

## LAB 5: MODULES

### Bài 1: Quản lý danh sách công việc

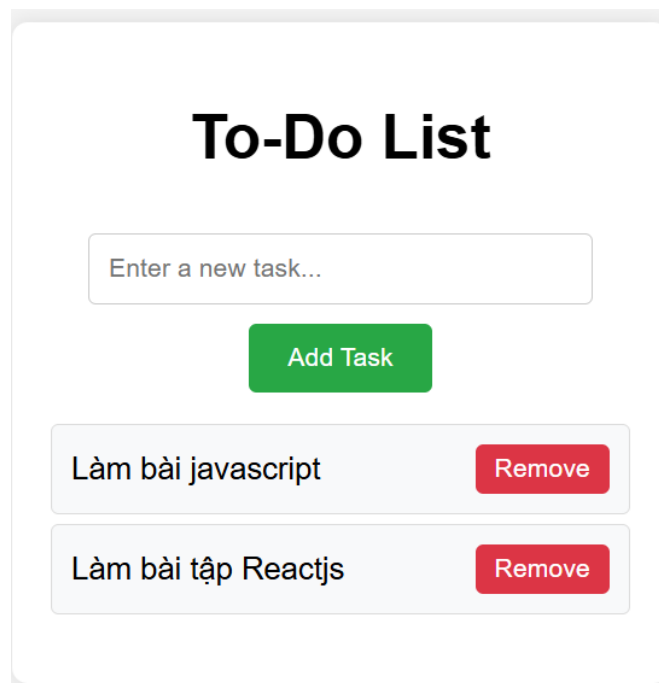
Cấu trúc 1 dự án

```
project-folder/  
|  
├─ index.html  
├─ style.css  
├─ script/  
|   ├─ main.js  
|   ├─ task.js  
|   └─ taskManager.js
```

The screenshot shows a web application titled "To-Do List". It features a text input field with the placeholder "Enter a new task...", a green "Add Task" button, and a list of three tasks. Each task is displayed in a light gray box with a red "Remove" button to its right. The tasks are: "Làm bài javascript", "Làm bài tập Vuejs", and "Làm bài tập Reactjs".

Task	Action
Làm bài javascript	Remove
Làm bài tập Vuejs	Remove
Làm bài tập Reactjs	Remove

Xóa công việc

**Yêu cầu:**

Thực hiện thêm task mới và xóa task

Hướng dẫn

File .js

```
<body>
  <div class="container">
    <h1>To-Do List</h1>
    <input type="text" id="taskInput" placeholder="Enter a new task...">
    <button id="addTaskBtn">Add Task</button>
    <ul id="taskList"></ul>
  </div>
  <script type="module" src="script/main.js"></script>
</body>
```

Thư mục script

File task.js

```
export default class Task {
  constructor(description) {
    this.description = description;
    this.isCompleted = false;
  }

  toggleCompletion() {
    this.isCompleted = !this.isCompleted;
  }
}
```

## File taskManager.js

```
import Task from './task.js';

export default class TaskManager {
  constructor() {
    this.tasks = [];
  }

  addTask(description) {
    const task = new Task(description);
    this.tasks.push(task);
  }

  removeTask(index) {
    this.tasks.splice(index, 1);
  }

  toggleTaskCompletion(index) {
    this.tasks[index].toggleCompletion();
  }

  getTasks() {
    return this.tasks;
  }
}
```

## File main.js

```
import TaskManager from './taskManager.js';

const taskManager = new TaskManager();
const taskInput = document.getElementById('taskInput');
const addTaskBtn = document.getElementById('addTaskBtn');
const taskList = document.getElementById('taskList');

function renderTasks() {
  taskList.innerHTML = '';
  taskManager.getTasks().forEach((task, index) => {
    const li = document.createElement('li');
    li.innerText = task.description;

    const removeBtn = document.createElement('button');
    removeBtn.innerText = 'Remove';
    removeBtn.addEventListener('click', () => {
      taskManager.removeTask(index);
      renderTasks();
    });

    li.appendChild(removeBtn);
    taskList.appendChild(li);
  });
}
```

```
addTaskBtn.addEventListener('click', () => {  
  const taskDescription = taskInput.value.trim();  
  if (taskDescription !== '') {  
    taskManager.addTask(taskDescription);  
    taskInput.value = '';  
    renderTasks();  
  }  
});
```

## Bài 2: Giỏ hàng cho trang web thương mại điện tử

Cấu trúc dự án

shopping-cart/

- |
- |— index.html
- |— style.css
- |— script/
  - |— main.js
  - |— product.js
  - |— cart.js
  - |— ui.js

