

ReactJS

Lab Assignment


Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.0
Effective Date	19/10/2021

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	29/Oct/2021	Create new assignment	Create	GiangPT12	
2					
3					

Contents

I. Mục tiêu	4
II. Tổng quát về ứng dụng	4
1. Tính năng:	4
2. Giao diện:	4
III. Chi tiết	5
1. Trang sản phẩm (Products O):	5
2. Trang thanh toán (Checkout page):	8
3. Mã màu và font chữ:	9
4. Lưu ý:	10

	CODE:	N/A
	TYPE:	N/A
	LOC:	N/A
	DURATION:	N/A

I. Mục tiêu

- Khởi tạo dự án CRA với Redux/MobX.
- Nắm vững các khái niệm cơ bản của Redux và vận dụng vào thực hành.
- Xử dụng Redux/MobX để quản lí global State bên trong một dự án React.
- Xử lý các thao tác bất đồng bộ cơ bản với Redux-Saga.

II. Tổng quát về ứng dụng

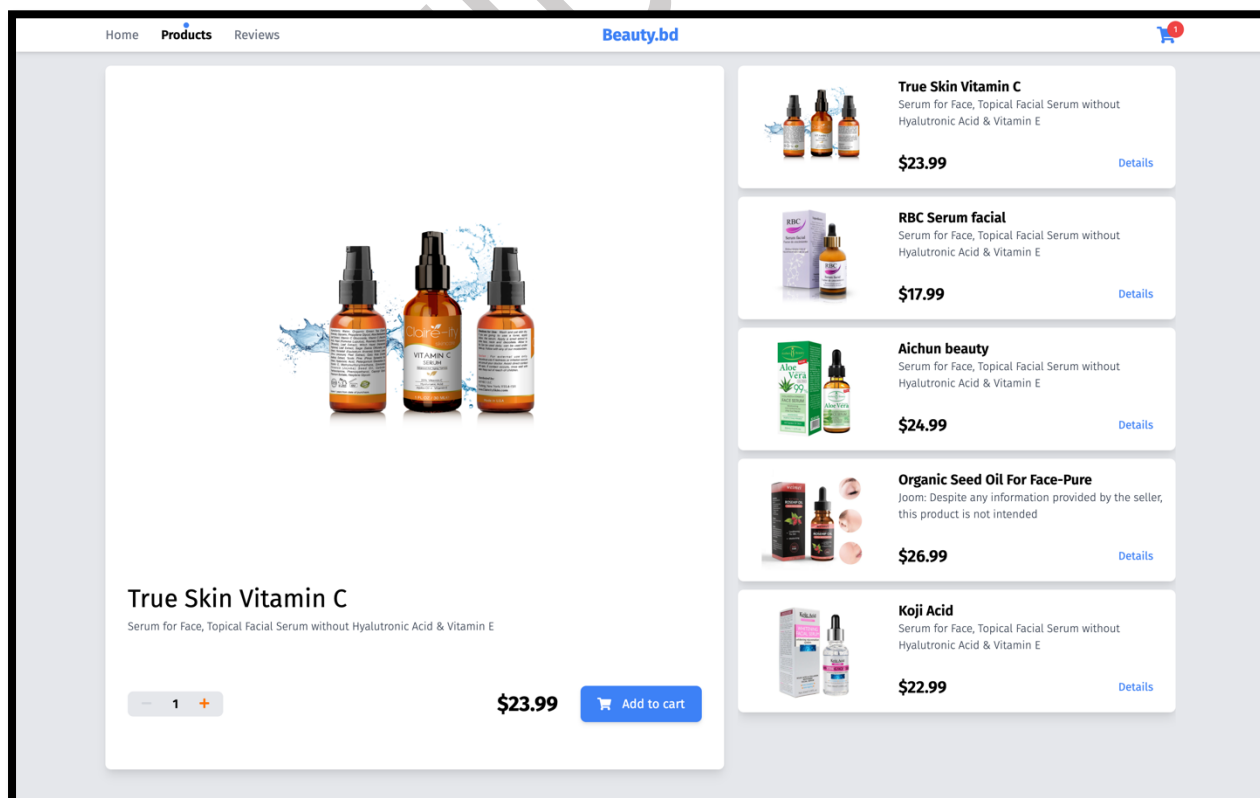
1. Tính năng:

