

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CNTT VÀ TT

BÁO CÁO THỰC HÀNH NHÓM
MÔN: THỰC HÀNH CƠ SỞ DỮ LIỆU
THẦY NGUYỄN HỮU ĐỨC

ĐỀ TÀI: WEB BÁN ĐỒ UỐNG (CAFE)

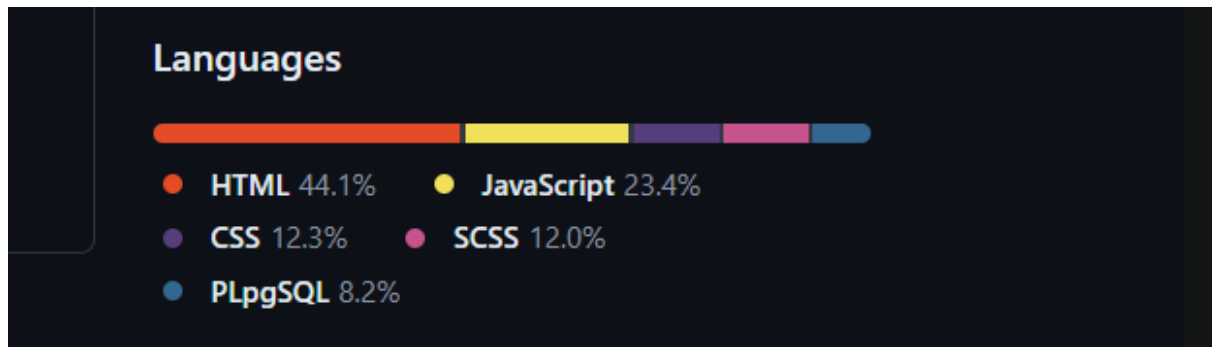
TRỊNH MINH ĐỨC 20225813

NGUYỄN KHÁNH DUY 20225830

NGUYỄN VĂN PHÚ 20225658

LỚP:IT3290 – 147780

I) CÔNG NGHỆ SỬ DỤNG



1) HTML,CSS,SCSS :

- HTML được sử dụng để xây dựng cấu trúc và nội dung của một trang web. Nó là ngôn ngữ đánh dấu chính được sử dụng trên web. Sử dụng Định nghĩa các phần tử của trang web như tiêu đề, đoạn văn, liên kết, hình ảnh, và các khối nội dung khác.
- CSS được sử dụng để kiểm soát giao diện và bố cục của các phần tử HTML. Nó giúp làm cho trang web trở nên đẹp mắt và hấp dẫn hơn. Định nghĩa các phong cách như màu sắc, font chữ, kích thước, khoảng cách, và các hiệu ứng khác cho các phần tử HTML. Giúp tách biệt nội dung (HTML) khỏi thiết kế (CSS), làm cho việc bảo trì và cập nhật trang web dễ dàng hơn. Hỗ trợ responsive design để trang web hiển thị tốt trên nhiều thiết bị khác nhau (máy tính, điện thoại, máy tính bảng).
- SCSS là một phần mở rộng của CSS, là một trong những cú pháp của SASS (Syntactically Awesome Stylesheets). Nó cung cấp các tính năng nâng cao giúp viết CSS dễ dàng và hiệu quả hơn. Hỗ trợ biến (variables), cho phép lưu trữ các giá trị tái sử dụng (như màu sắc, font chữ) để sử dụng lại trong toàn bộ stylesheet. Hỗ trợ mixins, cho phép tái sử dụng các khối CSS code, làm giảm sự lặp lại và tối ưu hóa code. cho phép viết CSS lồng nhau giống như cấu trúc HTML, giúp mã nguồn dễ đọc và gọn gàng hơn.

2) JS : một ngôn ngữ lập trình được sử dụng rộng rãi để tạo ra các trang web tương tác và động.

- Tạo tính tương tác cho trang web:
 - **Event Handling:** JavaScript cho phép xử lý các sự kiện như nhấp chuột, di chuyển chuột, nhập liệu từ bàn phím, và các sự kiện khác của người dùng để làm cho trang web trở nên tương tác hơn.
 - **Dynamic Content Updates:** JavaScript có thể cập nhật nội dung của trang web mà không cần phải tải lại trang, giúp cải thiện trải nghiệm người dùng.
- Thao tác với DOM (Document Object Model):
 - **DOM Manipulation:** JavaScript có thể thay đổi, thêm, hoặc xóa các phần tử HTML trên trang, thay đổi thuộc tính và nội dung của các phần tử, và điều chỉnh kiểu dáng của chúng.
 - JavaScript được sử dụng để kiểm tra và xác thực dữ liệu nhập liệu từ các biểu mẫu trước khi gửi về máy chủ.

