

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="./modules/style.css">
  <title>Product list</title>
</head>

<body>
  <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: #008080; font-size: 1em;">
      <span style="transform: rotate(0);">&#128722;</span>
      <span id="total" style="transform: rotate(0);"><b>0</b></span>
    </div>
  </div>
  <script src="./modules/script.js" type="module"></script>
  <script src="./modules/events.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:

```

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

// code here
let body = document.querySelector("body");
body.id = "product-list"
body.className = "p-6 bg-blue-50 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 products from "../products/products.js";

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

// toggle search bar
document.querySelector(".icon").addEventListener("click", function () {
  document.querySelector("input").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", () => {
    if (cart.itemID.includes(id + 1)) {
      cart.items.forEach((item) => {
        if (item.productDetails.productCode == id + 1) {
          item.quantity += 1;
          item.totalPrice = item.price * item.quantity;
        }
      });
    }
  });
}

```

```

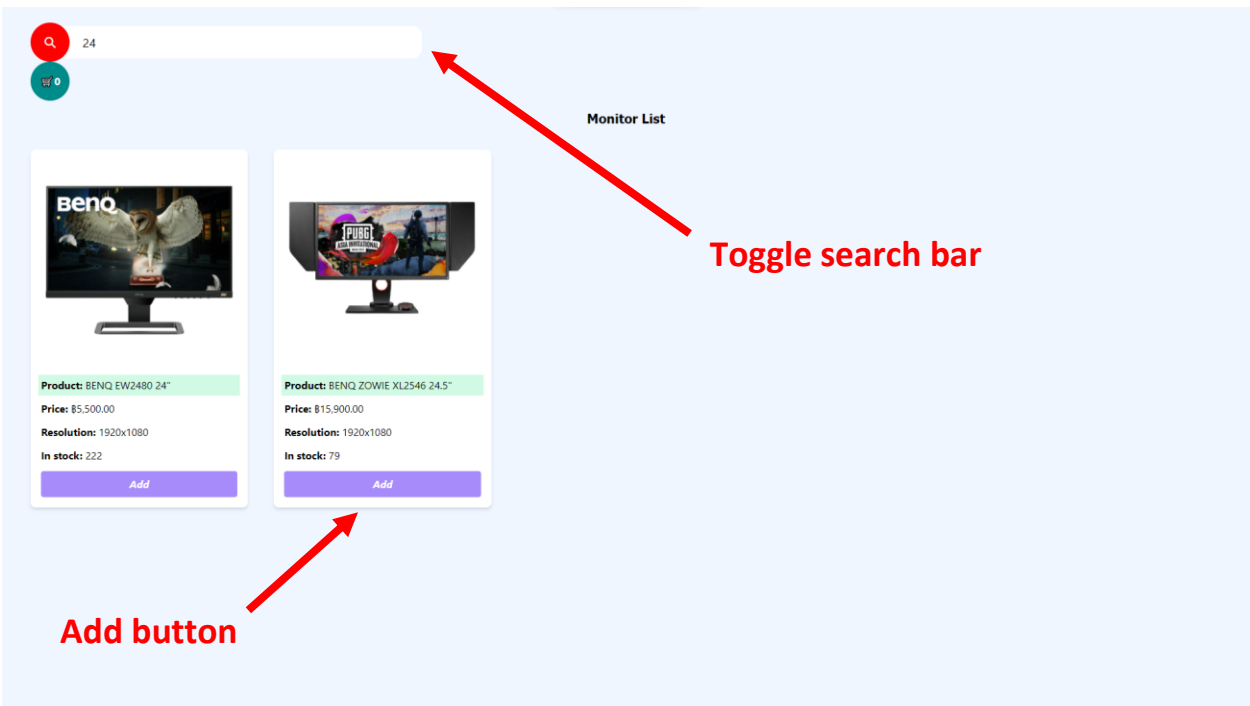
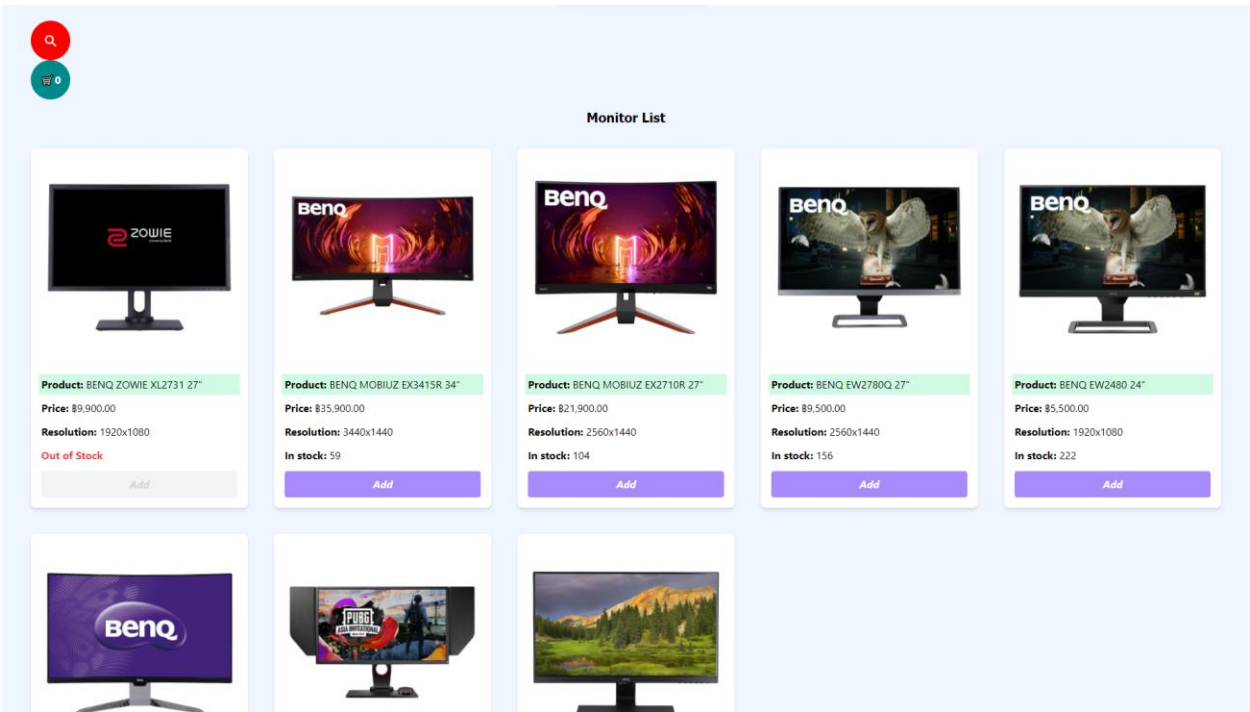
        cart.totalQuantity += 1;
    } else {
        cart.itemID.push(id + 1);
        cart.items.push({
            productDetails: {
                productCode: products[id].code,
                productName: products[id].name,
                resolution: products[id].resolution,
                sizeInInches: products[id].size,
            },
            price: products[id].price,
            quantity: 1,