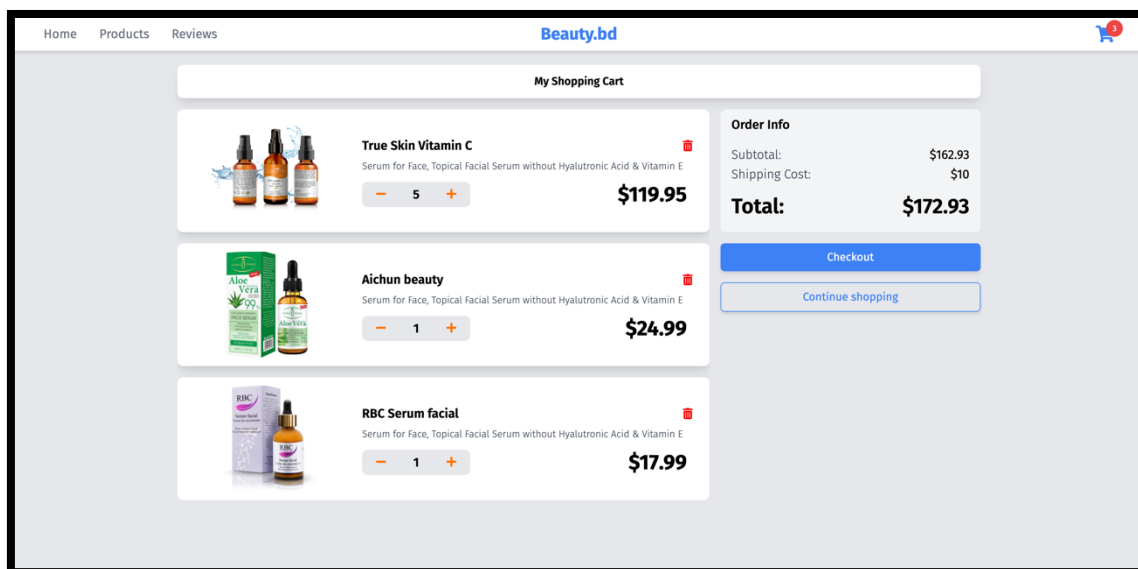
- Hiển thị danh sách sản phẩm và chi tiết sản phẩm.
- Có thể thêm sản phẩm vào giỏ hàng, thay đổi số lượng sản phẩm.
- Có tính năng thanh toán đơn giản.

2. Giao diện:

- Có thể tham khảo giao diện của ứng dụng tại link sau:

<https://fa-react-shopping-carts.vercel.app>

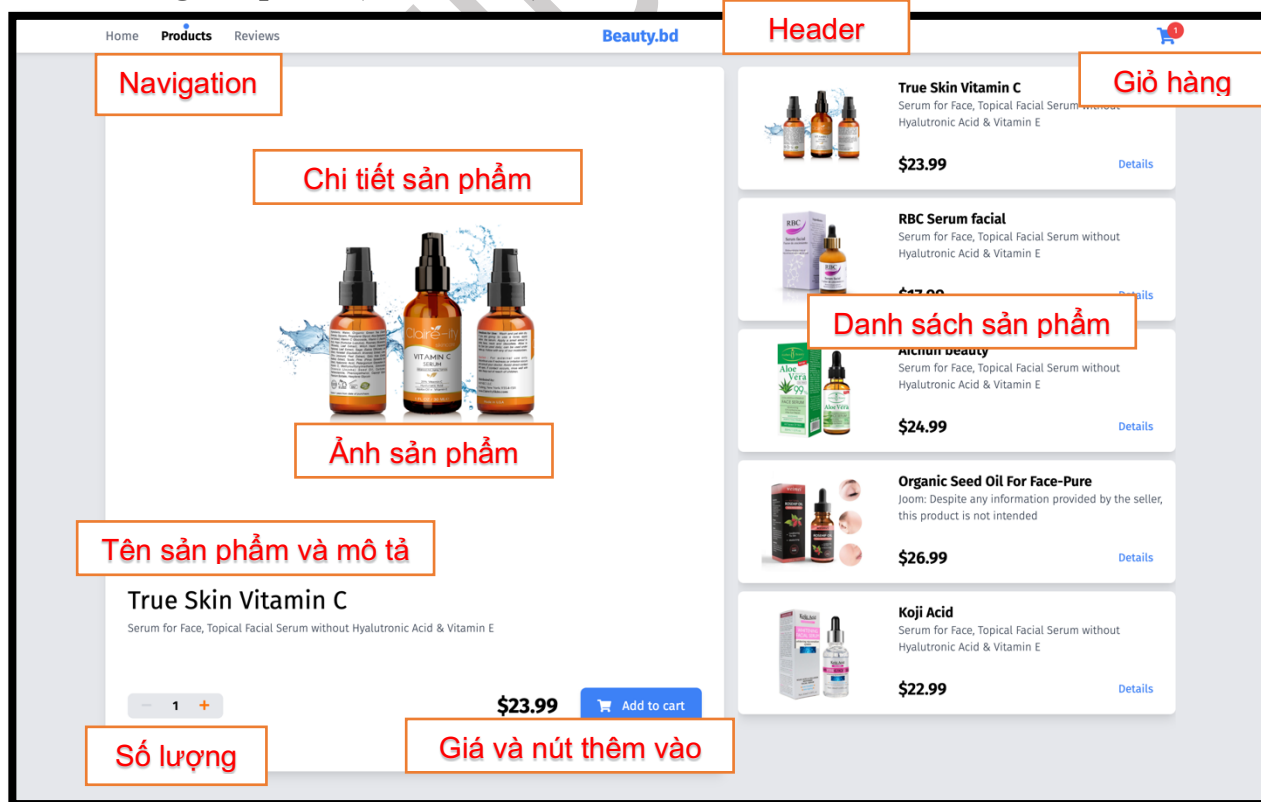




- Ứng dụng có 2 màn hình:
 - Trang sản phẩm (Products page).
 - Trang thanh toán (Checkout page).

III. Chi tiết

1. Trang sản phẩm (Products O):



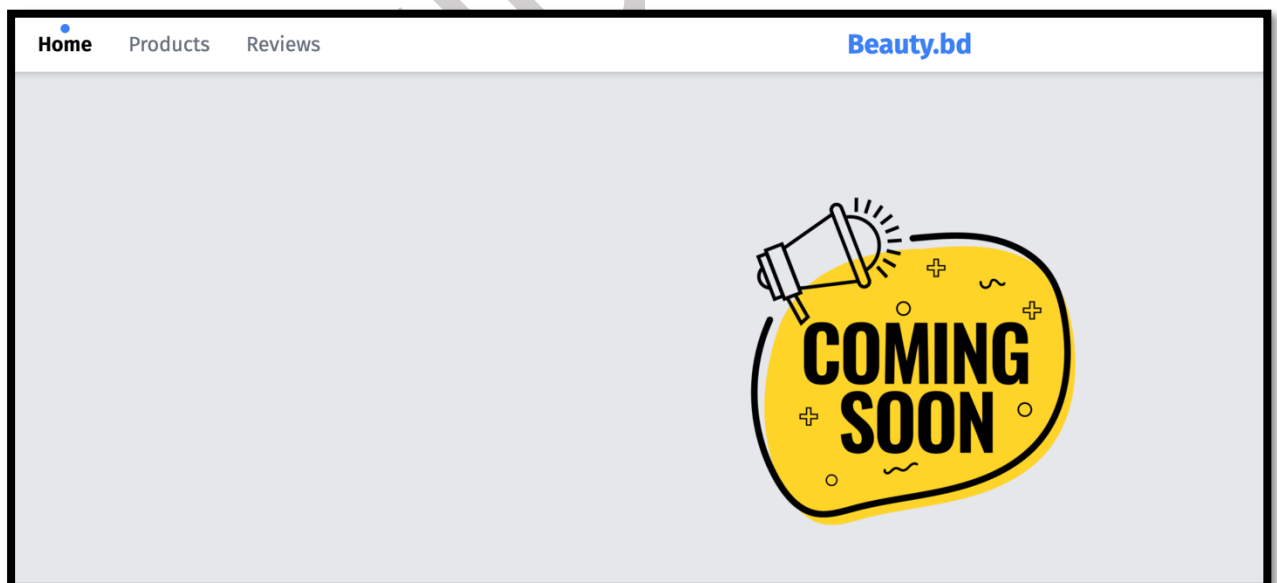
Yêu cầu:

a) Giao diện:

- Header bao gồm 2 phần:
 - Navigation: Gồm Home, Products và Review.
 - Cart: Giỏ hàng.
- Danh sách sản phẩm:
 - Hiển thị danh sách các sản phẩm bao gồm các thông tin sau: Ảnh, tên, mô tả và giá tiền sản phẩm.
- Chi tiết sản phẩm: Hiển thị thông tin chi tiết của một sản phẩm.
 - Ảnh sản phẩm.
 - Tên sản phẩm và mô tả sản phẩm.
 - Số lượng sản phẩm khách hàng muốn đặt.
 - Giá sản phẩm và nút thêm giỏ hàng.

