index:

```
<!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="./styles/style.css">
    <link rel="stylesheet" href="./styles/modal-box.css">
    <title>Product list</title>
</head>

<body>
    <div class="toggle-switch">
        <p class="dark-mode">&#x1f319;</p>
        <span>
            <label class="switch">
                <input id="theme-switch" type="checkbox">
                <span class="slider round"></span>
            </label>
        </span>
        <p class="light-mode">&#x1f506;</p>
    </div>
    <div class="search">
        <div class="icon">
            <span>&#9906;</span>
        </div>
        <input type="text" placeholder="Search" id="search">
    </div>
    <div class="shopping-cart">
        <div id="shopping-cart" class="icon" style="background-color: darkcyan; font-size: 1em;">
            <span style="transform: rotate(0);">&#128722;</span>
            <span id="total" style="transform: rotate(0);"><b>0</b></span>
        </div>
    </div>
    <div id="modal-box" class="modal">
        <div class="modal-content">
            <div class="modal-header">
                <span class="close">&times;</span>
                <b>Shopping cart</b>
            </div>
            <div class="modal-body"></div>
        </div>
    </div>
    <script src="./modules/script.js" type="module"></script>
    <script src="./modules/events.js" type="module"></script>
    <script src="./modules/theme.js" type="module"></script>
    <script src="./modules/modalBox.js" type="module"></script>
    <script src="./modules/loadStorage.js" type="module"></script>
</body>

</html>
```

monitor.js:

```
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
     */
    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:

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

const products = [];

products.push(new Monitor(1, "BENQ ZOWIE XL2731", 27, "1920x1080"));
products[0].price = 9900;
products[0].stock = 0;
products[0].imgURL = "./images/ZOWIE-XL2731.jpg";

products.push(new Monitor(2, "BENQ MOBIUZ EX3415R", 34, "3440x1440"));
products[1].price = 35900;
products[1].stock = 59;
products[1].imgURL = "./images/MOBIUZ-EX3415R.jpg";

products.push(new Monitor(3, "BENQ MOBIUZ EX2710R", 27, "2560x1440"));
products[2].price = 21900;
products[2].stock = 104;
products[2].imgURL = "./images/MOBIUZ-EX2710R.jpg";

products.push(new Monitor(4, "BENQ EW2780Q", 27, "2560x1440"));
products[3].price = 9500;
products[3].stock = 156;
products[3].imgURL = "./images/EW2780Q.jpg";

products.push(new Monitor(5, "BENQ EW2480", 24, "1920x1080"));
products[4].price = 5500;
products[4].stock = 222;
products[4].imgURL = "./images/EW2480.jpg";

products.push(new Monitor(6, "BENQ EX3203R", 31.5, "2560x1440"));
products[5].price = 16900;
products[5].stock = 25;
products[5].imgURL = "./images/EX3203R.jpg";

products.push(new Monitor(7, "BENQ ZOWIE XL2546", 24.5, "1920x1080"));
products[6].price = 15900;
products[6].stock = 79;
products[6].imgURL = "./images/ZOWIE-XL2546.jpg";

products.push(new Monitor(8, "BENQ GW2780", 27, "1920x1080"));
products[7].price = 5900;
products[7].stock = 0;
products[7].imgURL = "./images/GW2780.jpg";

export { products as default };
```

script.js:

```
// html management file

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

// code here
let body = document.querySelector("body");
body.id = "product-list";
body.className = "p-6 justify-center";

// add h1 to body
let title = document.createElement("h1");
title.className = "font-bold text-xl text-center";
title.textContent = "Monitor List";
body.appendChild(title);

let divContainer = document.createElement("div");
divContainer.className = "product-container grid grid-cols-5";
divContainer.id = "store-product";

// add container to body
let container = body.appendChild(divContainer);

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 m-5 bg-white rounded-lg transition duration-200 ease-in-out hover:bg-purple-200 shadow-xl transform hover:scale-110 product";
    divList.setAttribute("name", product.name + " " + product.size + '"');
    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%; ";
    img.className = "p-2";
    divItem.appendChild(img);

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

    let name = document.createElement("div");
    name.className = " p-1 bg-green-100 flex-1 m-1";
    name.innerHTML = `<b>Product: </b>${product.name} ${product.size}"`;
    divDetails.appendChild(name);
```

```
    let price = document.createElement("div");
    price.className = "m-1 p-1";
    price.innerHTML = `<b>Price: </b>${new Intl.NumberFormat("th-TH", {
        style: "currency",
        currency: "THB",
    }).format(product.price)}`;
    divDetails.appendChild(price);

    let res = document.createElement("div");
    res.className = "m-1 p-1";
    res.innerHTML = `<b>Resolution: </b>${product.resolution}`;
    divDetails.appendChild(res);

    let stock = document.createElement("div");
    // check if have in stock
    stock.className = "m-1 p-1";
    if (product.stock == 0) {
        let outOfStock = document.createElement("b");
        outOfStock.className = "text-red-500";
        outOfStock.textContent = "Out of Stock";
        stock.appendChild(outOfStock);
    } else {
        stock.innerHTML = `<b>In stock: </b>${product.stock}`;
    }
    divDetails.appendChild(stock);

    // add all detail in item container
    divItem.appendChild(divDetails);

    // add "Add" button
    let divBtn = document.createElement("div");
    divBtn.className = "m-1 p-1";
    let btn = document.createElement("button");
    btn.className = `add-to-cart w-full p-2 rounded ${
        product.stock == 0
            ? "bg-gray-100 text-gray-300 pointer-events-none"
            : "bg-purple-400 text-white transition duration-150 ease-in-out"
    }`;
    btn.innerHTML = "<b><i>Add</i></b>";

    divBtn.appendChild(btn);
    divDetails.appendChild(divBtn);
}
```