            totalPrice: products[id].price,
        });
        cart.totalQuantity += 1;
    }
    if (cart.totalQuantity <= 99) {
        document.getElementById(
            "total"
        ).innerHTML = `<b>${cart.totalQuantity}</b>`;
    } else {
        document.getElementById("total").innerHTML = `<b>99+</b>`;
    }
});
}

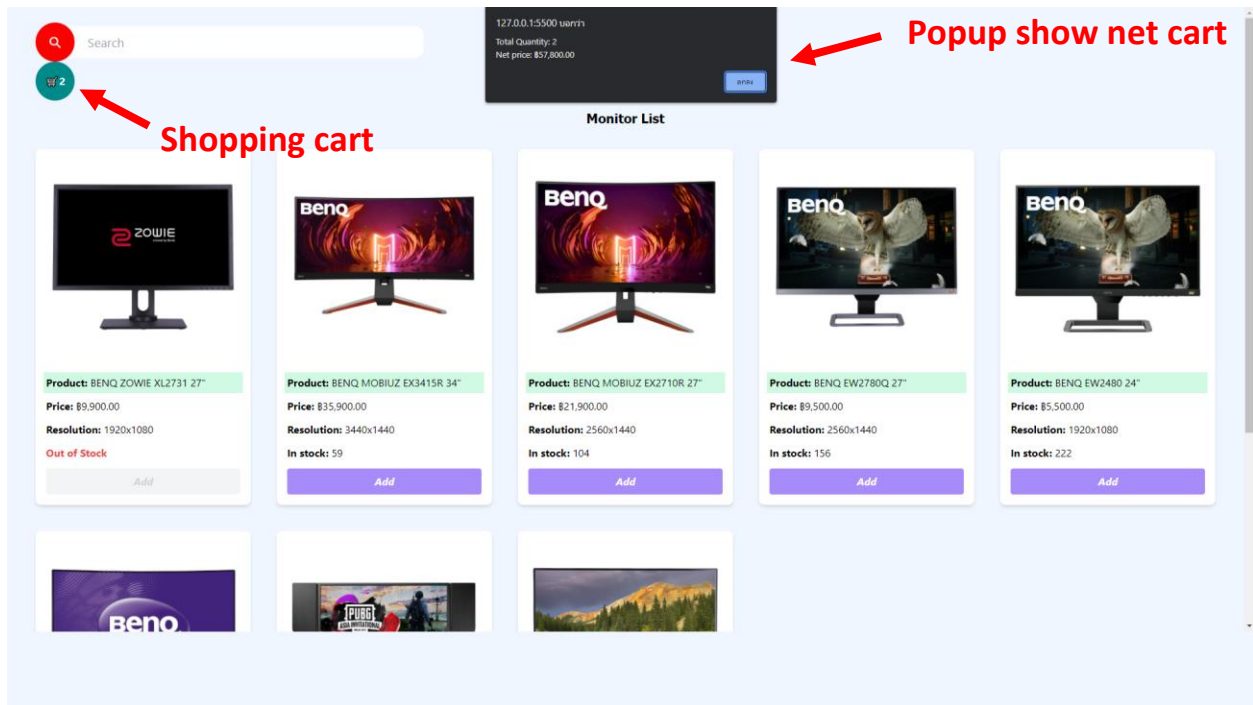
// show net price function
const shoppingCart = document.querySelector("#shopping-cart");

shoppingCart.addEventListener("click", () => {
    let netPrice = 0;
    cart.netPrice = cart.items.reduce(
        (total, item) => total + item.totalPrice,
        netPrice
    );
    console.log(cart);
    alert(
        `Total Quantity: ${
            cart.totalQuantity
        }\nNet price: ${new Intl.NumberFormat("th-TH", {
            style: "currency",
            currency: "THB",
        }).format(cart.netPrice)}`
    );
});
});