b) Chức năng:

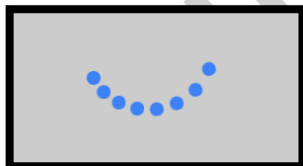
- Header:
 - Navigation: Bao gồm 3 links Home - (/home) , Products - (/products) , Reviews (/reviews).
 - Ở đây chúng ta chỉ làm việc với trang Products và khi click vào Home và Reviews thì sẽ hiển thị như sau:



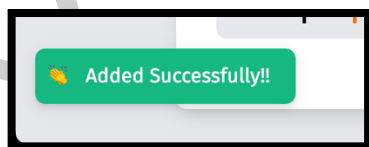
- Cart: Khi click vào giỏ hàng sẽ chuyển đến trang thanh toán - (/checkout). Khi thêm vào một sản phẩm bất kì vào giỏ hàng thì sẽ hiển thị số lượng sản phẩm trong giỏ hàng.



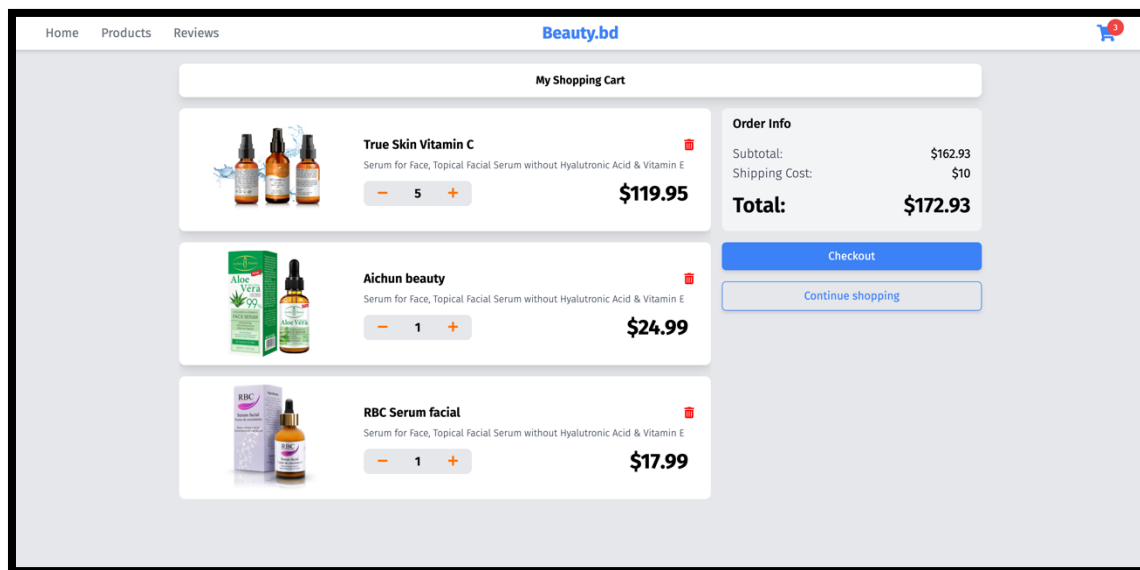
- Danh sách sản phẩm:
 - Hiển thị danh sách tất cả sản phẩm. Khi click vào tên sản phẩm hoặc nút Details, thì thông tin sản phẩm ở phần chi tiết sản phẩm bên trái sẽ được cập nhật theo tương ứng.
- Chi tiết sản phẩm:
 - Ảnh, tên và mô tả sản phẩm sẽ được cập nhật tương ứng khi người dùng chọn bất kì một sản phẩm từ danh sách sản phẩm.
 - Hiển thị số lượng sản phẩm mặc định là 1. Người dùng có thể tăng hoặc giảm số lượng sản phẩm bằng cách lick vào nút (+) hoặc (-). Khi số lượng sản phẩm đạt tối thiểu hoặc tối đa, nút (+) hoặc (-) sẽ bị disabled và không thể click được nữa (tối thiểu 1 và tối đa 99).
 - Khi tăng hoặc giảm số lượng sản phẩm thì giá tiền sẽ được thay đổi theo tương ứng.
 - Khi click vào nút thêm giỏ hàng sẽ thêm sản phẩm đó vào giỏ hàng.
- Bonus:
 - Hiển thị Loading khi lấy sản phẩm từ server.