### Shopping Cart

Tulbot Runhof - \$30.99

Add to Cart

Dolce & Gabbana - \$14.99

Add to Cart

Very sexy - \$15.99

Add to Cart

#### Your Cart

• Tulbot Runhof (x2) - \$61.98

Remove

• Dolce & Gabbana (x1) - \$14.99

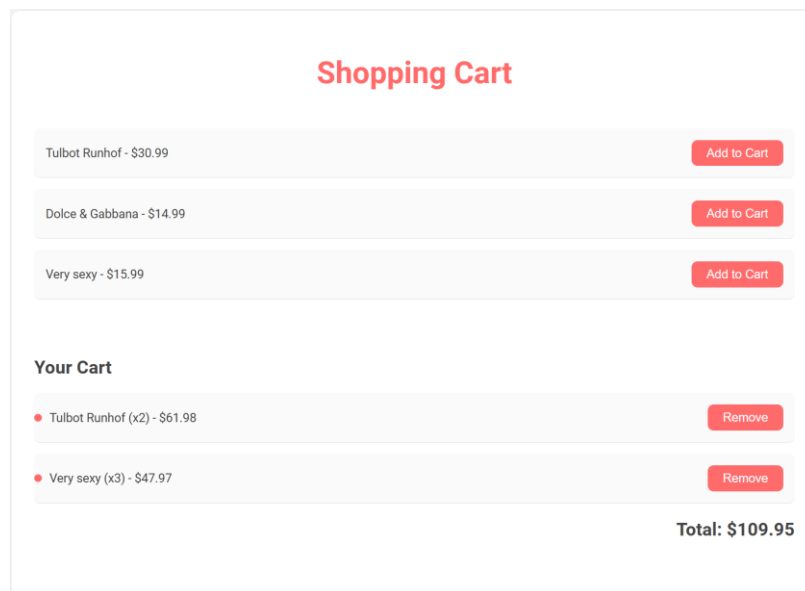
Remove

• Very sexy (x3) - \$47.97

Remove

Total: \$124.94

Xóa



Hướng dẫn

File cart.js

- Thực hiện phương thức: thêm, xóa, tính tổng

File product.js

- Thực hiện tạo 1 class product gồm 2 thuộc tính: name, price

File ui.js

- Thực hiện phương thức hiển thị sản phẩm, hiển thị giỏ hàng


File main.js

Thực hiện:

- Import các file.js cần sử dụng
- Tạo danh sách các sản phẩm
- Tạo giỏ hàng
- Hiển thị sản phẩm
- Thêm sản phẩm vào giỏ hàng
- Xóa sản phẩm khỏi giỏ hàng


**Bài 3: Cải tiến từ bài tập 2**

Shopping Cart




Tulbot Runhof - \$30.99

Add to Cart



Dolce & Gabbana - \$14.99


Add to Cart



Very sexy - \$15.99

Add to Cart


**Your Cart**

•


Tulbot Runhof - \$30.99 each

1

Remove

•


Dolce & Gabbana - \$14.99 each

3

Remove

Total: \$75.96

**Yêu cầu:**

Thực hiện mỗi sản phẩm là 1 hình ảnh tương ứng

Sản phẩm được thêm vào giỏ hàng thì được edit sản phẩm, chỉ giảm số lượng mua tới 1

Hướng dẫn:

Cart.js

```
updateQuantity(index, quantity) {
  this.items[index].quantity = quantity;
}
```

Product.js

```
export default class Product {
  constructor(name, price, image) {
    this.name = name;
    this.price = price;
    this.image = image; // Đường dẫn đến hình ảnh
  }
}
```

Main.js

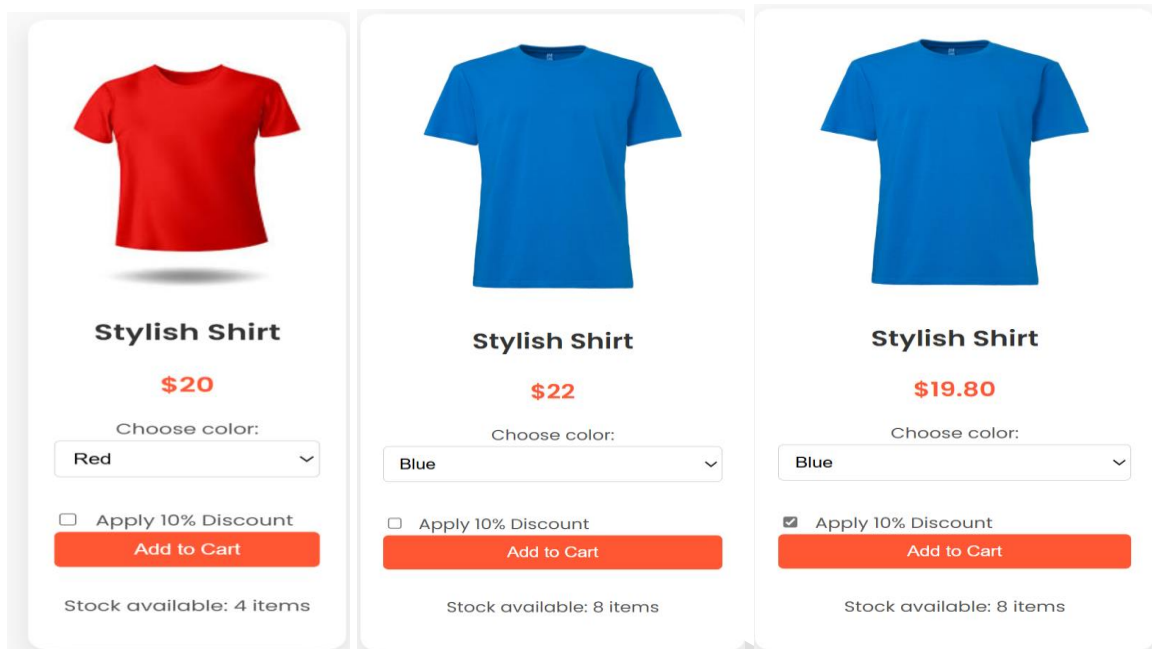
```
// Danh sách sản phẩm
const products = [
  new Product('Tulbot Runhof', 30.99, 'images/1.jpg'),
  new Product('Dolce & Gabbana', 14.99, 'images/2.jpg'),
  new Product('Very sexy', 15.99, 'images/3.jpeg')
];
```

```
// Cập nhật số lượng sản phẩm trong giỏ hàng
document.getElementById('cartItems').addEventListener('input', (e) => {
  if (e.target.classList.contains('quantity-input')) {
    const index = e.target.dataset.index;
    const newQuantity = parseInt(e.target.value, 10);
    if (newQuantity > 0) {
      cart.updateQuantity(index, newQuantity);
      UI.displayCart(cart);
    }
  }
});
```