```


Output:





Dynamic source code:

```

<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body id="product-list" class="p-6 bg-blue-50 justify-center" data-new-gr-c-s-check-loaded="14.1038.0" data-gr-ext-installed="">
    <div class="search">
      <div class="icon">
        <span>Q</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);>W</span>
        <span id="total" style="transform: rotate(0);>2</span>
      </div>
    </div>
    <script src="/modules/script.js" type="module"></script>
    <script src="/modules/events.js" type="module"></script>
    <!-- Code injected by live-server -->
    <script type="text/javascript">
      <h1 class="font-bold text-xl text-center">Monitor List</h1>
      <div class="product-container grid grid-cols-5" id="store-product">
        <div>
          <img alt="BenQ ZOWIE XL2731 27 inch monitor" data-bbox="145 175 255 245" data-kind="parent" data-rs="4">
          <div>
            <p>Product: BENQ ZOWIE XL2731 27"</p>
            <p>Price: $9,900.00</p>
            <p>Resolution: 1920x1080</p>
            <p>Out of Stock</p>
            <button>Add</button>
          </div>
        </div>
        <div>
          <img alt="BenQ MOBIUZ EX3415R 34 inch monitor" data-bbox="295 175 405 245" data-kind="parent" data-rs="4">
          <div>
            <p>Product: BENQ MOBIUZ EX3415R 34"</p>
            <p>Price: $35,900.00</p>
            <p>Resolution: 3440x1440</p>
            <p>In stock: 59</p>
            <button>Add</button>
          </div>
        </div>
        <div>
          <img alt="BenQ MOBIUZ EX2710R 27 inch monitor" data-bbox="445 175 555 245" data-kind="parent" data-rs="4">
          <div>
            <p>Product: BENQ MOBIUZ EX2710R 27"</p>
            <p>Price: $21,900.00</p>
            <p>Resolution: 2560x1440</p>
            <p>In stock: 104</p>
            <button>Add</button>
          </div>
        </div>
        <div>
          <img alt="BenQ EW2780Q 27 inch monitor" data-bbox="595 175 705 245" data-kind="parent" data-rs="4">
          <div>
            <p>Product: BENQ EW2780Q 27"</p>
            <p>Price: $9,500.00</p>
            <p>Resolution: 2560x1440</p>
            <p>In stock: 156</p>
            <button>Add</button>
          </div>
        </div>
        <div>
          <img alt="BenQ EW2480 24 inch monitor" data-bbox="745 175 855 245" data-kind="parent" data-rs="4">
          <div>
            <p>Product: BENQ EW2480 24"</p>
            <p>Price: $5,500.00</p>
            <p>Resolution: 1920x1080</p>
            <p>In stock: 222</p>
            <button>Add</button>
          </div>
        </div>
      </div>
    </body>
    <script type="text/javascript">
      <div>
        <img alt="BenQ logo" data-bbox="145 355 255 385" data-kind="parent" data-rs="3">
        <img alt="PUBG logo" data-bbox="295 355 405 385" data-kind="parent" data-rs="3">
        <img alt="Landscape image" data-bbox="445 355 555 385" data-kind="parent" data-rs="3">
      </div>
    </script>
  </html>