events.js:

```
import { add, updateTotalQty } from "./cart.js";

// toggle search bar
document.querySelector(".icon").addEventListener("click", function () {
    document.querySelector("#search").classList.toggle("active");
});

// search function
const productStore = document.querySelectorAll(".product");
const search = document.getElementById("search");

search.addEventListener("keyup", (e) => {
    const text = e.target.value.toLowerCase().trim();
    for (let i = 0; i < productStore.length; i++) {
        let productName = productStore[i].getAttribute("name");
        if (productName.toLowerCase().trim().includes(text)) {
            productStore[i].style.display = "";
        } else {
            productStore[i].style.display = "none";
        }
    }
});

// add to cart function
const addButton = document.querySelectorAll(".add-to-cart");

for (const [id, btn] of addButton.entries()) {
    btn.addEventListener("click", () => {
        add(id);
        updateTotalQty();
    });
}
```

cart.js:

```
import products from "./products/products.js";
import CookieUtil from "./cookieUtil.js";

const cart = {
    itemID: [],
    items: [],
    netPrice: 0,
    totalQuantity: 0,
};

function clearAll() {
    CookieUtil.unset("shopping_cart");
    let tableBody = document.getElementById("tbody");
    while (tableBody.firstChild) {
        tableBody.removeChild(tableBody.lastChild);
    }
    let totalQuantity = document.getElementById("total");
    totalQuantity.innerHTML = "<b>0</b>";
}

function updateTotalQty() {
    let sc = JSON.parse(CookieUtil.getCookie("shopping_cart"));
    if (sc.totalQuantity <= 99) {
        document.getElementById(
            "total"
        ).innerHTML = `<b>${sc.totalQuantity}</b>`;
    } else {
        document.getElementById("total").innerHTML = `<b>99+</b>`;
    }
}

/**
 * @param {number} id id of the button
 */
function remove(id) {
    let sc = JSON.parse(CookieUtil.getCookie("shopping_cart"));
    sc.totalQuantity -= sc.items[id].quantity;
    sc.netPrice -= sc.items[id].totalPrice;
    sc.itemID.splice(id, 1);
    sc.items.splice(id, 1);
    CookieUtil.setCookie("shopping_cart", JSON.stringify(sc, 0, 2), 1);
}
```

```
/**
 * @param {number} id id of the button
 */
function add(id) {
    if (CookieUtil.getCookie("shopping_cart") == null) {
        cart.itemID.push(id + 1);
        cart.items.push({
            productDetails: {
                img: products[id].imgURL,
                productCode: products[id].code,
                productName: products[id].name,
                resolution: products[id].resolution,
                sizeInInches: products[id].size,
            },
            price: products[id].price,
            quantity: 1,
            totalPrice: products[id].price,
        });
        cart.totalQuantity += 1;
        cart.netPrice += products[id].price;
        CookieUtil.setCookie("shopping_cart", JSON.stringify(cart, 0, 2), 1);
    } else {
        let sc = JSON.parse(CookieUtil.getCookie("shopping_cart"));
        if (sc.itemID.includes(id + 1)) {
            const index = sc.itemID.indexOf(id + 1);
            sc.items[index].quantity += 1;
            let price = sc.items[index].price;
            let quantity = sc.items[index].quantity;
            sc.items[index].totalPrice = price * quantity;
            sc.totalQuantity += 1;
        } else {
            sc.itemID.push(id + 1);
            sc.items.push({
                productDetails: {
                    img: products[id].imgURL,
                    productCode: products[id].code,
                    productName: products[id].name,
                    resolution: products[id].resolution,
                    sizeInInches: products[id].size,
                },
                price: products[id].price,
                quantity: 1,
                totalPrice: products[id].price,
            });
            sc.totalQuantity += 1;
        }
        sc.netPrice += products[id].price;
        CookieUtil.setCookie("shopping_cart", JSON.stringify(sc, 0, 2), 1);
    }
}

export { cart, add, remove, updateTotalQty, clearAll };
```