- Tạo hiệu ứng và hoạt ảnh:
 - **Animations:** JavaScript có thể được sử dụng để tạo ra các hiệu ứng hoạt ảnh, làm cho trang web trở nên sinh động hơn.
 - **Transitions:** Các hiệu ứng chuyển đổi mượt mà giữa các trạng thái khác nhau của phần tử HTML có thể được thực hiện bằng JavaScript.
 - Giao tiếp với máy chủ:
 - **AJAX (Asynchronous JavaScript and XML):** JavaScript có thể gửi và nhận dữ liệu từ máy chủ mà không cần tải lại trang, cho phép cập nhật nội dung trang web động.
 - **Axios API:** Giao diện lập trình ứng dụng hiện đại cho phép thực hiện các yêu cầu HTTP dễ dàng hơn so với AJAX.
 - Ứng dụng phía máy khách (Client-Side):
 - **Frameworks and Libraries:** JavaScript là cơ sở của nhiều framework và thư viện phổ biến như React, Angular, Vue.js, giúp xây dựng các ứng dụng web phức tạp dễ dàng hơn.
 - Tạo và sử dụng các API.
- 3) Express JS :** Express.js là một framework web ứng dụng dành cho Node.js, được thiết kế để giúp xây dựng các ứng dụng web và API một cách dễ dàng và nhanh chóng
- Xây dựng ứng dụng web và API:
 - Express.js cung cấp một bộ công cụ mạnh mẽ để xây dựng các ứng dụng web và API RESTful. Nó giúp quản lý các yêu cầu HTTP và phản hồi một cách hiệu quả.
 - Quản lý routing:
 - Express.js cho phép bạn xác định các route khác nhau trong ứng dụng của bạn để xử lý các yêu cầu từ người dùng. Nó hỗ trợ routing dựa trên URL và phương thức HTTP (GET, POST, PUT, DELETE, v.v.).
 - Xử lý các yêu cầu HTTP:
 - Express.js giúp xử lý các yêu cầu HTTP (GET, POST, PUT, DELETE, v.v.) một cách dễ dàng. Nó cung cấp các phương thức đơn giản để đọc dữ liệu từ yêu cầu và gửi dữ liệu trong phản hồi.
 - Tạo và phục vụ các trang HTML.
- 4) PostgreSQL :** Nó được sử dụng để lưu trữ, quản lý, và truy vấn dữ liệu trong các ứng dụng phần mềm
- Quản lý dữ liệu:
 - Hỗ trợ các ràng buộc dữ liệu và toàn vẹn dữ liệu:
 - Hỗ trợ các giao dịch (transactions):
 - Truy vấn dữ liệu mạnh mẽ:

II) Mô hình sử dụng cho FE,BE

1) Tổ chức CSS files với pattern 7-1 dành cho FE

JS	SignUp.js	21
▼	scss	22
>	abstracts	23
>	base	24
>	components	25
>	layout	26
>	pages	27
	main.css	28
	main.css.map	29
	main.scss	30
		31
		32

