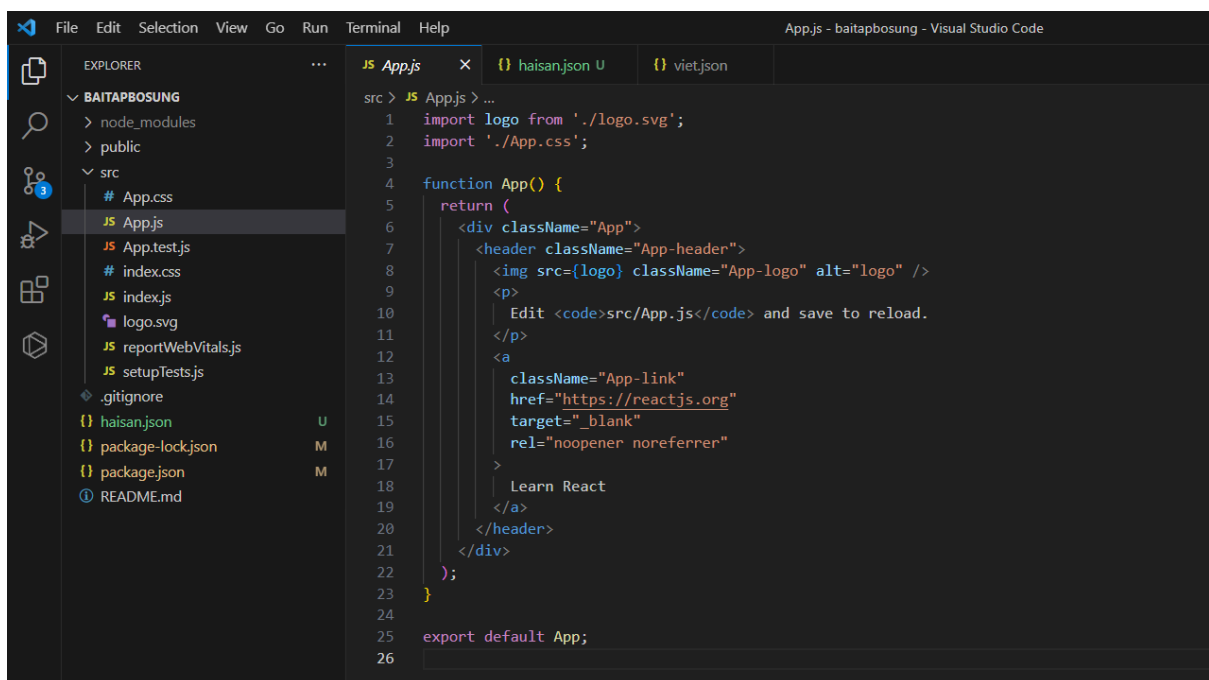


Báo cáo BTL mẫu - TLU

Đề bài: Viết ứng dụng và mô tả ReactJS CRUD Rest API (JSON Fake server, sử dụng thư viện Axios)

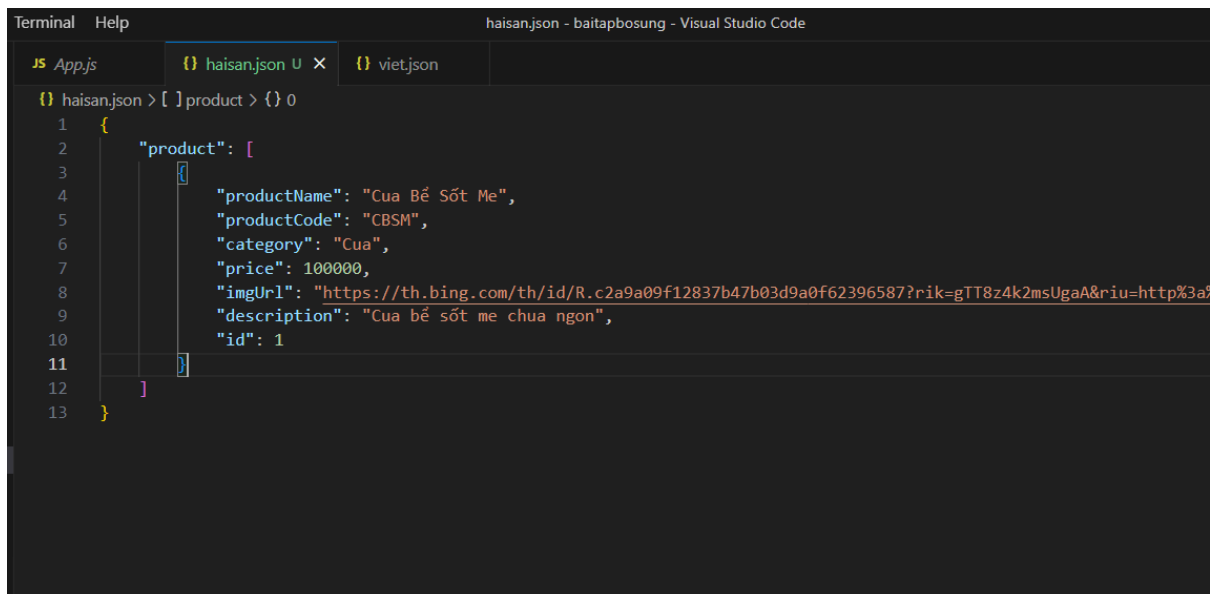
- READ:
- Bước 1: Tạo 1 ứng dụng React bằng câu lệnh **npx create-react-app baitapbosung** bằng CMD trong thư mục mới được tạo ra
- Bước 2: Mở VS Code lên để làm việc với ứng dụng