cookieUtil.js:

```
export default class CookieUtil {
    static getCookie(name) {
        console.log(`all cookies: ${document.cookie}`);
        let cookieName = `${encodeURIComponent(name)}=`;
        let cookieStart = document.cookie.indexOf(cookieName);
        let cookieValue = null;
        console.log(`cookieName = ${cookieName}`);
        console.log(`cookieStart = ${cookieStart}`);

        if (cookieStart > -1) {
            let cookieEnd = document.cookie.indexOf(";", cookieStart);
            console.log(`cookieEnd = ${cookieEnd}`);
            if (cookieEnd == -1) {
                cookieEnd = document.cookie.length;
            }
            cookieValue = decodeURIComponent(
                document.cookie.substring(
                    cookieStart + cookieName.length,
                    cookieEnd
                )
            );
            console.log(`cookieValue = ${cookieValue}`);
        }
        return cookieValue;
    }

    static setCookie(name, value, expires) {
        let cookieText = `${encodeURIComponent(name)}=${encodeURIComponent(
            value
        )}`;

        const expireDate = new Date();
        expireDate.setTime(
            expireDate.getTime() + expires * 24 * 60 * 60 * 1000
        );

        cookieText += `; expires=${expireDate.toUTCString()}`;
        // cookieText += `; expires=${expires}`;
        console.log(`cookieText = ${cookieText}`);
        document.cookie = cookieText;
    }

    static unset(name) {
        CookieUtil.setCookie(name, "", new Date(0));
    }
}
```

modalBox.js:

```
import CookieUtil from "./cookieUtil.js";
import { remove, updateTotalQty, clearAll } from "./cart.js";

let modalBody = document.querySelector("div.modal-body");
let table = document.createElement("table");
table.style = "width: 100%;";
modalBody.appendChild(table);

let thead = document.createElement("thead");
thead.id = "thead";
let tbody = document.createElement("tbody");
tbody.id = "tbody";
table.appendChild(thead);
table.appendChild(tbody);

let tr = document.createElement("tr");
thead.appendChild(tr);

const heads = [
    "#",
    "Picture",
    "Product code",
    "Product name",
    "Price",
    "Quantity",
    "Total",
    "Delete",
];
for (const head of heads) {
    let th = document.createElement("th");
    th.textContent = head;
    tr.appendChild(th);
}

function deleteRow(id) {
    let rw = document.querySelector(`#sc-row-${id + 1}`);
    rw.remove();
}

// show shopping cart function
const shoppingCart = document.querySelector("#shopping-cart");
let modal = document.getElementById("modal-box");

shoppingCart.addEventListener("click", () => {
    if (CookieUtil.getCookie("shopping_cart") != null) {
        let sc = JSON.parse(CookieUtil.getCookie("shopping_cart"));
        // console.log(JSON.stringify(cart, 0, 2));
        let tableBody = document.querySelector("#tbody");
        while (tableBody.firstChild) {
            tableBody.removeChild(tableBody.lastChild);
        }
        for (let i = 0; i < sc.items.length; i++) {
            let row = document.createElement("tr");
            row.id = `sc-row-${i + 1}`;
            tbody.appendChild(row);
```

```
        let noCol = document.createElement("td");
        noCol.className = "no-col";
        noCol.textContent = i + 1;
        noCol.style = "text-align: center;";
        row.appendChild(noCol);

        let imageCol = document.createElement("td");
        let productImg = document.createElement("img");
        imageCol.className = "image-col";
        productImg.src = sc.items[i].productDetails.img;
        productImg.alt = sc.items[i].productDetails.productName;
        imageCol.appendChild(productImg);
        row.appendChild(imageCol);

        let productCodeCol = document.createElement("td");
        productCodeCol.className = "product-code-col";
        productCodeCol.textContent = sc.items[i].productDetails.productCode;
        productCodeCol.style = "text-align: center;";
        row.appendChild(productCodeCol);

        let productNameCol = document.createElement("td");
        productNameCol.className = "product-name-col";
        productNameCol.textContent = sc.items[i].productDetails.productName;
        row.appendChild(productNameCol);

        let productPriceCol = document.createElement("td");
        productPriceCol.className = "product-price-col";
        productPriceCol.textContent = new Intl.NumberFormat("th-TH", {
            style: "currency",
            currency: "THB",
        }).format(sc.items[i].price);
        productPriceCol.style = "text-align: center;";
        row.appendChild(productPriceCol);

        let productQuantityCol = document.createElement("td");
        productQuantityCol.className = "product-quantity-col";
        productQuantityCol.textContent = sc.items[i].quantity;
        productQuantityCol.style = "text-align: center;";
        row.appendChild(productQuantityCol);

        let totalPriceCol = document.createElement("td");
        totalPriceCol.className = "total-price-col";
        totalPriceCol.textContent = new Intl.NumberFormat("th-TH", {
            style: "currency",
            currency: "THB",
        }).format(sc.items[i].totalPrice);
        totalPriceCol.style = "text-align: center;";
        row.appendChild(totalPriceCol);
```

```
            let deleteCol = document.createElement("td");
            let deleteButton = document.createElement("button");
            deleteButton.type = "button";
            deleteButton.innerHTML = "&#x1F5D1;";
            deleteButton.className = "delete-button";
            deleteButton.addEventListener("click", () => {
                remove(i);
                updateTotalQty();
                deleteRow(i);
            });

            deleteCol.style = "text-align: center;";
            deleteCol.appendChild(deleteButton);
            row.appendChild(deleteCol);
        }
    }
    modal.style.display = "block";
});

let span = document.getElementsByClassName("close")[0];
span.addEventListener("click", () => {
    modal.style.display = "none";
});

// When the user clicks anywhere outside of the modal, close it
window.onclick = function (event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
};

// Clear all products in cart
const clearAllButton = document.createElement("button");
clearAllButton.id = "clear-all-products";
clearAllButton.className = "m-2 p-2 bg-red-500 text-white rounded-lg";
clearAllButton.innerHTML = "<b>Clear All</b>";
clearAllButton.addEventListener("click", clearAll);
modalBody.appendChild(clearAllButton);
```