- Nếu sản phẩm đã có trong giỏ hàng, sẽ chỉ cập nhật số lượng của sản phẩm đó trong giỏ hàng chứ không thêm một sản phẩm mới vào giỏ hàng.
- Khi thêm sản phẩm thành công hiển thị Toast thông báo cho người dùng:



2. Trang thanh toán (Checkout page):



Yêu cầu:

a) Giao diện:

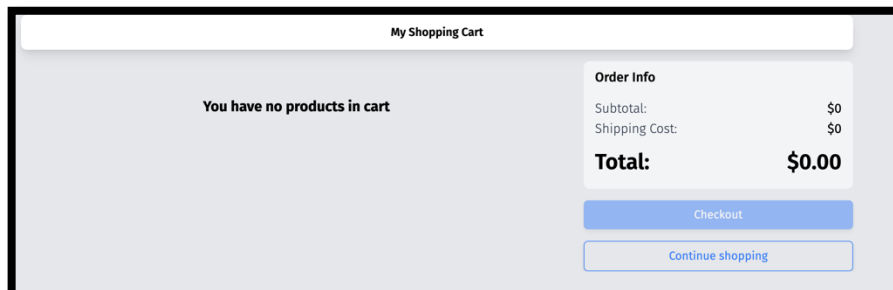
- Thông tin sản phẩm:
 - Hiển thị các sản phẩm đã được thêm vào giỏ hàng
 - Chi tiết từng sản phẩm: Bao gồm ảnh, tên và mô tả sản phẩm.
 - Hiển thị số lượng sản phẩm đó đã được thêm vào giỏ và tổng giá tiền tương ứng.
 - Delete Button để xóa sản phẩm.
- Giá tiền:
 - Subtotal: Hiển thị tổng giá tiền của toàn bộ sản phẩm trong giỏ hàng.
 - Shipping Cost: Phí giao hàng (mặc định là 10\$).
 - Total: Subtotal + Shipping Cost.
- Actions:
 - Checkout Button: Nút thanh toán.
 - Continue Shopping: Quay trở về trang sản phẩm.

b) Chức năng:

- Thông tin sản phẩm:
 - Mỗi sản phẩm sẽ tương ứng với một hàng.
 - Khi tăng, giảm hoặc xóa sản phẩm thì giá tiền ở từng sản phẩm và tổng giá tiền sẽ thay đổi tương ứng.
 - Khi không có sản phẩm nào trong giỏ hàng, hiển thị: You have no products in cart.

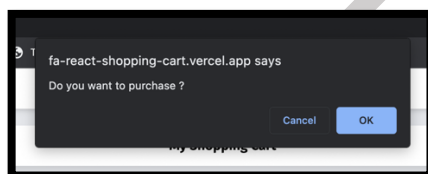
- Actions:

- Khi click Checkout Button, gửi lên server thông tin giỏ hàng. Sau khi server trả về kết quả thành công thì di chuyển người dùng đến trang Products page đồng thời xoá toàn bộ sản phẩm trong giỏ hàng.
- Khi không có sản phẩm nào trong giỏ hàng, Checkout Button sẽ bị disabled và không thể click được.

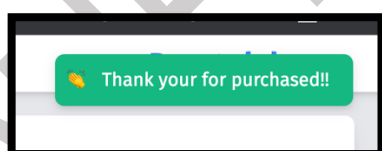


- **Bonus:**

- Giảm số lượng sản phẩm về 0 thì sản phẩm sẽ tự động xoá khỏi giỏ hàng.
- Trước khi checkout hiển thị pop-up xác nhận và chỉ thanh toán khi người dùng chọn Yes.