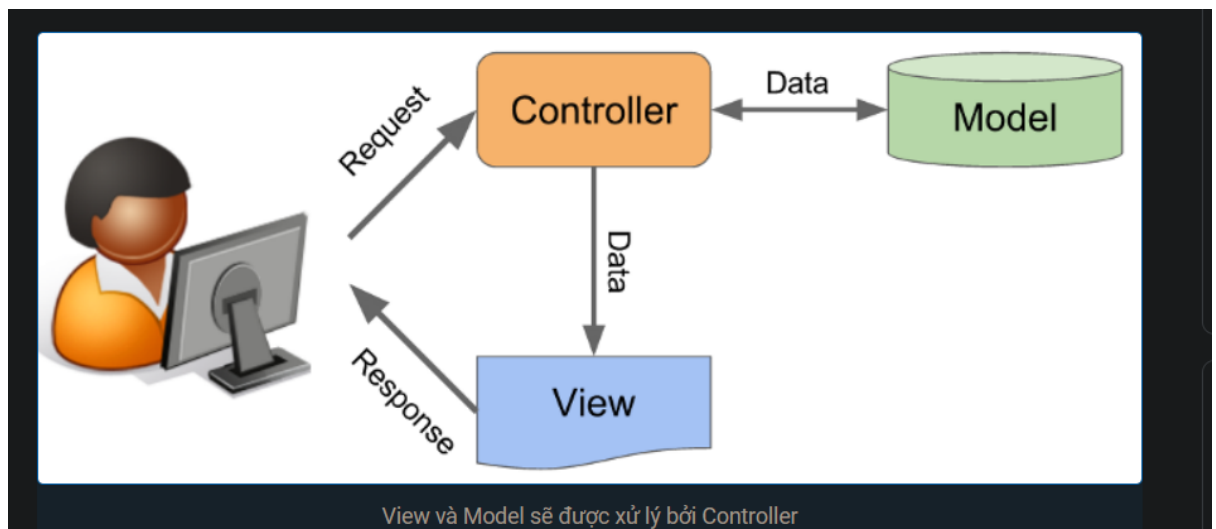
- **base:** Nơi chứa tất cả code CSS boilerplate của bạn. CSS Boilerplate ở đây bạn có thể hiểu là code CSS bạn sẽ viết mỗi lần start một dự án mới. Ví dụ: quy tắc typography, animations, utilities v.v..
- **components:** Cái tên khá là tường minh rồi. Thư mục này chứa tất cả các thành phần (components) CSS của bạn được sử dụng để tạo nên các trang như buttons, forms, swipers, popups, v.v..
- **layout:** Được sử dụng để bố trí các phần khác nhau của trang, có nghĩa là header, footer, navigation, section, grid do chính bạn định nghĩa v.v..
- **pages:** Ứng dụng của bạn sẽ có rất nhiều page, mỗi page lại có những style khác nhau, và đây chính là nơi bạn sẽ put CSS page của mình vào.
- **themes:** Nếu trong ứng dụng của bạn có nhiều themes khác nhau (dark mode, admin v.v..), hãy đặt chúng trong thư mục này.
- **abstracts:** Đặt tất cả các function CSS của bạn ở đây, cùng với variables và mixin. (Hiểu nôm na chúng là CSS helpers của bạn đấy)
- **vendors:** Nơi bạn chứa tất cả các file không phụ thuộc nếu ứng dụng của bạn sử dụng thư viện bên ngoài (third party). Chúng có thể là FontAwesome, Bootstrap và các nội dung tương tự v.v..
- **main file:** import tất cả các partial SCSS của bạn

```

5 connect / sess / ? main.sess
  Nguyen Khanh Duy, 2 weeks a
  @use "./base";
  @use "./layout";
  @use "./components";
  @use "./pages";

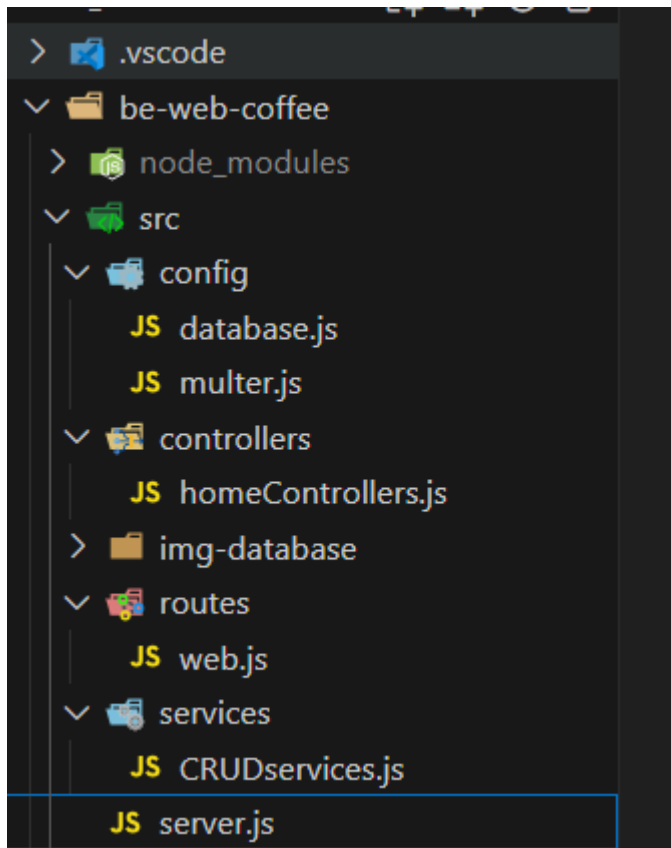
```

2) Mô hình MVC cho Backend



Mô hình MVC (MVC pattern) thường được dùng để **phát triển giao diện** người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho **máy tính** hoặc **điện thoại di động**, cũng như là các **ứng dụng web**.

Mô hình MVC gồm 3 loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:



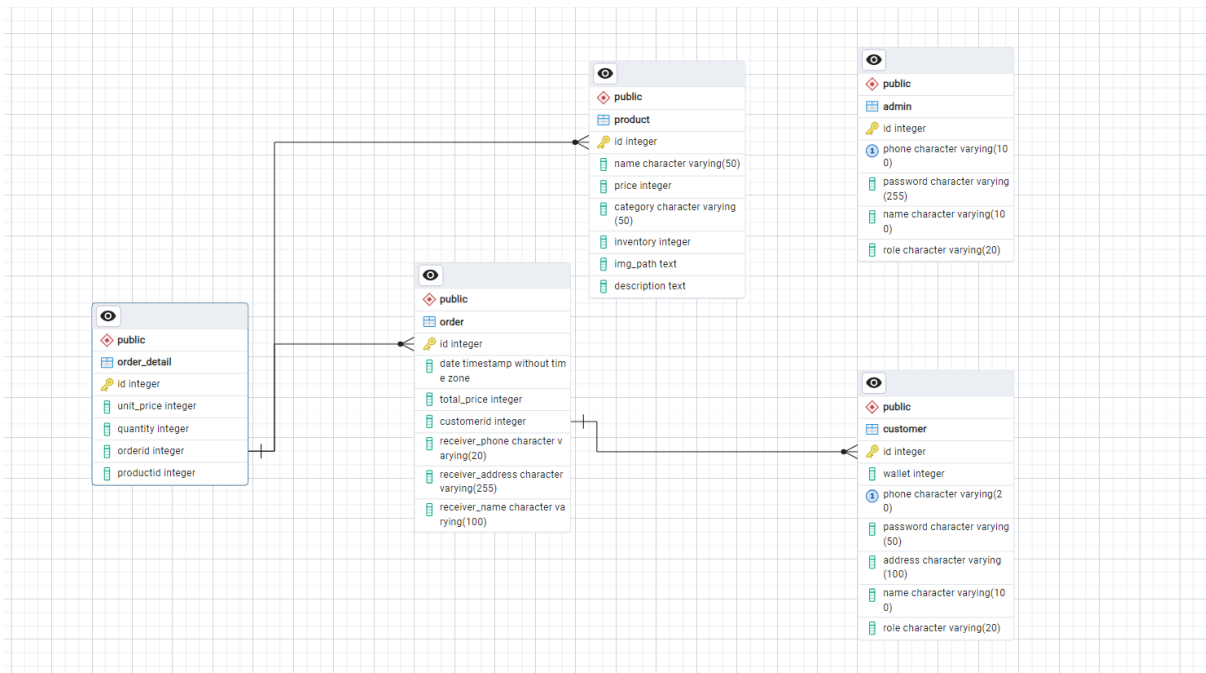
- **Model:** Là bộ phận có chức năng **lưu trữ toàn bộ dữ liệu** của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là **View** và **Controller**. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản. Chẳng hạn như biểu tượng hay là một nhân vật trong game.
 - **View:** Đây là phần giao diện (theme) dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
 - **Controller:** Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả **Model** lẫn **View**. Nó nhận input và thực hiện các update tương ứng.
- ➔ ở dự án này em dùng **controller** xử lý yêu cầu người dùng nhận input và thực hiện các update tương ứng . và file **service** dùng để viết các truy vấn .

III) Thiết kế dữ liệu

a) Tổng quan: Cơ sở dữ liệu bao gồm các collection sau:

- Bảng customer lưu trữ thông tin về khách hàng trong hệ thống.
- Bảng admin lưu trữ thông tin về các quản trị viên trong hệ thống.
- Bảng product lưu trữ thông tin về các sản phẩm được bán trong hệ thống.
- Bảng order lưu trữ thông tin về các đơn hàng được đặt bởi khách hàng.
- Bảng order_detail lưu trữ chi tiết về các sản phẩm trong từng đơn hàng.

b) Sơ lược sơ đồ dữ liệu



c) Chi tiết các thuộc tính bảng

Bảng customer:

- ☐ id serial PRIMARY KEY: Khóa chính, định danh duy nhất cho mỗi khách hàng.
- ☐ wallet int NOT NULL DEFAULT 1000000: Số tiền trong ví của khách hàng, mặc định là 1,000,000.
- ☐ phone varchar(20) NOT NULL UNIQUE: Số điện thoại của khách hàng, đảm bảo tính duy nhất.
- ☐ password varchar(50) NOT NULL: Mật khẩu của khách hàng.
- ☐ address varchar(100) NOT NULL: Địa chỉ của khách hàng.
- ☐ name varchar(100) NOT NULL: Tên của khách hàng.
- ☐ role varchar(20) DEFAULT 'Customer': Vai trò của người dùng, mặc định là 'Customer'.

Bảng admin :

- ☐ id serial PRIMARY KEY: Khóa chính, định danh duy nhất cho mỗi quản trị viên.
- ☐ phone varchar(100) NOT NULL UNIQUE: Số điện thoại của quản trị viên, đảm bảo tính duy nhất.
- ☐ password varchar(255) NOT NULL: Mật khẩu của quản trị viên.
- ☐ name varchar(100) NOT NULL: Tên của quản trị viên.
- ☐ role varchar(20) DEFAULT 'ADMIN': Vai trò của người dùng, mặc định là 'ADMIN'.

Bảng product :

- ☐ id serial PRIMARY KEY: Khóa chính, định danh duy nhất cho mỗi sản phẩm.
- ☐ name varchar(50) NOT NULL: Tên của sản phẩm.
- ☐ price int NOT NULL: Giá của sản phẩm.
- ☐ category varchar(50): Danh mục của sản phẩm.
- ☐ inventory integer NOT NULL: Số lượng sản phẩm còn trong kho.
- ☐ img_path text: Đường dẫn đến hình ảnh của sản phẩm.
- ☐ description text: Mô tả chi tiết về sản phẩm.