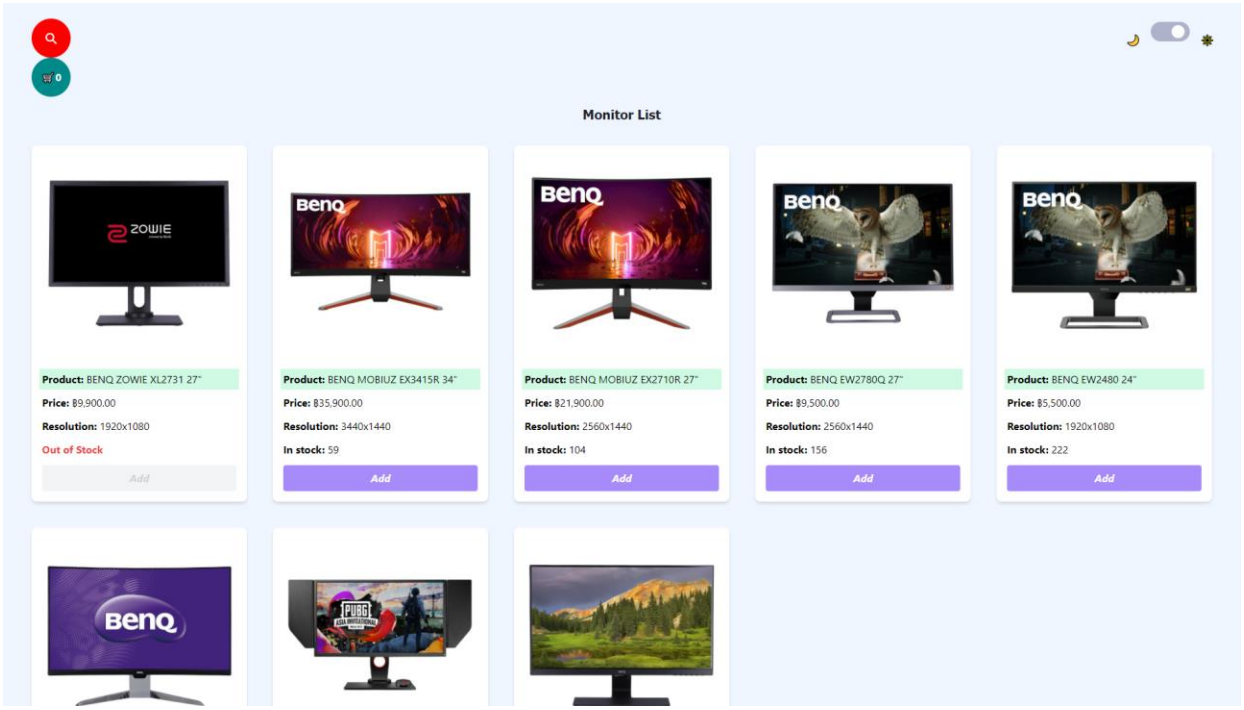
theme.js:

```
//add local storage theme
const themeSwitcher = document.getElementById("theme-switch");

themeSwitcher.checked = false;
function clickHandler() {
    if (this.checked) {
        document.body.classList.remove("light");
        document.body.classList.add("dark");
        localStorage.setItem("theme", "dark");
    } else {
        document.body.classList.add("light");
        document.body.classList.remove("dark");
        localStorage.setItem("theme", "light");
    }
}
themeSwitcher.addEventListener("click", clickHandler);

function checkTheme() {
    const localStorageTheme = localStorage.getItem("theme");

    if (localStorageTheme === "dark") {
        document.body.classList.add(localStorageTheme);
        const themeSwitch = document.getElementById("theme-switch");
        themeSwitch.checked = true;
    } else {
        document.body.classList.add(localStorageTheme);
        const themeSwitch = document.getElementById("theme-switch");
        themeSwitch.checked = false;
    }
}

export { checkTheme as default };
```

