index:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content=
"IE=edge">
    <meta name="viewport" content=
"width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./modules/style.css">
    <title>Product list</title>
</head>
<body class="product-list p-6 bg-blue-50">
    <h1>Product list</h1>
    <script src="./modules/script.js" type="module"></
script>
</body>
</html>
```

monitor.js:

```javascript
export default class Monitor {
    constructor(code = 0, name = "unknown", size = 0, resolution = "1440x1080"
) {
        this._code = code;
        this._name = name;
        this._sizeInInches = size;
        this._stock = 0;
        this._price = 0;
        this._resolution = resolution;
        this._imgURL = "path";
    }

    get name() {
        return this._name;
    }

    get resolution() {
        return this._resolution;
    }

    get code() {
        return this._code;
    }

    get imgURL() {
        return this._imgURL;
    }

    get size() {
        return this._sizeInInches;
    }

    get price() {
        return this._price;
    }

    get stock() {
        return this._stock;
    }

    /**
     * @param {string} path set a new path
     */
    set imgURL(path) {
        this._imgURL = path;
    }

    /**
     * @param {number} amount set a new amount
     */
```
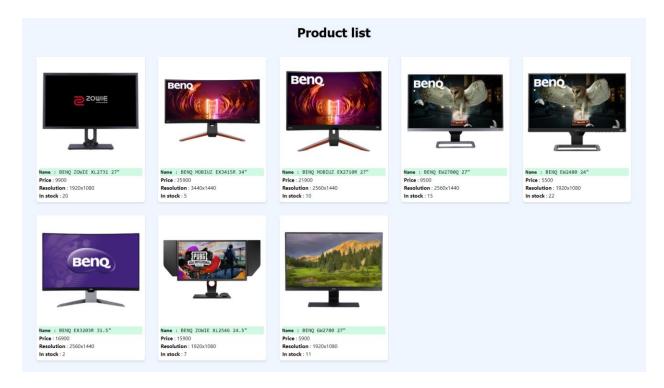
```javascript
    set stock(amount) {
        this._stock += amount;
    }

    /**
     * @param {number} size set a new size
     */
    set size(size) {
        this._sizeInInches = size;
    }

    /**
     * @param {string} name set a new name
     */
    set name(name) {
        this._name = name;
    }

    /**
     * @param {string} code set a new code
     */
    set code(code) {
        this._code = code;
    }

    /**
     * @param {string} resolution set a new resolution
     */
    set resolution(resolution) {
        this._resolution = resolution;
    }

    /**
     * @param {number} size set a new size
     */
    set size(size) {
        this._sizeInInches = size;
    }

    /**
     * @param {number} price set a new price
     */
    set price(price) {
        this._price = price;
    }
}
```
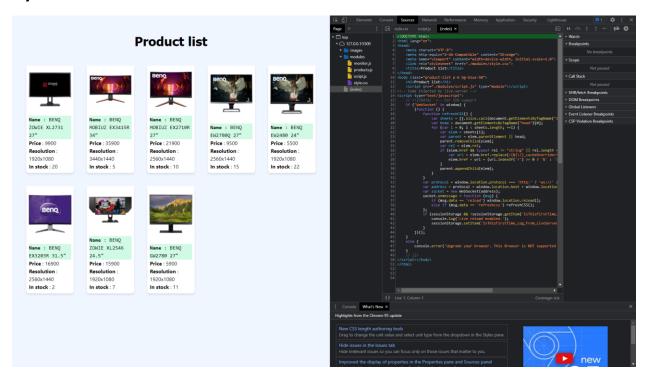
products.js:

```javascript
import Monitor from "./monitor.js";

const products = [];

products.push(new Monitor(1, "BENQ ZOWIE XL2731", 27, "1920x1080"));
products[0]._price = 9900;
products[0]._stock = 20;
products[0]._imgURL = "./images/ZOWIE-XL2731.jpg";

products.push(new Monitor(2, "BENQ MOBIUZ EX3415R", 34, "3440x1440"));
products[1]._price = 35900;
products[1]._stock = 5;
products[1]._imgURL = "./images/MOBIUZ-EX3415R.jpg";

products.push(new Monitor(3, "BENQ MOBIUZ EX2710R", 27, "2560x1440"));
products[2]._price = 21900;
products[2]._stock = 10;
products[2]._imgURL = "./images/MOBIUZ-EX2710R.jpg";

products.push(new Monitor(4, "BENQ EW2780Q", 27, "2560x1440"));
products[3]._price = 9500;
products[3]._stock = 15;
products[3]._imgURL = "./images/EW2780Q.jpg";

products.push(new Monitor(5, "BENQ EW2480", 24, "1920x1080"));
products[4]._price = 5500;
products[4]._stock = 22;
products[4]._imgURL = "./images/EW2480.jpg";

products.push(new Monitor(6, "BENQ EX3203R", 31.5, "2560x1440"));
products[5]._price = 16900;
products[5]._stock = 2;
products[5]._imgURL = "./images/EX3203R.jpg";

products.push(new Monitor(7, "BENQ ZOWIE XL2546", 24.5, "1920x1080"));
products[6]._price = 15900;
products[6]._stock = 7;
products[6]._imgURL = "./images/ZOWIE-XL2546.jpg";

products.push(new Monitor(8, "BENQ GW2780", 27, "1920x1080"));
products[7]._price = 5900;
products[7]._stock = 11;
products[7]._imgURL = "./images/GW2780.jpg";

export { products as default };
```

script.js:

```javascript
import products from "./products.js";

// code here
let body = document.querySelector("body.product-list");

let divContainer = document.createElement("div");
divContainer.className = "product-container";

// add container to body
let container = body.appendChild(divContainer);
container.className = "grid grid-cols-5";

for (const product of products) {
    // add div to container in body
    let divList = document.createElement("div");
    divList.id = product.code;
    divList.className = "flex flex-col bg-white rounded-lg shadow-md m-5";
    let divItem = container.appendChild(divList);

    // add img to div in container
    let img = document.createElement("img");
    img.src = product.imgURL;
    img.alt = product.name;
    img.style = "width: 100%; height: 100%;";
    divItem.appendChild(img);

    // add Description to div in container
    let divDesc = document.createElement("div");
    divDesc.className = "p-2";

    let name = document.createElement("div");
    name.className = "bg-green-100 font-mono flex-1";
    name.innerHTML = `<b>Name</b> : ${product.name} ${product.size}"`;
    divDesc.appendChild(name);

    let price = document.createElement("div");
    price.className = "";
    price.innerHTML = `<b>Price</b> : ${product.price}`;
    divDesc.appendChild(price);

    let res = document.createElement("div");
    res.className = "";
    res.innerHTML = `<b>Resolution</b> : ${product.resolution}`;
    divDesc.appendChild(res);

    let stock = document.createElement("div");
    stock.className = ""
    stock.innerHTML = `<b>In stock</b> : ${product.stock}`;
    divDesc.appendChild(stock);

    divItem.appendChild(divDesc);
}
```

Dynamic source code:

GIT LINK **(This Assignment): https://github.com/xbklyn/INT201-G06-GroupWorks/tree/main/Assignment6**

GIT LINK **(All Assignment): https://github.com/xbklyn/INT201-G06-GroupWorks**