- Trong quá trình checkout, hiển thị Loading cho người dùng.
- Sau khi thanh toán thành công thì hiển thị Toast thông báo:



3. Mã màu và font chữ:

- App background color: #E5E7EB
- Logo/Cart/Buttons Blue: #3B82F6
- Plus/minus quantity button: #FF7300
- Red cart quantity / Delete button: #EF4444
- Green: #10B981
- Font link nhúng vào thẻ head: Google font Fira-Sans
<https://fonts.googleapis.com/css2?family=Fira+Sans:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap>

4. Lưu ý:

- Yêu cầu học viên sử dụng Redux hoặc MobX. Khuyến khích sử dụng thêm Redux-saga để thao tác bất đồng bộ.
- Học viên được cung cấp source backend. Yêu cầu thao tác với API để lấy sản phẩm và thanh toán. Không sử dụng data phía client.
- Để chạy được source backend học viên làm theo các bước như sau:
 - Tại thư mục chứa file package.json chạy lệnh: npm install để tải về các packages cần thiết.
 - Sau khi quá trình tải hoàn tất, chạy lệnh: npm run dev để chạy server.
 - Khi màn hình log hiển thị như sau là chạy server thành công:

```
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [InstanceLoader] ConfigHostModule dependencies initialized +0ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [InstanceLoader] ProductsModule dependencies initialized +1ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [InstanceLoader] CheckoutModule dependencies initialized +0ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [InstanceLoader] AppModule dependencies initialized +0ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [InstanceLoader] ConfigModule dependencies initialized +1ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [RoutesResolver] ApplicationController {/}: +164ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [RoutesResolver] ProductsController {/api/products}: +0ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [RouterExplorer] Mapped {/api/products, GET} route +2ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [RoutesResolver] CheckoutController {/api/checkout}: +1ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [RouterExplorer] Mapped {/api/checkout, POST} route +0ms
[Nest] 39268 - 10/17/2021, 2:52:50 PM LOG [NestApplication] Nest application successfully started +2ms
Server is running on PORT: 4000
```

- Server sẽ chạy với url: http://localhost:400

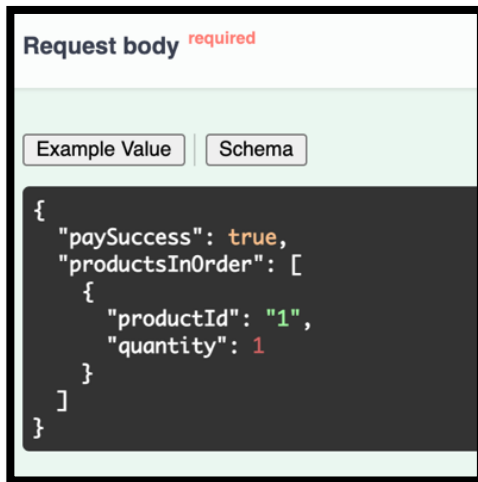
GET /api/products - Lấy về toàn bộ sản phẩm

Response (success):

Status: 200

data:

```
Response body
[
  {
    "productId": "1",
    "productName": "True Skin Vitamin C",
    "description": "Serum for Face, Topical Facial Serum without Hyaluronic Acid & Vitamin E",
    "imageUrl": "https://res.cloudinary.com/dyotzt92h/image/upload/v1632673370/Fsoft-academy/favpng_vitamin-c-vitamin-e-serum-skin_sligc2.p",
    "price": 23.99
  },
  {
    "productId": "2",
    "productName": "RBC Serum facial",
    "description": "Serum for Face, Topical Facial Serum without Hyaluronic Acid & Vitamin E",
    "imageUrl": "https://res.cloudinary.com/dyotzt92h/image/upload/v1632909666/Fsoft-academy/pngwing.com_ixjcnb.png",
    "price": 17.99
  },
  {
    "productId": "3",
    "productName": "Aichun beauty",
    "description": "Serum for Face, Topical Facial Serum without Hyaluronic Acid & Vitamin E",
    "imageUrl": "https://res.cloudinary.com/dyotzt92h/image/upload/v1632909875/Fsoft-academy/c5e83bbc7b60fc7d61da7bc07ec129dd31fc0d9b_origin",
    "price": 24.99
  }
]
```

POST /api/products - Thanh toán

Request body - required. Ví dụ:

- paySuccess: Xác nhận thanh toán thành công.
- productsInOrder:
 - Danh sách sản phẩm đã thanh toán.
 - Mỗi sản phẩm bao gồm productId và số lượng.

Response (success):

Status: 201

data:



- Để xem chi tiết hơn các APIs, học viên có thể truy cập công cụ swagger UI với đường dẫn <http://localhost:4000/api/document> sau khi chạy server.
- Học viên được phép sử dụng bất kì thư viện UI nào để làm bài.
- Để làm Toast, học viên có thể tham khảo và lựa chọn một thư viện bất kì để làm Toastify trên NPM. Không yêu cầu sử dụng một thư viện cụ thể.

-----**(END && GOODLUCK)**-----