The screenshot shows the Visual Studio Code interface. On the left, the Explorer panel displays the file structure of the 'BAITAPBOSUNG' project, including 'node_modules', 'public', and 'src' folders. The 'App.js' file is selected in the 'src' folder. The main editor area shows the content of 'App.js', which includes imports for 'logo.svg' and 'App.css', a function 'App()' that returns a JSX element, and an 'export default App;' statement. The JSX element contains a header with a logo and a link to 'https://reactjs.org'.

```
src > JS App.js > ...
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <img src={logo} className="App-logo" alt="logo" />
9          <p>
10             Edit <code>src/App.js</code> and save to reload.
11          </p>
12          <a
13             className="App-link"
14             href="https://reactjs.org"
15             target="_blank"
16             rel="noopener noreferrer"
17           >
18             Learn React
19          </a>
20        </header>
21      </div>
22    );
23  }
24
25  export default App;
26
```

- Bước 3: Tạo 1 file json chứa các list product. Tại đây, ta giả sử tạo ra cứng 1 sản phẩm nhất định để phục vụ việc test CRUD

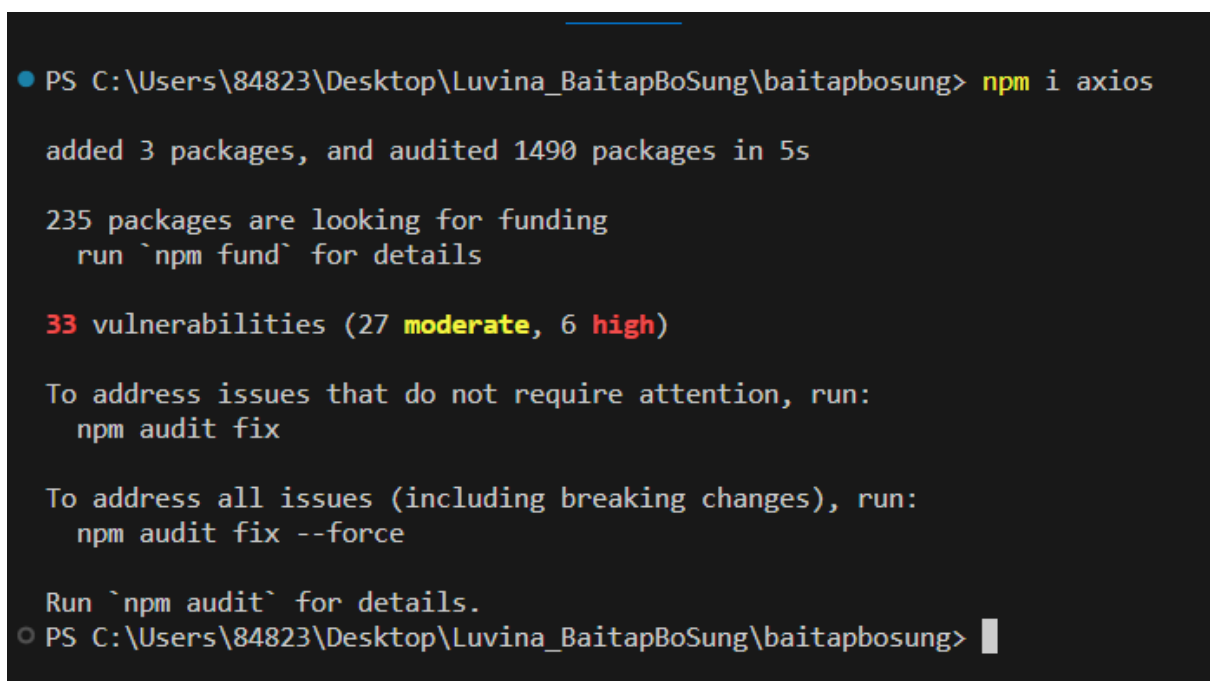


```
Terminal  Help  haisan.json - baitapbosung - Visual Studio Code

JS App.js  {} haisan.json U x  {} viet.json

{} haisan.json > [ ] product > {} 0
1  {
2    "product": [
3      {
4        "productName": "Cua Bể Sốt Me",
5        "productCode": "CBSM",
6        "category": "Cua",
7        "price": 100000,
8        "imgUrl": "https://th.bing.com/th/id/R.c2a9a09f12837b47b03d9a0f62396587?rik=TT8z4k2msUgaA&riu=http%3a%2f%2fwww.baitapbosung.com%2fimages%2fuploads%2fproducts%2f1%2f1.jpg",
9        "description": "Cua bể sốt me chua ngon",
10       "id": 1
11     }
12   ]
13 }
```

