# Problem 2.Expense Tracker

**Environment Specifics**

Please be aware that every JS environment may **behave differently** when executing code. Certain things that work in the browser are not supported in **Node.js**, which is the environment used by **Judge**.

The following actions are **NOT** supported:

* **.forEach()** with **NodeList** (returned by **querySelector()** and **querySelectorAll()**)
* **.forEach()** with **HTMLCollection** (returned by **getElementsByClassName()** and **element.children**)
* using the **spread-operator** (**...**) to convert a **NodeList** into an array
* **append()** (use only **appendChild()**)
* **prepend()**
* **replaceWith()**
* **replaceAll()**
* **closest()**
* **replaceChildren()**

If you want to perform these operations, you may use **Array.from()** to first convert the collection into an array.

**Use the provided skeleton to solve this problem.**

**Note**: You **can't** and you have no permission to **change** directly the given HTML code (index.html file).



# Your Task

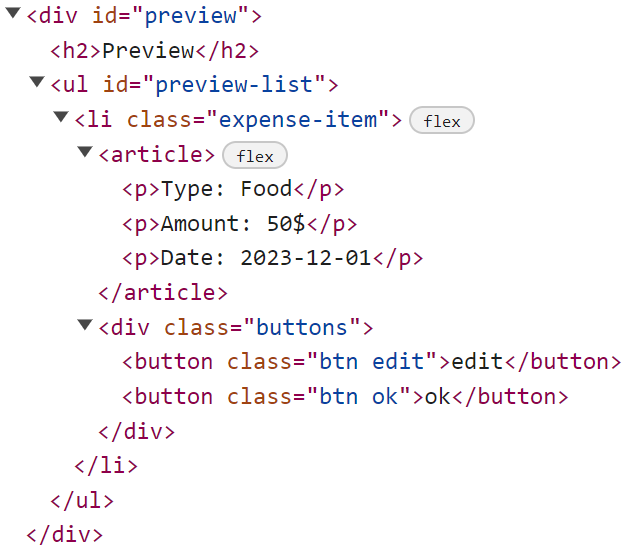
**Write the missing JavaScript code** to make the **Expense Tracker** work as expected:

* + **Expense Type, Amount,** and **Date** should be **non**-**empty** **strings**. If any of them are empty, the program should not do anything.

# Get the information from the form

When you click the **[Add]** button, the information from the input fields must be added to the <ul> with the id "preview-list",**[Add]** button must be **disabled** and **the input fields should be cleared**.

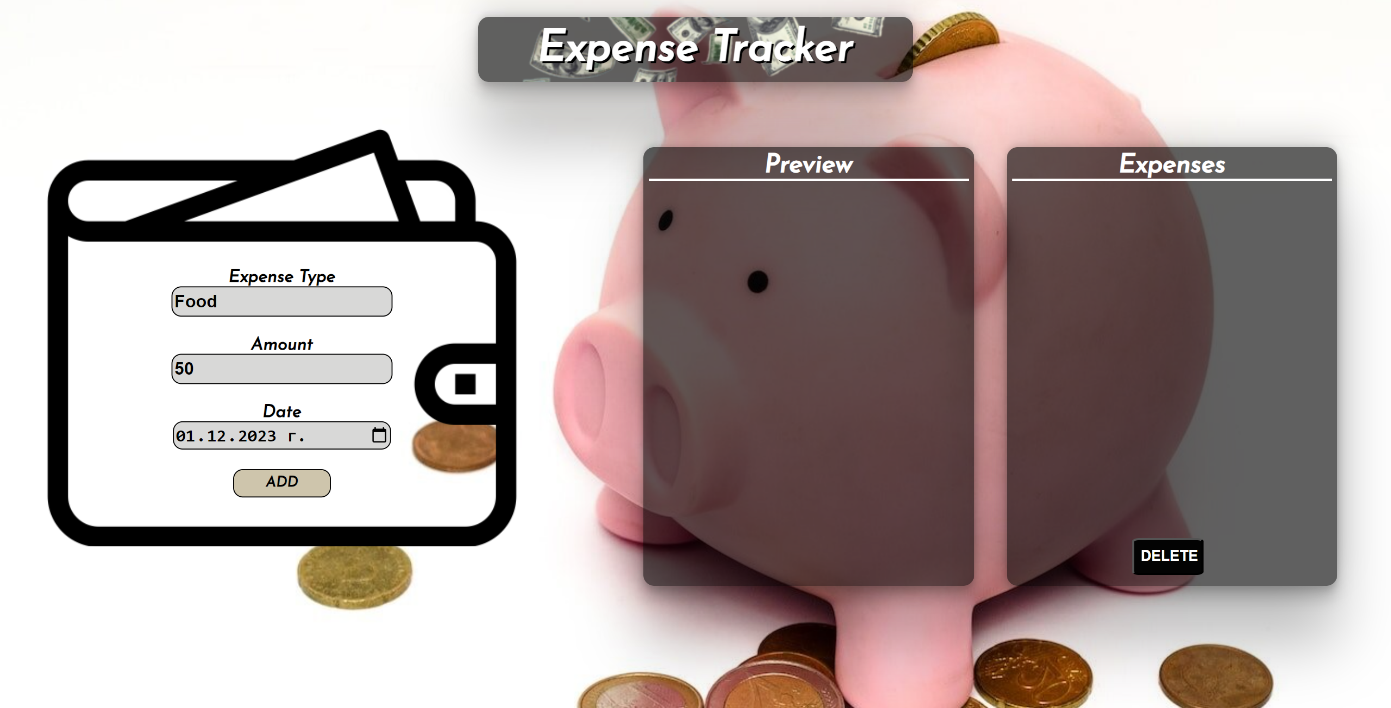
The HTML structure should look like this:



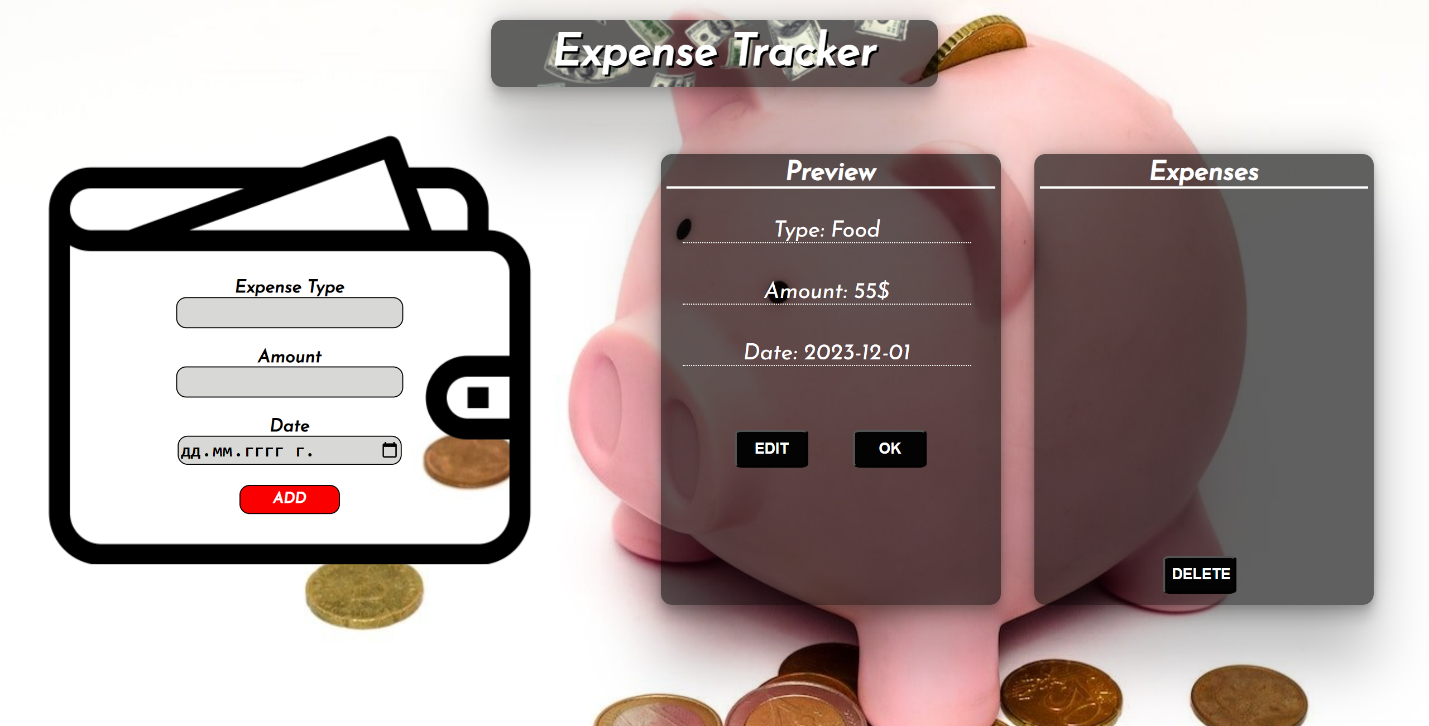


# Edit the Information

When the **[Edit]** button is clicked, the information from the post must be sent to the input fields on the left side, the record should be deleted from the <ul> "preview-list" and **[Add]** button must be **enabled** again.