Bảng order :

- ☐ id serial PRIMARY KEY: Khóa chính, định danh duy nhất cho mỗi đơn hàng.
- ☐ date timestamp NOT NULL: Ngày và giờ tạo đơn hàng.
- ☐ total_price int NOT NULL: Tổng giá trị của đơn hàng.
- ☐ customerid int NOT NULL: ID của khách hàng đã đặt đơn hàng, liên kết với bảng customer.
- ☐ receiver_phone varchar(20) NOT NULL: Số điện thoại của người nhận.
- ☐ receiver_address varchar(255) NOT NULL: Địa chỉ của người nhận.
- ☐ receiver_name varchar(100) NOT NULL: Tên của người nhận.
- ☐ FOREIGN KEY (customerid) REFERENCES customer (id): Khóa ngoại, liên kết với ID của khách hàng trong bảng customer.

Bảng order_detail :

- ☐ id: Khóa chính, xác định duy nhất mỗi chi tiết đơn hàng.
- ☐ unit_price: Giá đơn vị của sản phẩm tại thời điểm đặt hàng.
- ☐ quantity: Số lượng sản phẩm trong đơn hàng.
- ☐ orderid: Mã đơn hàng, khóa ngoại tham chiếu đến order(id).
- ☐ productid: Mã sản phẩm, khóa ngoại tham chiếu đến product(id).
- ☐ **Bảng customer** có mối quan hệ một-nhiều với **bảng order**: Mỗi khách hàng có thể có nhiều đơn hàng.
- ☐ **Bảng order** có mối quan hệ một-nhiều với **bảng order_detail**: Mỗi đơn hàng có thể chứa nhiều chi tiết đơn hàng.
- ☐ **Bảng product** có mối quan hệ một-nhiều với **bảng order_detail**: Mỗi sản phẩm có thể xuất hiện trong nhiều chi tiết đơn hàng khác nhau.

IV) Thiết kế chức năng hệ thống

1) Đối với phân đăng nhập, đăng ký và đăng xuất

- Người sử dụng có thể đăng nhập hoặc đăng ký bằng ba phương thức sau: Sử dụng SĐT và password

- Sau khi đăng ký, người dùng được chuyển thẳng sang trang chính
- Khi một tài khoản đã đăng nhập, người dùng không cần phải đăng nhập lại mỗi khi truy cập web

2) Đối với phần người sử dụng:

Tất cả các chức năng của người sử dụng đều yêu cầu phải đăng nhập trước.

Xem và tra cứu thông tin sản phẩm

- Người dùng có thể tra cứu sản phẩm theo tên gọi và xem thông tin chi tiết của sản phẩm
- Khi người dùng chọn món thì sẽ lưu vào giỏ hàng bao gồm các thuộc tính cần thiết
- Khi người dùng đặt hàng và hoàn tất điền vào các thông tin của đơn hàng, đơn hàng sẽ được thêm vào cơ sở dữ liệu

Xem thông tin cá nhân

- Người dùng có thể xem và cập nhật một số thông tin cá nhân
- Có thể đổi mật khẩu của người dùng

Xem và chỉnh sửa địa chỉ đã lưu

- Người dùng có thể xem, chỉnh sửa hoặc xóa thông tin địa chỉ đã lưu trong ứng dụng

Xem lịch sử đơn hàng

- Người dùng có thể xem lịch sử cũng như đơn hàng và chi tiết của những đơn hàng đó(bao gồm tình trạng của đơn hàng)

3) Đối với admin

Xem danh sách sản phẩm

- Quản trị viên có quyền xem thông tin chi tiết và chỉnh sửa sản phẩm
- Bên cạnh đó có quyền thêm và xóa sản phẩm

V) Thiết kế và mô tả web

1) Màn hình đăng nhập

THE COFFEE HOUSE

Đăng Nhập

Chào mừng bạn quay trở lại The Coffee House!

1234567890

.....