- Bước 4: Import package axios bằng câu lệnh **npm i axios**



```
PS C:\Users\84823\Desktop\Luvina_BaitapBoSung\baitapbosung> npm i axios

added 3 packages, and audited 1490 packages in 5s

235 packages are looking for funding
  run `npm fund` for details

33 vulnerabilities (27 moderate, 6 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\Users\84823\Desktop\Luvina_BaitapBoSung\baitapbosung>
```

- Bước 5: Trong file App.js, xây dựng template HTML và CSS để tạo ra 1 bảng quản lí hải sản

Manager Seafood

Name	Code	Category	Price	Image	Description	Actions
------	------	----------	-------	-------	-------------	---------

- Bước 6: Mở terminal, nhập lệnh `npx json-server --watch haisan.json --port 4200` để khởi chạy 1 DB giả

```
PS C:\Users\84823\Desktop\Luvina_BaitapBoSung\baitapbosung> npx json-server --watch haisan.json --port 4200

\{^_^}/ hi!

Loading haisan.json
Done

Resources
http://localhost:4200/product

Home
http://localhost:4200

Type s + enter at any time to create a snapshot of the database
Watching...
```

- Bước 7: Trong file App.js, khởi tạo 1 biến seafood là nơi sẽ chứa list hải sản khi call API lấy toàn bộ dữ liệu dưới DB. Tại đây, ta sử dụng useEffect để mục đích tránh các side effect xảy ra (re-render lại liên tục) với đối số của useEffect là rỗng để render 1 lần sau khi component được tạo. Trong hàm này, ta sẽ sử dụng thư viện axios để gọi API với phương thức GET cho mục đích lấy list dữ liệu dưới DB lên. Nếu thành công sẽ chạy vào hàm .then() và set biến seafood khởi tạo từ đầu là 1 list hải sản có sẵn dưới DB đã lấy lên. Còn nếu không thành công sẽ in ra lỗi dưới tab console trong browser

```
> JS App.js > App
1 import './App.css'
2 import { useState, useEffect } from 'react';
3 import axios from 'axios'
4
5 function App() {
6
7   const [seafood, setSeafood] = useState([])
8
9   useEffect(() => {
10     axios.get("http://localhost:4200/product")
11       .then(response => {
12         setSeafood(response.data)
13       })
14       .catch(error => { console.log(error) })
15   }, [])
16 }
```


- Bước 8: Sau khi call thành công ta sẽ binding dữ liệu của biến seafood ra màn hình

```

7   return (
8     <>
9     <h1 style={{ textAlign: "center" }} className="mt-5">Manager Seafood</h1>
10    <div style={{ padding: "0 100px" }} className="text-center">
11      <button className="mt-3 btn btn-primary" >Add Seafood</button>
12      <table id="customers" className="mt-4">
13        <tr>
14          <th>ID</th>
15          <th> Name</th>
16          <th> Code</th>
17          <th> Category</th>
18          <th>Price</th>
19          <th>Image</th>
20          <th>Description</th>
21          <th>Actions</th>
22        </tr>
23        {seafood.map((item, index) => {
24          return (
25            <>
26              <tr key={item.id}>
27                <td>{item.id}</td>
28                <td>{item.productName}</td>
29                <td>{item.productCode}</td>
30                <td>{item.category}</td>
31                <td>{item.price.toLocaleString()}đ</td>
32                <td>
33                  <img src={item.imageUrl} height="200px" width="300px" />
34                </td>
35                <td>{item.description}</td>
36                <td>
37                  <button className="btn btn-warning me-2">Edit</button>
38                  <button className="btn btn-danger">Delete</button>
39                </td>
40              </tr>
41            </>
42          )
43        })}
44      </table>

```

- Kết quả màn hình:

Manager Seafood							
Add Seafood							
ID	Name	Code	Category	Price	Image	Description	Actions
1	Cua Bể Sốt Me	CBSM	Cua	100.000đ		Cua bể sốt me chua ngon	<div>Edit</div> <div>Delete</div>