After editing the information, add a new item to the <ul> "preview-list" with the updated information.



# Add to Expenses

When you click the **[Ok]** button, the record must be **deleted** from the <ul>withid"preview-list" and appended to the <ul> with id "expenses-list".

The **buttons [Edit]** and **[Ok]** should be removed from the <li> element and the **[Add]** button must be **enabled** again.

# Delete Expenses

When the **[Delete]** button is clicked, you must **reload** the application**.**

# Submission

Submit only your **solve()** function.

// function confirmInfo(e) {

    //     previewLi.innerHTML = ""

    //     currentFood = e.target.parentNode

    //     currentFood.querySelector(".edit").remove()

    //     currentFood.querySelector(".ok").remove()

    //     expensesLi.appendChild(currentFood)

    //     addBtn.disabled = true

    // }

window.addEventListener("load", solve);

function solve() {

    const addBtn = document.getElementById("add-btn")

    const expense = document.getElementById("expense")

    const amount = document.getElementById("amount")

    const date = document.getElementById("date")

    const previewLi = document.getElementById("preview-list")

    const expensesLi = document.getElementById("expenses-list")

    const deleteBtn = document.querySelector(".delete")

    let formElement = document.querySelector("form") ///

    let editBtn, okBtn

    addBtn.addEventListener("click", addInfo)

    deleteBtn.addEventListener("click", clearInfo)

    let food = {}

    function addInfo(e) {

        e.preventDefault()

        if (expense.value && amount.value && date.value) {

            food = {

                "expense": expense.value,

                "amount": amount.value,

                "date": date.value,

            }

            let newLi = document.createElement("li")

            newLi.className = "expense-item"

            newLi.innerHTML = `

            <article>

              <p>Type: ${expense.value}</p>

              <p>Amount: ${amount.value}$</p>

              <p>Date: ${date.value}</p>

            </article>

            <div class="buttons">

             <button class="btn edit">edit</button>

             <button class="btn ok">ok</button>

             </div>

            `

            previewLi.appendChild(newLi)

            expense.value = ""

            amount.value = ""

            date.value = ""

            editBtn = newLi.querySelector(".edit")

            okBtn = newLi.querySelector(".ok")

            editBtn.addEventListener("click", editInfo)

            okBtn.addEventListener("click", confirmInfo)

            addBtn.disabled = true

        }

    }

    function editInfo() {

        expense.value = food.expense

        amount.value = food.amount

        date.value = food.date

        previewLi.innerHTML = ""

        addBtn.disabled = false

    }

    // function confirmInfo(e) {

    //     // const currentFood = e.target.closest('.expense-item');

    //     const currentFood = e.target.parentNode.parentNode

    //     console.log(currentFood)

    //     currentFood.querySelector(".edit").remove()

    //     currentFood.querySelector(".ok").remove()

    //     expensesLi.appendChild(currentFood);

    //     previewLi.innerHTML = ""

    //     addBtn.disabled = false;

    //     formElement.reset() ///

    // }

    function confirmInfo(e) {

        const currentFood = e.target.closest('.expense-item')

        let newLi = document.createElement("li")

        newLi.className = "expense-item"

        newLi.innerHTML = `

        <article>

            <p>Type: ${currentFood.querySelector("p:nth-child(1)").textContent.split(': ')[1]}</p>

            <p>Amount: ${currentFood.querySelector("p:nth-child(2)").textContent.split(': ')[1]}</p>

            <p>Date: ${currentFood.querySelector("p:nth-child(3)").textContent.split(': ')[1]}</p>

        </article>

    `

        expensesLi.appendChild(newLi);

        currentFood.remove();

        previewLi.innerHTML = "";

        addBtn.disabled = false;

        formElement.reset();

    }

    function clearInfo() {

        location.reload()

         }

}