#### Bài 4: Thực hiện chọn màu sản phẩm

Cấu trúc thư mục

```
/product-color-picker
  /css
    style.css
  /js
    product.js
    main.js
  index.html
```



### Yêu cầu:

- Khi người dùng chọn màu sắc sản phẩm từ menu thả xuống, hình ảnh, số lượng và giá sẽ thay đổi tương ứng.
- Nếu chọn khuyến mãi, giá sẽ được giảm 10%.

### Hướng dẫn

### File index.html

```
<body>
  <div class="product-container">
    <div class="product-image-wrapper">
      
    </div>
    <div class="product-details">
      <h2 id="product-name">Stylish Shirt</h2>
      <p id="product-price">$20.00</p>
      <label for="color-selector">Choose color:</label>
      <select id="color-selector">
        <option value="red" data-price="20.00" data-img="images/red-shirt.jpg" data-stock="4">Red</option>
        <option value="blue" data-price="22.00" data-img="images/blue-shirt.jpg" data-stock="8">Blue</option>
        <option value="green" data-price="18.00" data-img="images/green-shirt.jpg" data-stock="0">Green</option>
      </select>

      <div class="promo-option">
        <input type="checkbox" id="promo" name="promo">
        <label for="promo">Apply 10% Discount</label>
      </div>


      <button id="add-to-cart">Add to Cart</button>
      <p id="stock-info">Stock available: 4 items</p> <!-- Thông tin số lượng tồn kho -->
    </div>
  </div>

  <script type="module" src="script/main.js"></script>
</body>
```



**Bài 5: Cải tiến từ bài tập 4**

LY CHAU



### Stylish Shirt

\$22


Choose color:

Blue

☐ Apply 10% Discount

Add to Cart

Stock available: 7 items



### Stylish Shirt

\$22

Choose color:

Blue


☐ Apply 10% Discount

Add to Cart

Stock available: 4 items

### Shopping Cart

Stylish Shirt (red)




2

Update

\$40.00

Stylish Shirt (blue)



1


Update

\$22.00

Total Price: \$62.00

### Shopping Cart

Stylish Shirt (red)




2

Update

\$40.00

Stylish Shirt (blue)




4

Update

\$88.00

Total Price: \$128.00



Stylish Shirt

\$18.00


Choose color:

Red

☒ Apply 10% Discount

Add to Cart

Stock available: 0 items



Stylish Shirt

\$16.20

Choose color:

Green


☒ Apply 10% Discount

Add to Cart

Stock available: 0 items

Shopping Cart

Stylish Shirt (red)




4

Update

\$80.00

Stylish Shirt (blue)



4


Update

\$88.00

Total Price: \$168.00

Shopping Cart

Stylish Shirt (red)




4

Update

\$80.00

Stylish Shirt (blue)



4

Update

\$88.00

Total Price: \$168.00

CHÂU TRẦN TRÚC LY

TRANG 11

**Yêu cầu:**

- Thực hiện thêm chức năng mua hàng vào trong giỏ hàng
- Nếu mua hàng thì số lượng tồn giảm lại, nếu số lượng bằng 0 thì ẩn nút mua hàng lại
- Người dùng được quyền update số lượng mua trong giỏ hàng, đồng thời được đồng bộ số lượng tồn lại.
- Tính tổng tiền mua hàng trong giỏ hàng