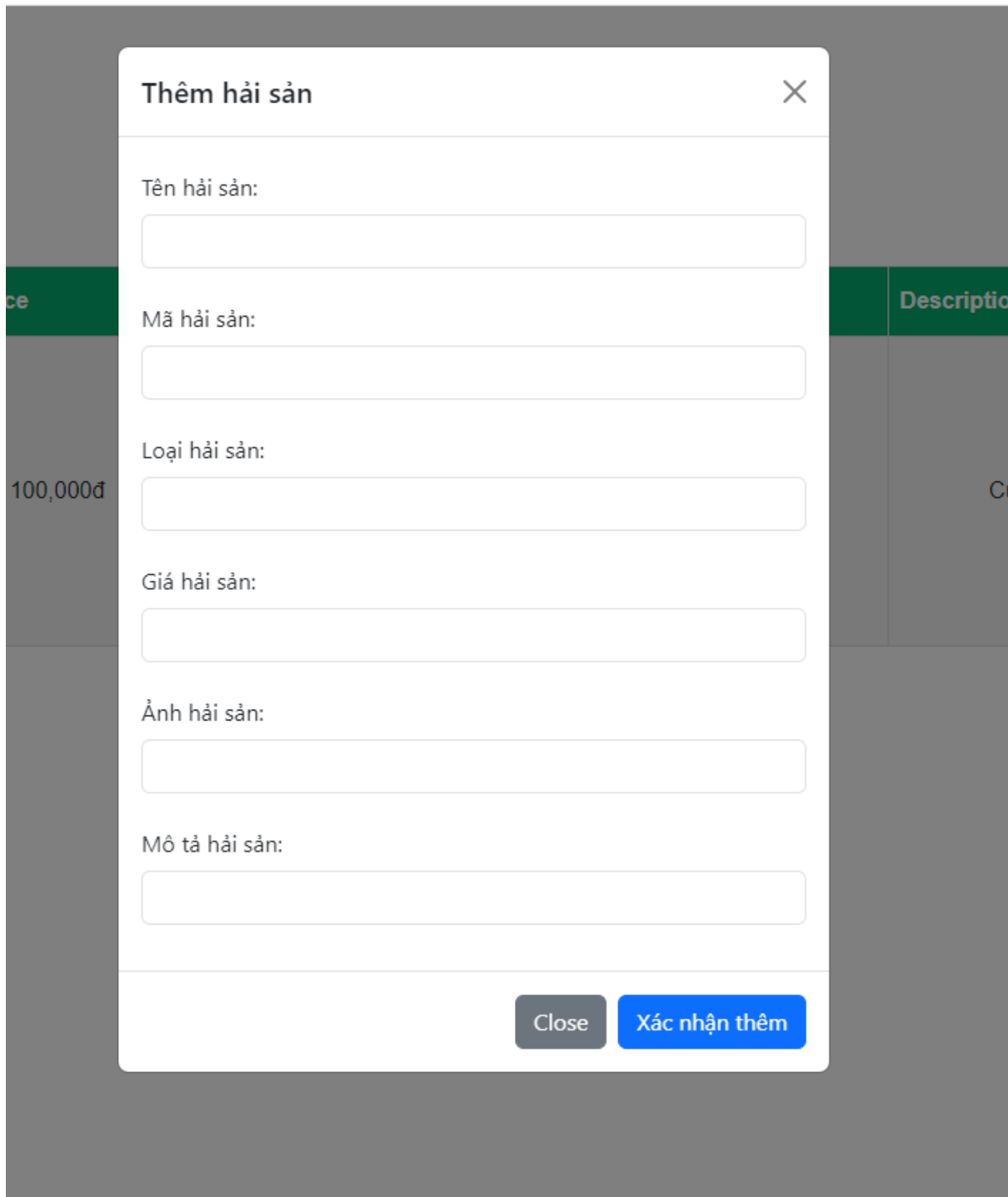
- ADD:
- Bước 9: Bây giờ ta sẽ viết phần Add, đầu tiên ta sẽ thêm 1 format Modal từ Bootstrap 5 để tiện thêm các trường cho thông tin hải sản mới mà ta thêm vào

```

56
57 { /* Modal */}
58 <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
59   <div class="modal-dialog">
60     <div class="modal-content">
61       <div class="modal-header">
62         <h1 class="modal-title fs-5" id="exampleModalLabel">Thêm hải sản</h1>
63         <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
64       </div>
65       <div class="modal-body row">
66         <form>
67           <div className="mb-3">
68             <label for="product-name" className="col-form-label">Tên hải sản:</label>
69             <input type="text" className="form-control" />
70           </div>
71           <div className="mb-3">
72             <label for="product-name" className="col-form-label">Mã hải sản:</label>
73             <input type="text" className="form-control" />
74           </div>
75           <div className="mb-3">
76             <label for="product-name" className="col-form-label">Loại hải sản:</label>
77             <input type="text" className="form-control" />
78           </div>
79           <div className="mb-3">
80             <label for="product-name" className="col-form-label">Giá hải sản:</label>
81             <input type="number" min={1000000} className="form-control" />
82           </div>
83           <div className="mb-3">
84             <label for="product-name" className="col-form-label">Ảnh hải sản:</label>
85             <input type="text" className="form-control" />
86           </div>
87           <div className="mb-3">
88             <label for="product-name" className="col-form-label">Mô tả hải sản:</label>
89             <input type="text" className="form-control" />
90           </div>
91           <div className="mb-3">
92             <label for="product-name" className="col-form-label">Loại hải sản:</label>
93             <input type="text" className="form-control" />
94           </div>