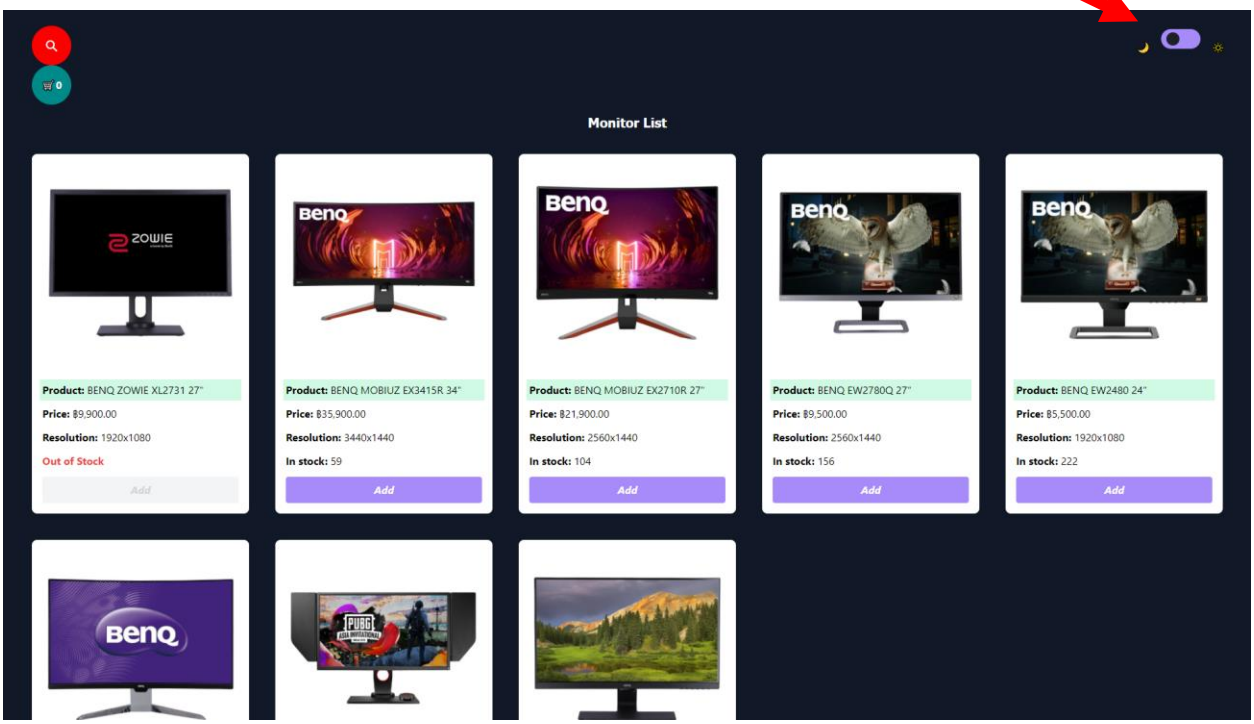
loadStorage.js:

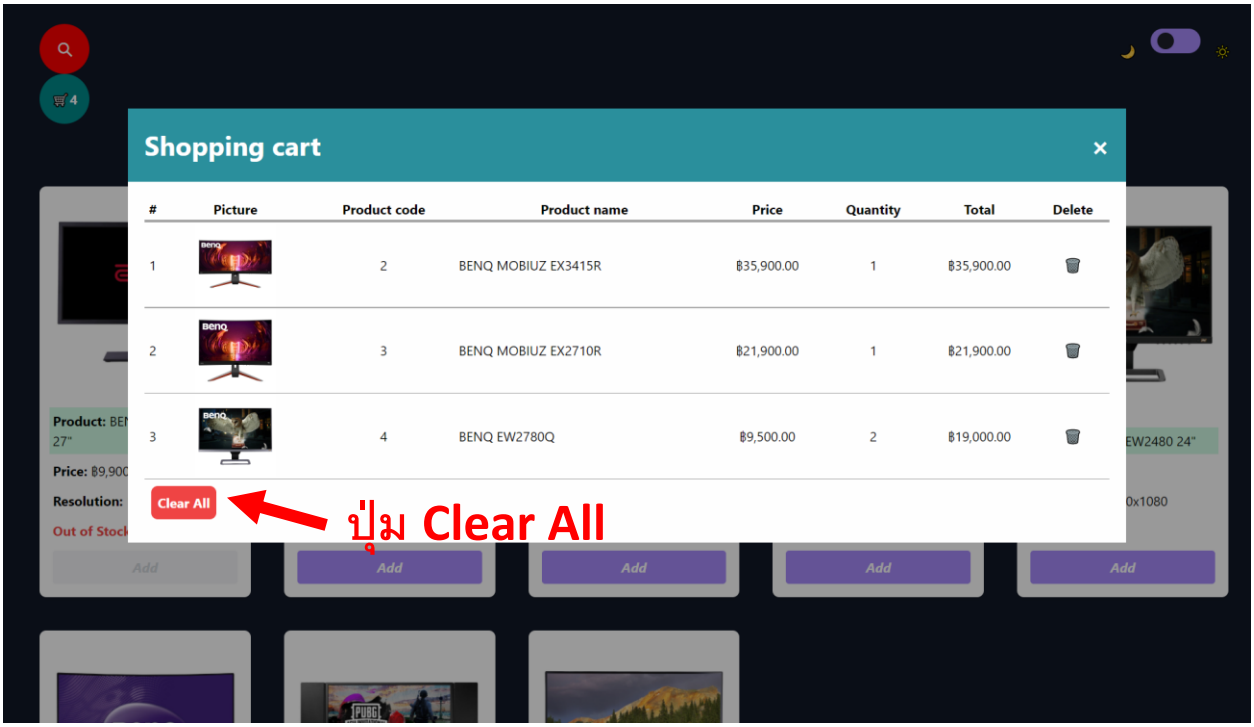
```
import checkTheme from "./theme.js";
import CookieUtil from "./cookieUtil.js";
import { updateTotalQty } from "./cart.js";

window.onload = checkTheme();

if (CookieUtil.getCookie("shopping_cart") != null) {
    updateTotalQty();
}
```

Output:



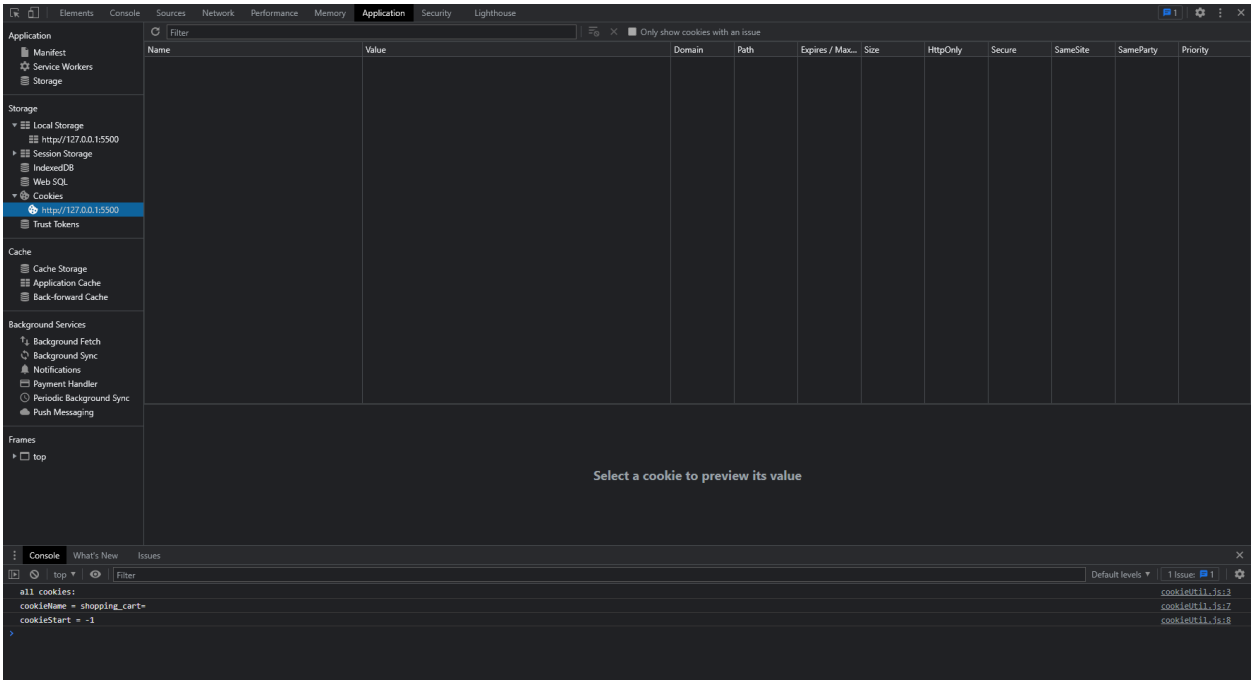
มี Theme



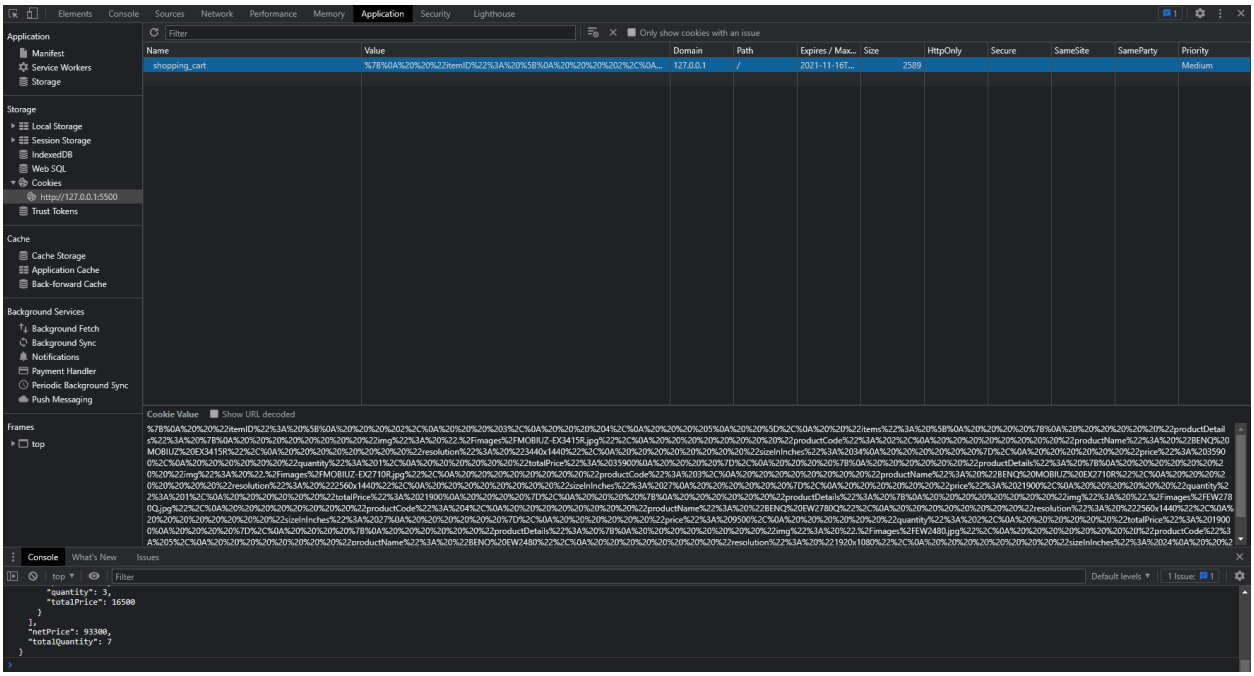

Shopping cart
#
Picture
Product code
Product name
Price
Quantity
Total
Delete
1
2
BENQ MOBIUZ EX3415R
฿35,900.00
1
฿35,900.00
2
3
BENQ MOBIUZ EX2710R
฿21,900.00
1
฿21,900.00
3
4
BENQ EW2780Q
฿9,500.00
2
฿19,000.00
Clear All
ปุ่ม Clear All
Add