Khôi phục mật khẩu

Đăng nhập

Đăng ký

THE COFFEE HOUSE

Đăng ký

Hãy đăng ký trở thành thành viên của The Coffee House

Số điện thoại

Mật khẩu

Xác nhận mật khẩu

Tên

Địa chỉ

☐ Đặt làm mật khẩu mặc định

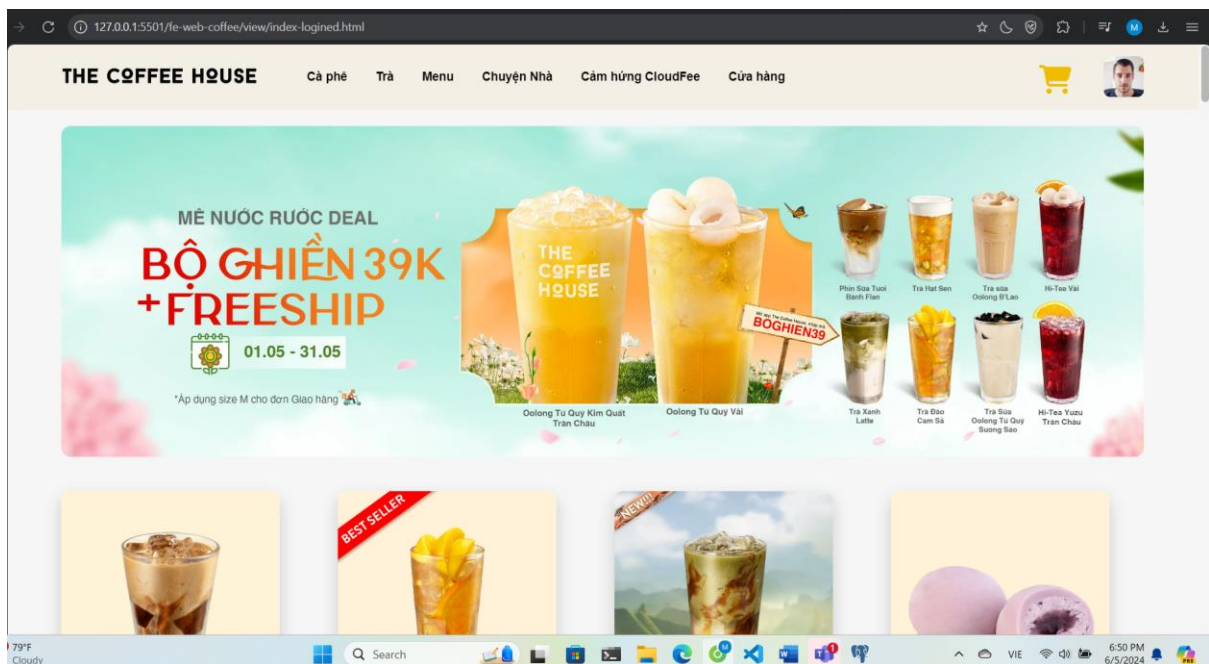
Khôi phục mật khẩu

Đăng ký

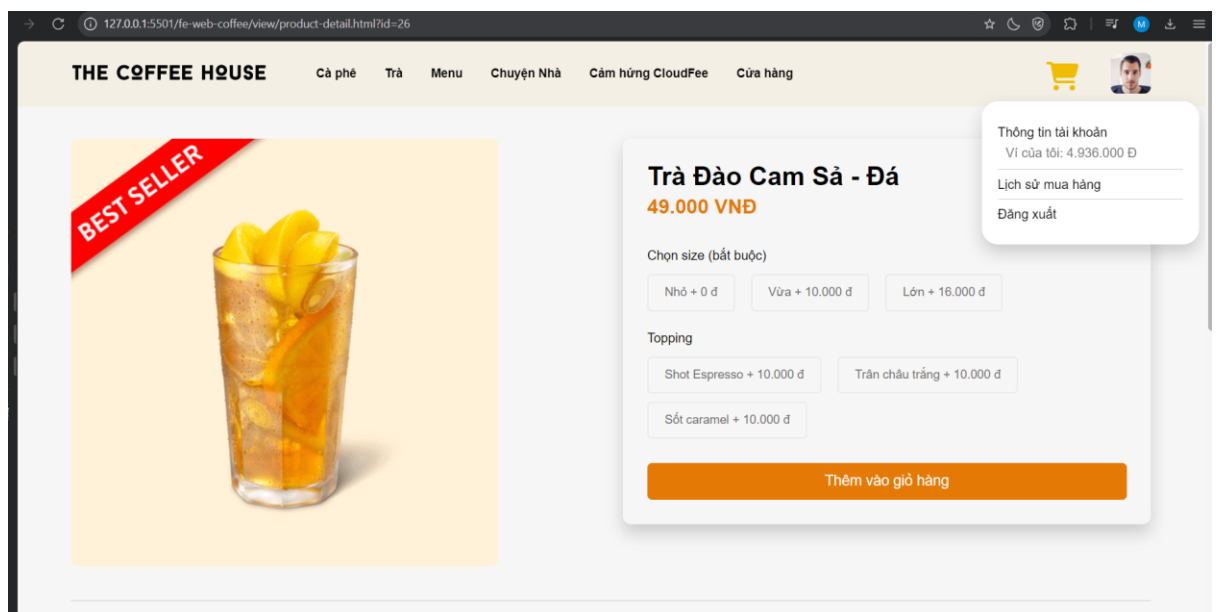
Bạn đã có tài khoản? [Đăng nhập](#)