```

- Kết quả màn hình:



The image shows a modal window titled "Thêm hải sản" (Add Seafood) with a close button (X) in the top right corner. The form contains six input fields, each with a label above it: "Tên hải sản:" (Seafood Name), "Mã hải sản:" (Seafood Code), "Loại hải sản:" (Seafood Type), "Giá hải sản:" (Seafood Price), "Ảnh hải sản:" (Seafood Image), and "Mô tả hải sản:" (Seafood Description). At the bottom right of the modal, there are two buttons: "Close" and "Xác nhận thêm" (Confirm Add).

Thêm hải sản	
Tên hải sản:	<input type="text"/>
Mã hải sản:	<input type="text"/>
Loại hải sản:	<input type="text"/>
Giá hải sản:	<input type="text"/>
Ảnh hải sản:	<input type="text"/>
Mô tả hải sản:	<input type="text"/>
<div>Close Xác nhận thêm</div>	

- Bước 10: Ta sẽ tạo ra các biến ứng với các trường có trong Modal và trong mỗi element Modal ta sẽ set thẳng từng giá trị ứng với từng giá trị mà người dùng nhập vào ở ô input

```
function App() {
  const [seafood, setSeafood] = useState([])
  const [name, setName] = useState("")
  const [code, setCode] = useState("")
  const [category, setCategory] = useState("")
  const [price, setPrice] = useState()
  const [img, setImg] = useState("")
  const [description, setDescription] = useState("")
  useEffect(() => {
```

```
<form>
  <div className="mb-3">
    <label for="product-name" className="col-form-label">Tên hải sản:</label>
    <input type="text" className="form-control"
      onChange={(e) => setName(e.target.value)}
    />
  </div>
  <div className="mb-3">
    <label for="product-name" className="col-form-label">Mã hải sản:</label>
    <input type="text" className="form-control"
      onChange={(e) => setCode(e.target.value)}
    />
  </div>
  <div className="mb-3">
    <label for="product-name" className="col-form-label">Loại hải sản:</label>
    <input type="text" className="form-control"
      onChange={(e) => setCategory(e.target.value)}
    />
  </div>
  <div className="mb-3">
    <label for="product-name" className="col-form-label">Giá hải sản:</label>
    <input type="number" min={1000000} className="form-control"
      onChange={(e) => setPrice(e.target.value)}
    />
  </div>
  <div className="mb-3">
    <label for="product-name" className="col-form-label">Ảnh hải sản:</label>
    <input type="text" className="form-control"
      onChange={(e) => setImg(e.target.value)}
    />
  </div>
  <div className="mb-3">
    <label for="product-name" className="col-form-label">setDescription(value: React.SetSt
    <input type="text" className="form-control"
      onChange={(e) => setDescription(e.target.value)}
    />
  </div>
</form>
```


- Bước 11: Ở nút xác nhận thêm, ta thêm 1 hàm `handleAdd`. Trong hàm này ta sẽ dùng `axios` với phương thức `POST` để gọi API lưu hải sản mới theo format định sẵn ngay từ đầu (giá trị ta fix cứng từ đầu để test). Tương tự như `GET`, trong hàm `.then`, ta sẽ cho 1 alert thông báo nếu add thành công và trong `.catch` ta sẽ in lỗi ra dưới console. Ở đây, ta sẽ bổ sung 1 biến boolean `isChanged` để khi ta add thì đồng thời hàm `useEffect` sẽ được render lại do biến `isChanged` thay đổi nhằm mục đích render lại 1 list hải sản mới và bind dữ liệu ra màn hình (thuận tiện không cần phải reload trang lại, hiển thị tức thì)

```
8   const [code, setCode] = useState("")
9   const [category, setCategory] = useState("")
10  const [price, setPrice] = useState()
11  const [img, setImg] = useState("")
12  const [description, setDescription] = useState("")
13  const [isChanged, setIsChanged] = useState(false)
14
15  useEffect(() => {
16    axios.get("http://localhost:4200/product")
17      .then(response => {
18        setSeafood(response.data)
19      })
20      .catch(error => { console.log(error) })
21  }, [isChanged])
22
23
24  const handleAdd = () => {
25    setIsChanged(!isChanged)
26    axios.post("http://localhost:4200/product", {
27      productName: name,
28      productCode: code,
29      category: category,
30      price: price,
31      imgUrl: img,
32      description: description,
33    })
34      .then(response => {
35        alert("Thêm mới thành công")
36      })
37      .catch(error => {
38        console.log(error)
39      })
40  }
41
```