```

```

Elements Console Sources Network Performance Memory Application Security Lighthouse
<html lang="en">
<head></head>
<body id="product-list" class="p-6 bg-blue-50 justify-center" data-new-gr-c-s-check-loaded="14.1030.0" data-gr-ext-installed="">
  <div class="search"></div> (flex)
  <div class="shopping-cart"></div>
  <script src="/modules/script.js" type="module"></script>
  <script src="/modules/events.js" type="module"></script>
  <!-- Code injected by live-server -->
  <script type="text/javascript">
    // <![CDATA[ <-- For SVG support
    if ('WebSocket' in window) {
      (function () {
        function refreshCSS() {
          var sheets = [].slice.call(document.getElementsByTagName("link"));
          var head = document.getElementsByTagName("head")[0];
          for (var i = 0; i < sheets.length; ++i) {
            var elem = sheets[i];
            var parent = elem.parentElement || head;
            parent.removeChild(elem);
            var rel = elem.rel;
            if (elem.href && typeof rel != "string" || rel.length == 0 || rel.toLowerCase() == "stylesheet") {
              var url = elem.href.replace(/(&|\?)_cacheOverride=(\d+/, '');
              elem.href = url + (url.indexOf('?') >= 0 ? '?' : '?') + '_cacheOverride=' + (new Date().valueOf());
            }
            parent.appendChild(elem);
          }
        }

        var protocol = window.location.protocol === 'http:' ? 'ws://' : 'wss://';
        var address = protocol + window.location.host + window.location.pathname + '/ws';
        var socket = new WebSocket(address);
        socket.onmessage = function (msg) {
          if (msg.data == 'reload') window.location.reload();
          else if (msg.data == 'refreshcss') refreshCSS();
        };
        if (sessionStorage && !sessionStorage.getItem('IsThisFirstTime_Log_From_LiveServer')) {
          console.log('Live reload enabled.');
```

```

Elements Console Sources Network Performance Memory Application Security Lighthouse
<!DOCTYPE html>
<html lang="en">
<head></head>
<body id="product-list" class="p-6 bg-blue-50 justify-center" data-new-gr-c-s-check-loaded="14.1030.0" data-gr-ext-installed="">
  <div class="search"> (flex)
    <span>9</span>
  </div>
  <div class="shopping-cart">
    <div id="shopping-cart" class="icon" style="background-color: darkcyan; font-size: 1em;"> (flex) -- $0
      <span style="transform: rotate(0);">
      <span id="total" style="transform: rotate(0);">
    </div>
    <script src="/modules/script.js" type="module"></script>
    <script src="/modules/events.js" type="module"></script>
    <!-- Code injected by live-server -->
    <script type="text/javascript"></script>
    <h1 class="font-bold text-xl text-center">Monitor List</h1>
    <div class="product-container grid grid-cols-5" id="store-product"></div> (grid)
  </body>
  <grammaly-desktop-integration data-grammaly-shadow-root="true"></grammaly-desktop-integration>
</html>

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility
Ancestors All Framework Listeners
click
  <div#shopping-cart.icon Remove events.js:72
    useCapture: false
    passive: false
    once: false
    handler: () => {
      [[Scopes]]: Scopes[2]
      1: Global {window: Window, self: Window, document: document, name: "", location: Location, ...}
      0: Module {cart: {-}, productStore: NodeList(8), products: Array(8)}
      [[Prototype]]: f ()
      [[Scopes]]: Scopes[0]
      [[Prototype]]: Object
      [[FunctionLocation]]: <unknown>
      set caller: f ()
      get caller: f ()
      set arguments: f ()
      get arguments: f ()
      Symbol(Symbol.hasInstance): f [Symbol.hasInstance]()
      toString: f toString()
      name: ""
      length: 0
      constructor: f Function()
      caller: (...)
      call: f call()
      bind: f bind()
      arguments: (...)
      apply: f apply()
      [[FunctionLocation]]: <unknown>
      caller: (...)
      arguments: (...)
      name: ""
      length: 0
    }
  </div>

```

The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays a JSON object representing a list of products. The object has the following structure:

```

{itemID: Array(4), items: Array(4), netPrice: 150100, totalQuantity: 7}
  itemID: Array(4)
    0: 2
    1: 3
    2: 4
    3: 5
    length: 4
  items: Array(4)
    0: {
      price: 35900,
      productDetails: {productCode: 2, productName: 'BENQ MOBIUZ EX3415R', resolution: '3440x1440', sizeInInches: 34},
      quantity: 3,
      totalPrice: 107700
    }
    1: {
      price: 21900,
      productDetails: {productCode: 3, productName: 'BENQ MOBIUZ EX2710R', resolution: '2560x1440', sizeInInches: 27},
      quantity: 1,
      totalPrice: 21900
    }
    2: {
      price: 9500,
      productDetails: {productCode: 4, productName: 'BENQ EW2780Q', resolution: '2560x1440', sizeInInches: 27},
      quantity: 1,
      totalPrice: 9500
    }
    3: {
      price: 5500,
      productDetails: {productCode: 5, productName: 'BENQ EW2480', resolution: '1920x1080', sizeInInches: 24},
      quantity: 2,
      totalPrice: 11000
    }
  netPrice: 150100
  totalQuantity: 7
  
```

The console also shows 'Live reload enabled.' at the top and a filter bar with 'top' and 'Filter' options. The right side of the console shows 'Default levels', '1 issue', and '1 hidden'.

GIT LINK (This Assignment => include work ratio): [Click here](#)

GIT LINK (All Assignment): [Click here](#)