Khi chạy ứng dụng sẽ vào màn hình đăng nhập và đăng ký .Khách hàng sẽ dùng SĐT password đăng nhập vào hệ thống. Đối với màn hình đăng ký người dùng đăng ký bằng Tên khách hàng, Số điện thoại, Địa chỉ,Tên Mật khẩu

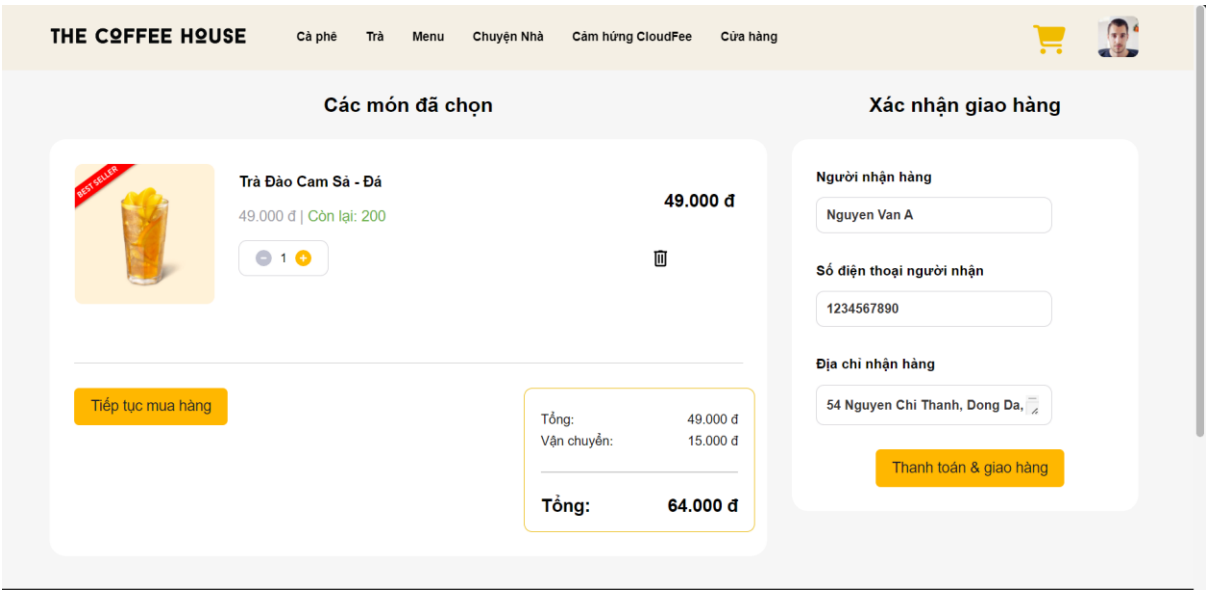
2) Màn hình trang chủ



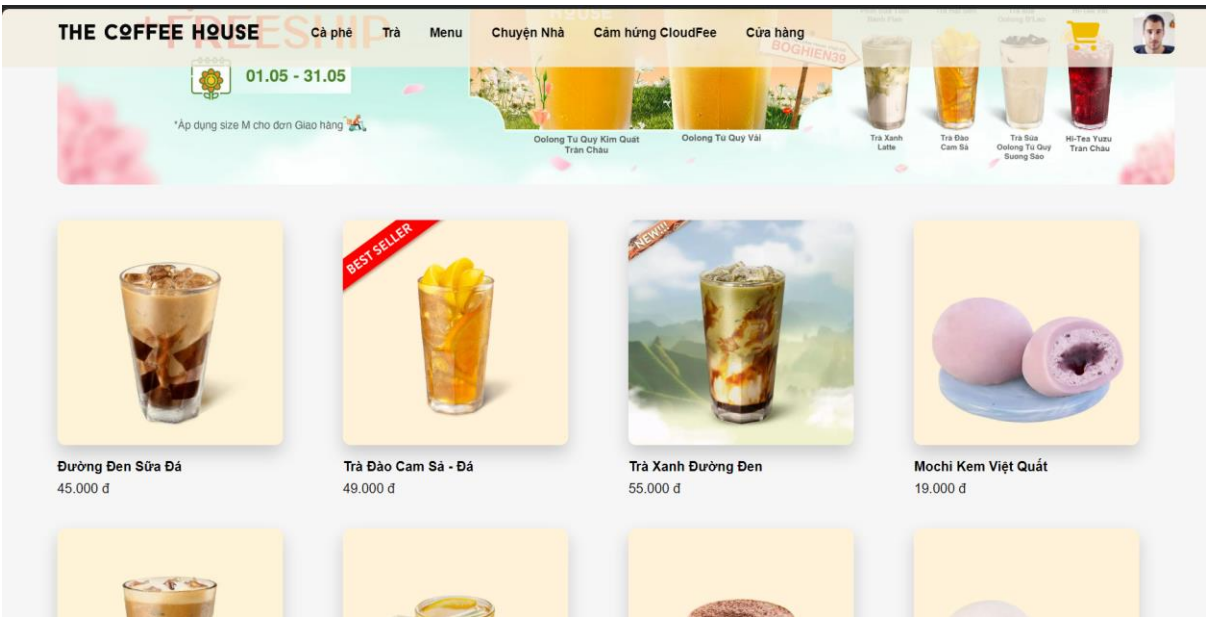