- Kết quả màn hình:

Thêm hải sản ✕

Tên hải sản:

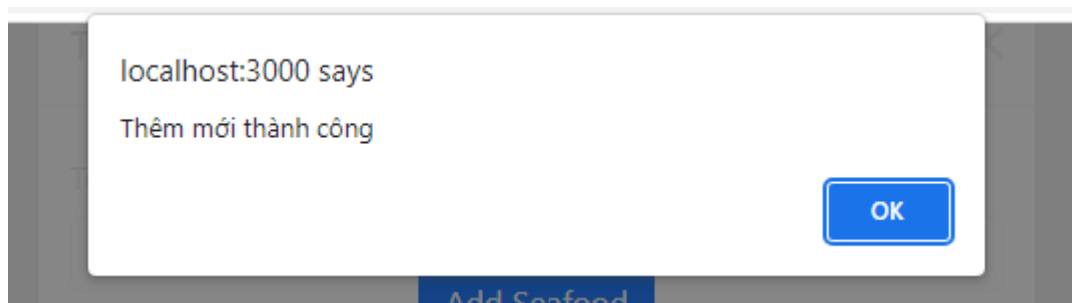
Mã hải sản:

Loại hải sản:

Giá hải sản:



Ảnh hải sản:

Mô tả hải sản:



Manager Seafood

Add Seafood

ID	Name	Code	Category	Price	Image	Description	Actions
1	Cua Bể Sốt Me	CBSM	Cua	100,000đ		Cua bể sốt me chua ngon	<div>Edit</div> <div>Delete</div>
2	Tôm hùm Alaska	THA	Tôm	2000000đ		Tôm Hùm Alaska Hấp Tự Nhiên: Giữ trọn hương vị tự nhiên của thịt Tôm Hùm Alaska: thơm tự nhiên và ngọt đậm đà, từng thớ thịt dai, săn chắc từng khối, ...	<div>Edit</div> <div>Delete</div>

- EDIT:
- Bước 12: Bây giờ ta sẽ viết phần Edit, ở nút Edit ta thêm 1 hàm handleEdit với tham số nhận vào là id của hải sản mình chọn

```
<button className="btn btn-warning me-2" data-bs-toggle="modal" data-bs-target="#exampleModal"
  onClick={() => handleEdit(item.id)}>Edit</button>
```

