**1.1**

**Overall Code**

let submitBtn = document.**querySelector**("#btn-submit");

submitBtn.**addEventListener**("click" , () => {

*//get value in input tag*

    let item = document.**getElementById**("item").value;

    let qty = document.**getElementById**("qty").value;

*//get list*

    let orderList = document.**querySelector**("#buying-list");

    let list = document.**createElement**("li");

    list.innerHTML += `${item} , ${qty} `;

    list.id = `${item}`

*//insert image*

    let img = document.**createElement**("img");

    img.src = "./images/pending.svg";

*//add event to img*

    img.**addEventListener**("click" , () => {

        img.src = "./images/done.svg";

        list.style = "text-decoration: line-through; color:grey";

*//remove 1 out from pending*

        let countPendingItems = document.**getElementById**("countPendingItems");

        countDoneItems.innerHTML = `${parseInt(countDoneItems.innerHTML) + 1}`;

*//add 1 in to done*

        let countDoneItems = document.**getElementById**("countDoneItems");

        countPendingItems.innerHTML = `${parseInt(countPendingItems.innerHTML) - 1}`;

    })

    list.**appendChild**(img);

    orderList.**appendChild**(list);

*//reset text field*

    document.**getElementById**("qty").value = "";

    document.**getElementById**("item").value = "";

    document.**getElementById**("item").**focus**();

*//set pending item +=1*

    let pending = document.**getElementById**("countPendingItems");

    pending.innerHTML = document.**querySelectorAll**("li").length;

*//set storage*

})

let clearBtn = document.**getElementById**("btn-clear");

clearBtn.**addEventListener**("click" , () => {

    if (document.**querySelectorAll**("li").length == 0){

**alert**("No item in pending list.")

    }else{

**alert**("Remove all item in pending list!");

        let li = document.**querySelectorAll**("li");

        for (let l of li){

            l.parentNode.**removeChild**(l);

        }

    let pending = document.**getElementById**("countPendingItems");

    pending.innerHTML = "0";

    }

})

**1.2.1**

**Output**

Before After

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Source Code

let submitBtn = document.**querySelector**("#btn-submit");

submitBtn.**addEventListener**("click" , () => {

*//get value in input tag*

    let item = document.**getElementById**("item").value;

    let qty = document.**getElementById**("qty").value;

*//get list*

    let orderList = document.**querySelector**("#buying-list");

    let list = document.**createElement**("li");

    list.innerHTML += `${item} , ${qty} `;

*//insert image*

    let img = document.**createElement**("img");

    img.src = "./images/pending.svg";

    list.**appendChild**(img);

    orderList.**appendChild**(list);

*//reset text field*

    document.**getElementById**("qty").value = "";

    document.**getElementById**("item").value = "";

    document.**getElementById**("item").**focus**();

*//set pending item +=1*

    let pending = document.**getElementById**("countPendingItems");

    pending.innerHTML = document.**querySelectorAll**("li").length;

})

**1.2.2**

**Output**

Before

Graphical user interface, text, application

Description automatically generated

After

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

Source Code

let clearBtn = document.**getElementById**("btn-clear");

clearBtn.**addEventListener**("click" , () => {

    if (document.**querySelectorAll**("li").length == 0){

**alert**("No item in pending list.")

    }else{

**alert**("Remove all item in pending list!");

        let li = document.**querySelectorAll**("li");

        for (let l of li){

            l.parentNode.**removeChild**(l);

        }

    let pending = document.**getElementById**("countPendingItems");

    pending.innerHTML = "0";

    }

})

**1.2.3**

**Output**

Before

Graphical user interface, text, application

Description automatically generated

After

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Source Code

let submitBtn = document.**querySelector**("#btn-submit");

let submitBtn = document.**querySelector**("#btn-submit");

submitBtn.**addEventListener**("click" , () => {

*//get value in input tag*

    let item = document.**getElementById**("item").value;

    let qty = document.**getElementById**("qty").value;

*//get list*

    let orderList = document.**querySelector**("#buying-list");

    let list = document.**createElement**("li");

    list.innerHTML += `${item} , ${qty} `;

*//insert image*

    let img = document.**createElement**("img");

    img.src = "./images/pending.svg";



*//add event to img*

    img.**addEventListener**("click" , () => {

        img.src = "./images/done.svg";

        list.style = "text-decoration: line-through; color:grey";

*//remove 1 out from pending*

        let countPendingItems = document.**getElementById**("countPendingItems");

        countPendingItems.innerHTML = `${parseInt(countPendingItems.innerHTML) - 1}`;

*//add 1 in to done*

        let countDoneItems = document.**getElementById**("countDoneItems");

        countDoneItems.innerHTML = `${parseInt(countDoneItems.innerHTML) + 1}`;

    })

    list.**appendChild**(img);

.

.

Other Codes