3) Giao diện sản phẩm khi đặt hàng



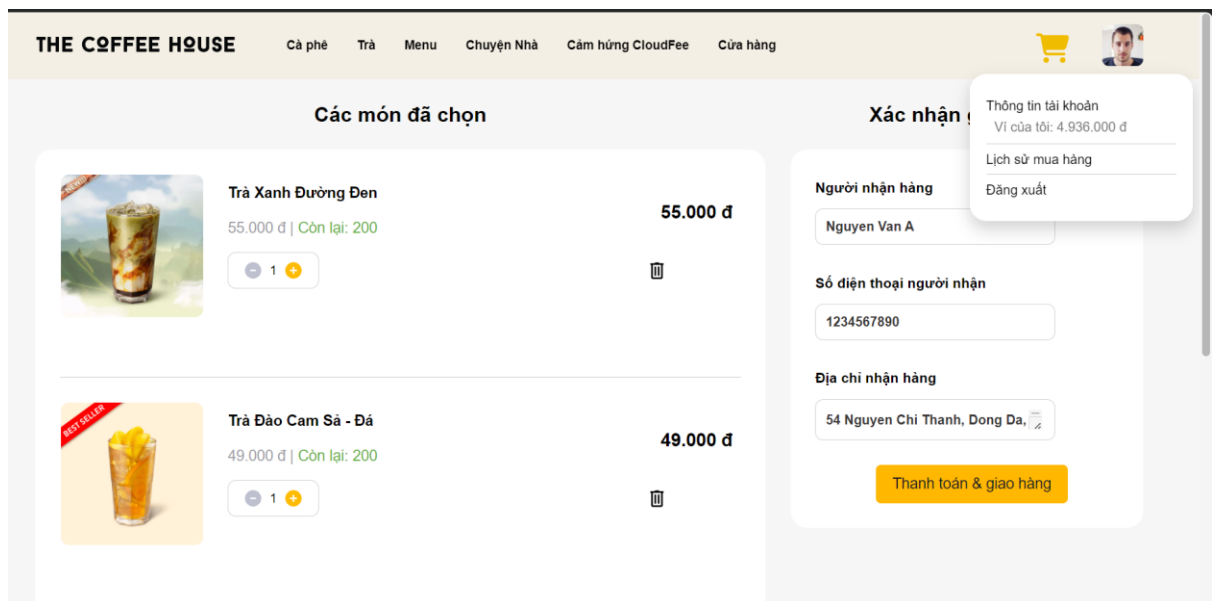
Khi ấn vào thêm giỏ hàng:



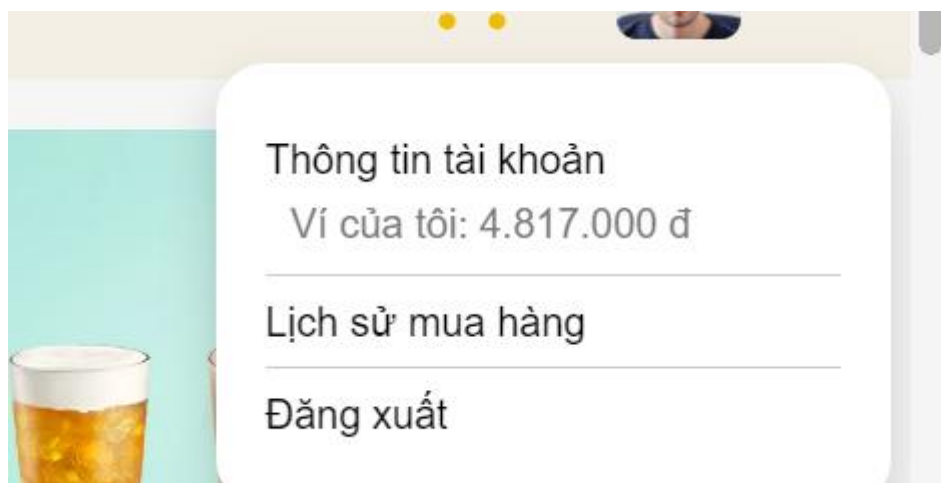
Khi ấn vào tiếp tục mua hàng:



Nó đã trở về trang chính ở đây chúng ta chọn thêm sản phẩm và đặt hàng



Sau khi ấn thanh toán:



Số tiền đã bị trừ