- Bước 13: Trong handleEdit, ta sẽ lấy id đầu vào để truy ra hải sản có id đó và lấy toàn bộ thông tin đó binding vào các trường trong modal và tạo ra 1 biến id để hứng id hải sản đó vào

```
const [id, setId] = useState()
```

```

const handleEdit = (id) => {
  setIsChanged(!isChanged)
  setIsEdit(true)
  seafood.map(item => {
    if (item.id === id) {
      setId(item.id)
      setName(item.productName)
      setCode(item.productCode)
      setCategory(item.category)
      setDescription(item.description)
      setImg(item.imageUrl)
      setPrice(item.price)
    }
  })
}

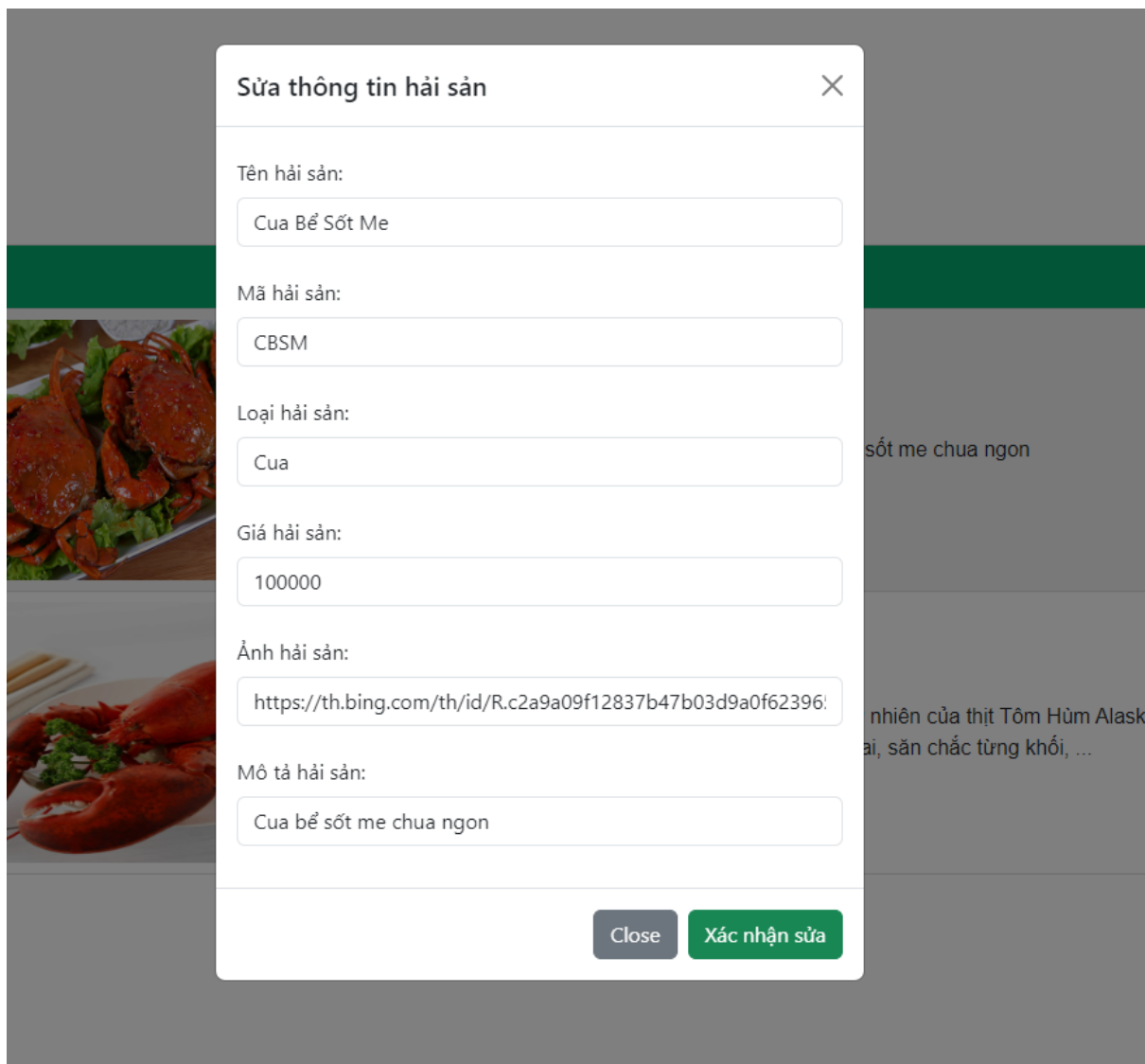
```

```

</form>
<div className="mb-3">
  <label for="product-name" className="col-form-label">Tên hải sản:</label>
  <input type="text" className="form-control" defaultValue={name}
    onChange={(e) => setName(e.target.value)}
  />
</div>
<div className="mb-3">
  <label for="product-name" className="col-form-label">Mã hải sản:</label>
  <input type="text" className="form-control" defaultValue={code}
    onChange={(e) => setCode(e.target.value)}
  />
</div>
<div className="mb-3">
  <label for="product-name" className="col-form-label">Loại hải sản:</label>
  <input type="text" className="form-control" defaultValue={category}
    onChange={(e) => setCategory(e.target.value)}
  />
</div>
<div className="mb-3">
  <label for="product-name" className="col-form-label">Giá hải sản:</label>
  <input type="number" min={1000000} className="form-control" defaultValue={price}
    onChange={(e) => setPrice(e.target.value)}
  />
</div>
<div className="mb-3">
  <label for="product-name" className="col-form-label">Ảnh hải sản:</label>
  <input type="text" className="form-control" defaultValue={img}
    onChange={(e) => setImg(e.target.value)}
  />
</div>
<div className="mb-3">
  <label for="product-name" className="col-form-label">Mô tả hải sản:</label>
  <input type="text" className="form-control" defaultValue={description}
    onChange={(e) => setDescription(e.target.value)}
  />
</div>

```

- Kết quả màn hình:



Sửa thông tin hải sản

Tên hải sản:
Cua Bể Sốt Me

Mã hải sản:
CBSM

Loại hải sản:
Cua

Giá hải sản:
100000

Ảnh hải sản:
<https://th.bing.com/th/id/R.c2a9a09f12837b47b03d9a0f62396>

Mô tả hải sản:
Cua bể sốt me chua ngon

Close Xác nhận sửa

- Bước 14: Tiến hành chỉnh sửa theo ý ta muốn nhưng tại nút Xác nhận sửa ta thêm 1 hàm handleSaveEdit. Trong hàm này ta sẽ dùng axios với phương thức PUT để gọi API chỉnh sửa thông tin hải sản mà mình muốn chỉnh thông qua id của hải sản. Tương tự như các phương thức trên trong hàm .then ta sẽ cho alert bắn ra thông báo khi thành công và in ra lỗi ở console khi thất bại

```
const handleSaveEdit = () => {  
  setIsEdit(false)  
  setIsChanged(!isChanged)  
  axios.put(`http://localhost:4200/product/${id}`, {  
    "productName": name,  
    "productCode": code,  
    "category": category,  
    "price": price,  
    "imgUrl": img,  
    "description": description,  
    "id": id  
  })  
  .then((response) => {  
    alert("Chỉnh sửa thành công")  
  })  
  .catch((error) => { console.log(error) })  
}
```