Cookies

Before

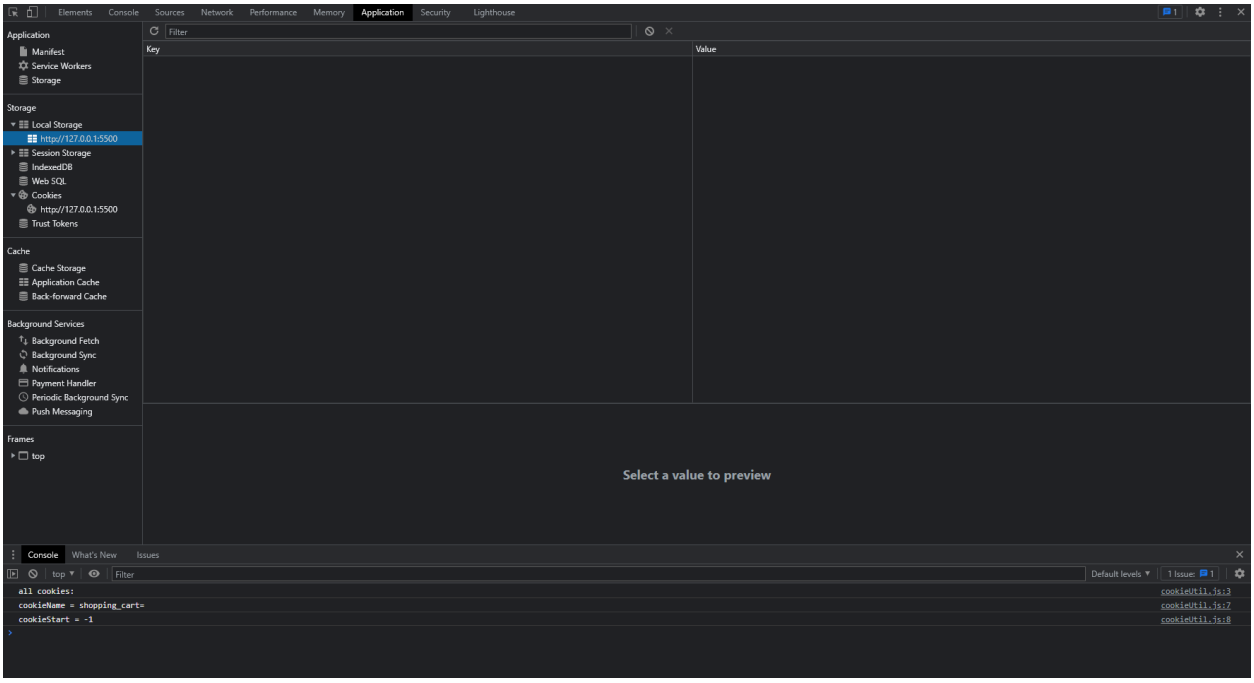


After

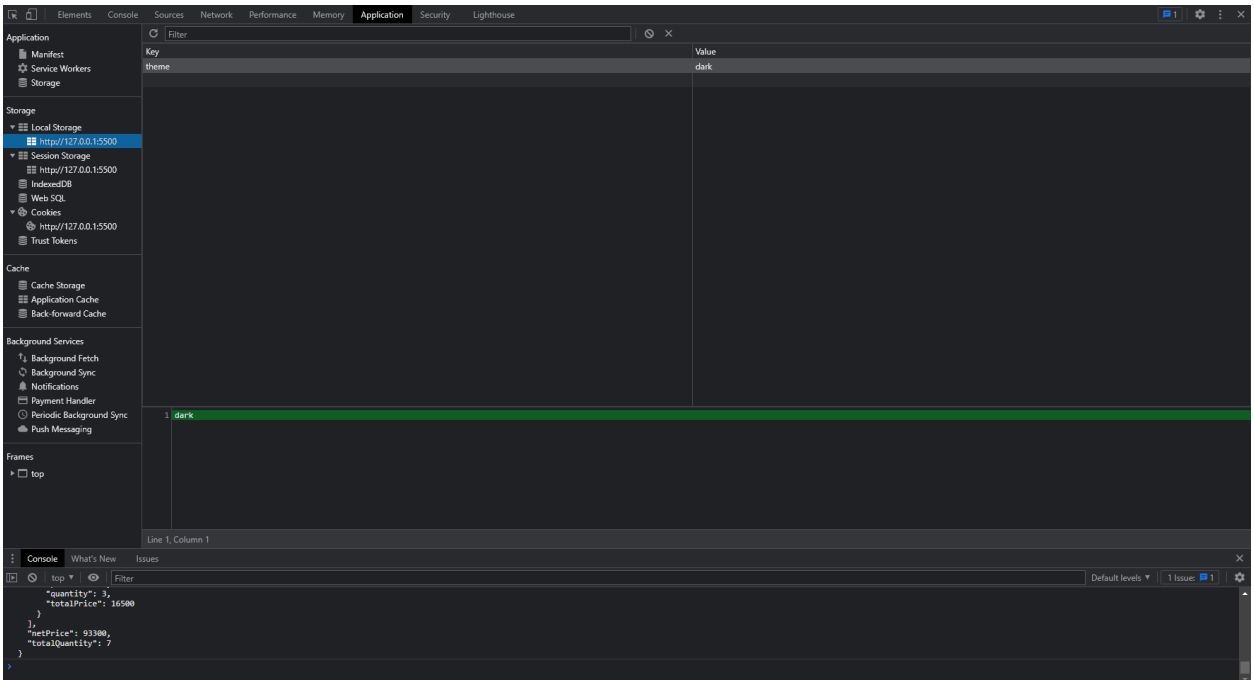


Local Storage

Before



After



GIT LINK* (This Assignment => include work ratio): *Click here

GIT LINK* (All Assignment): *Click here