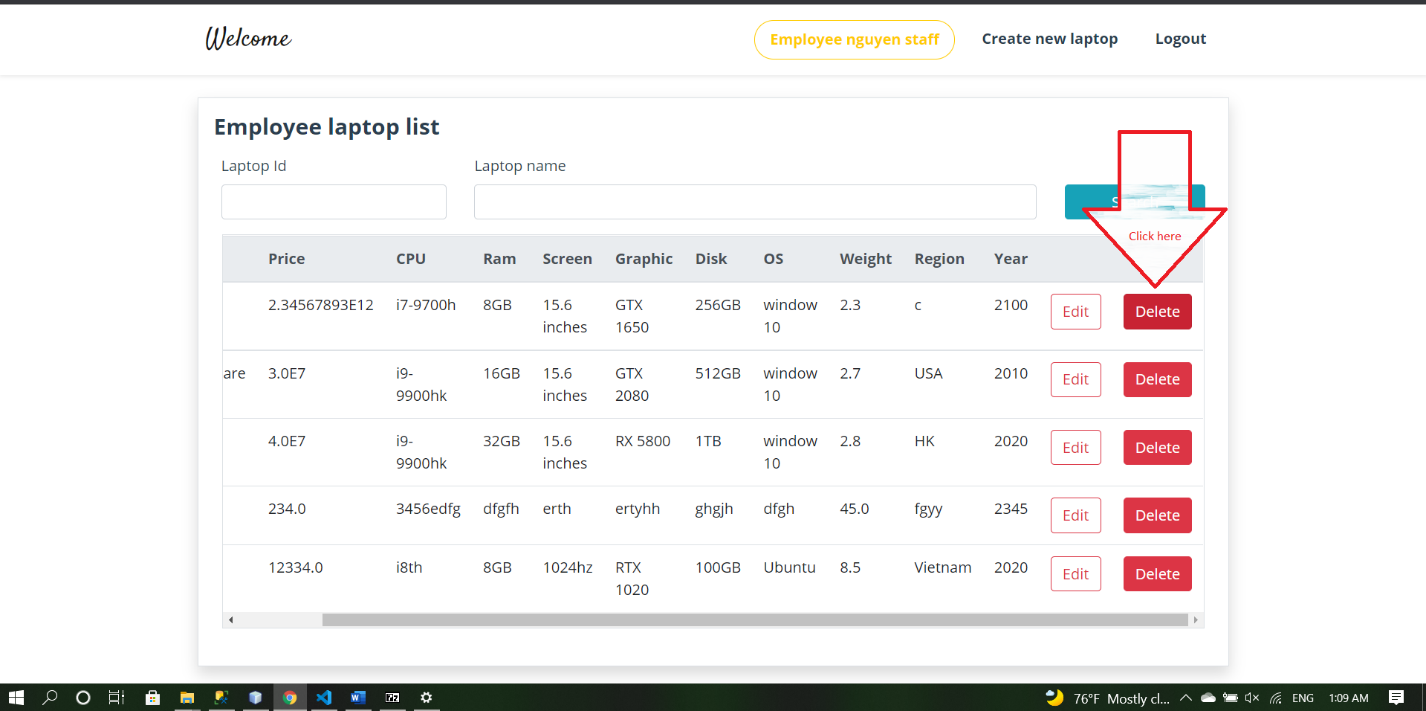
Show data error in Create new laptop page



Edit laptop



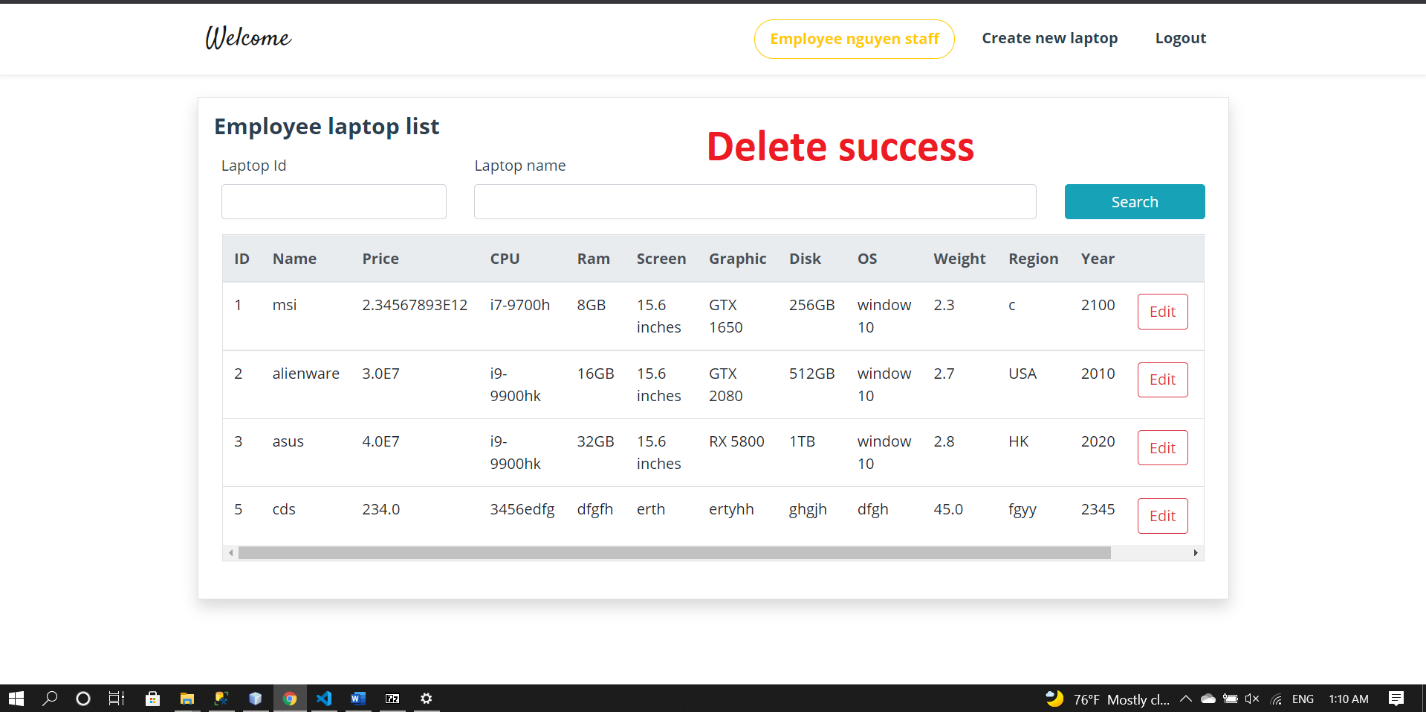
Edit laptop 2



Select laptop to delete



Confirm to delete



After delete

**V. Conclusion and Discussion**

* Overall, we have finished making a website selling laptop with some basic functions:

+ CRUD function for 3 role in the system (admin, employee, user)

+ Search list account, laptop.

+ Shopping cart, checkout make order

+ Some additional function such as edit profile for user.

+ Enhance UI by bootstrap framework

+ Learning git and github, making team-work more efficiently.

+ Friendly UI.

* Although the website we have made have all basic function for a website, but it is not ready to be use in market. Here are something that we should made in the future.

+ Encrypt password

+ Deploy website in cloud

+ Get more database

+ Provide payment method