Молдавский Государственный Университет  
Факультет Математики и Информатики  
Департамент Информатики

**Лабораторная работа №3  
по курсу “JavaScript/TypeScript”**

Проверил: Nartea Nichita  
Выполнил: студент группы I2302,  
Михайлов Пётр

Кишинев, 2024

**Цель индивидуальной работы:**

Ознакомить студентов с основами взаимодействия JS с DOM-деревом на основе веб-приложения для учета личных финансов.

**Описание задания:**

Разработать веб-страницу с помощью взаимодействия JS с DOM-деревом для учета личных финансов. Создать таблицу, куда будут добавляться транзакции, таблица должна содержать следующие столбцы: id, дата и время, категория транзакции, краткое описание транзакции и удаление транзакции с помощью кнопки.

**Структура проекта:**

1. Файл index.html: основная HTML-структура веб-страницы.
2. Файл style.css: определение стилей страницы.
3. Файл script.js: основной файл JavaScript

**Анализ кода:**

script.js

1.Добавление новой транзакции:

В функции addTransaction данные считываются из полей формы.

Проверяется корректность введенных данных.

Создается объект транзакции и добавляется в массив transactions.

Затем вызывается функция updateTable для обновления отображения таблицы.

function addTransaction() {

    const date = document.getElementById('date').value;

    const amount = parseFloat(document.getElementById('amount').value);

    const category = document.getElementById('category').value;

    const description = document.getElementById('description').value;

    if (!date || isNaN(amount) || !description) {

        alert("Пожалуйста, заполните все поля корректно.");

        return;

    }

    const transaction = {

        id: transactions.length + 1,

        date: date,

        amount: amount,

        category: category,

        description: description

    };

    transactions.push(transaction);

    updateTable();

2. Обновление таблицы транзакций:

В функции updateTable выбирается тело таблицы.

Создается новая строка и заполняется данными о последней добавленной транзакции.

Строке присваивается класс для стилизации в зависимости от значения транзакции.

function updateTable() {

    const tableBody = document.getElementById('transactionsTable').getElementsByTagName('tbody')[0];

    const transaction = transactions[transactions.length - 1];

    const row = tableBody.insertRow();

    row.className = transaction.amount >= 0 ? 'green' : 'red';

    row.innerHTML = `

        <td>${transaction.id}</td>

        <td>${transaction.date}</td>

        <td>${transaction.category}</td>

        <td>${transaction.description.split(' ').slice(0, 4).join(' ')}</td>

        <td><button onclick="removeTransaction(${transaction.id})">Удалить</button></td>

    `;

}

3.Удаление транзакции:

Функция removeTransaction удаляет транзакцию из массива transactions по заданному идентификатору.

Затем происходит обновление таблицы и общей суммы транзакций.

function removeTransaction(id) {

    const tableBody = document.getElementById('transactionsTable').getElementsByTagName('tbody')[0];

    transactions = transactions.filter(transaction => transaction.id !== id);

    while (tableBody.firstChild) {

        tableBody.removeChild(tableBody.firstChild);

    }

    transactions.forEach(transaction => updateTable());

    calculateTotal();

}

4.Вычисление общей суммы транзакций:

Функция calculateTotal использует метод reduce для вычисления суммы всех транзакций.

Результат выводится на страницу.

function calculateTotal() {

    const total = transactions.reduce((acc, cur) => acc + cur.amount, 0);

    document.getElementById('totalDisplay').innerText = 'Общая сумма: ' + total.toFixed(2);

}

5.Отображение подробного описания:

Обработчик события для таблицы отлавливает клик по ячейке.

Из строки таблицы извлекается идентификатор транзакции.

Находится соответствующая транзакция в массиве и выводится подробное описание.

document.getElementById('transactionsTable').addEventListener('click', function(event) {

    if (event.target && event.target.nodeName === "TD") {

        const row = event.target.parentNode;

        const id = row.firstChild.textContent;

        const transaction = transactions.find(t => t.id === parseInt(id));

        document.getElementById('detailedDescription').innerText = `Подробное описание: ${transaction.description}`;

    }

});

**Вывод:**

Этот код позволяет управлять транзакциями, например, добавлять новые и удалять существующие. Он хранит информацию о каждой транзакции в объекте Transaction и использует массив transactions для хранения всех транзакций. Приложение обновляет таблицу транзакций на веб-странице и отображает общую сумму транзакций. Также при клике на строку таблицы вы можете увидеть подробное описание выбранной транзакции.

Если вам нужно управлять своими финансами или отслеживать траты, это приложение может быть полезным инструментом.

**Ссылка на репозиторий Git:**

https://github.com/mihailovp04/JavaScript/tree/main/LI2

**Ответы на контрольные вопросы:**

1.Каким образом можно получить доступ к элементу на веб-странице с помощью JavaScript?

*Для доступа к элементам на веб-странице с помощью JavaScript можно использовать методы getElementById, getElementsByClassName, getElementsByTagName, querySelector и querySelectorAll.*

2.Что такое делегирование событий и как оно используется для эффективного управления событиями на элементах DOM?

*Делегирование событий - это прием программирования, который позволяет управлять событиями на множестве элементов с помощью одного обработчика, установленного на их общем родителе*.

3.Как можно изменить содержимое элемента DOM с помощью JavaScript после его выборки?

*Для изменения содержимого элемента DOM после его выборки можно использовать свойства элемента, такие как textContent, innerHTML или value (для элементов ввода).*

4.Как можно добавить новый элемент в DOM дерево с помощью JavaScript?

*Для добавления нового элемента в DOM дерево с помощью JavaScript можно использовать методы createElement для создания нового элемента и appendChild или insertBefore для его добавления в дерево.*

**Список использованных источников**

1. Chat GPT
2. Современный учебник JavaScript
3. Руководство по JavaScript
4. YouTube