- Kết quả màn hình:

Sửa thông tin hải sản

×

Tên hải sản:

Cua Bể Sốt Me Chua Ngọt Vắt Chanh Bỏ Vỏ

Mã hải sản:

CBSM

Loại hải sản:

Cua

Giá hải sản:

100000

Ảnh hải sản:

https://th.bing.com/th/id/R.c2a9a09f12837b47b03d9a0f62396?

Mô tả hải sản:

Cua bể sốt me chua ngon



Close

Xác nhận sửa

localhost:3000 says
Chỉnh sửa thành công

Manager Seafood

Add Seafood

ID	Name	Code	Category	Price	Image	Description	Actions
1	Cua Bể Sốt Me Chua Ngọt Vắt Chanh Bỏ Vỏ	CBSM	Cua	100,000đ		Cua bể sốt me chua ngon	<button>Edit</button> <button>Delete</button>
2	Tôm hùm Alaska	THA	Tôm	2000000đ		Tôm Hùm Alaska Hấp Tự Nhiên: Giữ trọn hương vị tự nhiên của thịt Tôm Hùm Alaska: thơm tự nhiên và ngọt đậm đà, từng thớ thịt dai, săn chắc từng khối, ...	<button>Edit</button> <button>Delete</button>

- DELETE:

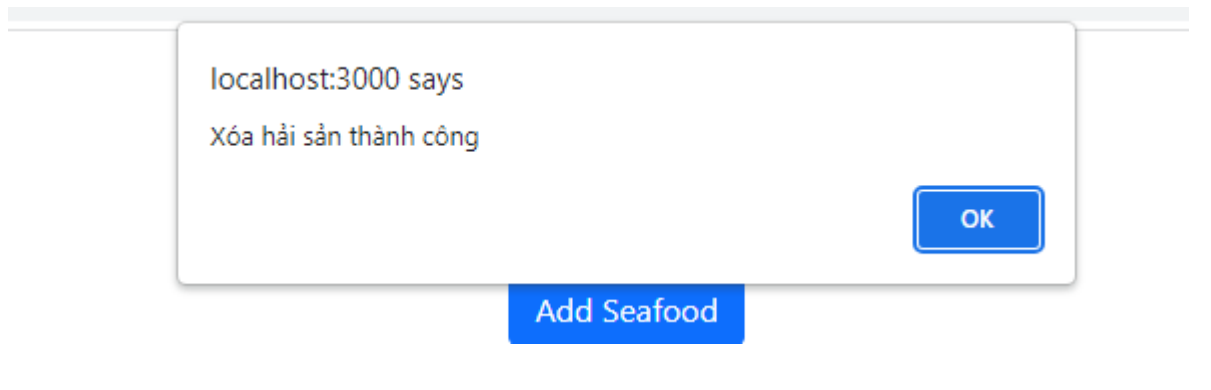
- Bước 15: Trong nút Delete, ta sẽ thêm 1 sự kiện onClick chứa hàm handleDelete và tham số nhận vào là id của hải sản ta chọn

```
<button className="btn btn-danger"
onClick={()=>handleDelete(item.id)}
>Delete</button>
</td>
```

- Bước 16: Trong hàm này, ta sẽ viết axios với phương thức Delete để thực hiện xóa hải sản đó thông qua id nhận vào. Tương tự như các phương thức trên trong hàm .then ta sẽ cho alert bắn ra thông báo khi thành công và in ra lỗi ở console khi thất bại

```
const handleDelete = (id) => {
  axios.delete(`http://localhost:4200/product/${id}`)
    .then((response) => { alert("Xóa hải sản thành công") })
    .catch((error) => { console.log(error); })
}
```


- Kết quả màn hình:



Manager Seafood

[Add Seafood](#)

ID	Name	Code	Category	Price	Image	Description	Actions
2	Tôm hùm Alaska	THA	Tôm	2000000đ		Tôm Hùm Alaska Hấp Tự Nhiên: Giữ trọn hương vị tự nhiên của thịt Tôm Hùm Alaska: thơm tự nhiên và ngọt đậm đà, từng thớ thịt dai, săn chắc từng khối, ...	Edit Delete