Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

**Дисциплина:** JavaScript

**Отчёт**

**по лабораторной работе №4**

**Тема:** Работа с DOM-деревом и событиями в JavaScript

Выполнил: студент группы IA2403

Демченко Юрий

Проверил: преподаватель

A.Murineț

Кишинев, 2025

**Лабораторная работа №4**

**Цель работы:**

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

**Условия работы:**

Шаг 1. Настройка и структурирование проекта

Шаг 2. Реализация функций для анализа транзакций

Шаг 3. Отображение транзакций

**Ход работы:**

1.Создание структуры проекта

Созданы файлы:

index.html - основная страница приложения

style.css - стили приложения

src/index.js - главный модуль приложения

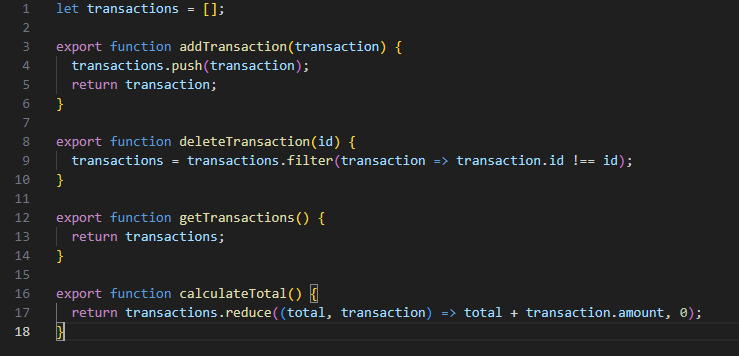
src/transactions.js - модуль работы с данными транзакций

src/ui.js - модуль работы с пользовательским интерфейсом

src/utils.js - вспомогательные функции

2.Реализация функционала

2.1. Модуль transactions.js



-Создан массив для хранения транзакций

-Реализованы функции для:

Добавления новых транзакций

Удаления транзакций по ID

Получения всех транзакций

Подсчета общей суммы

2.2. Модуль ui.js





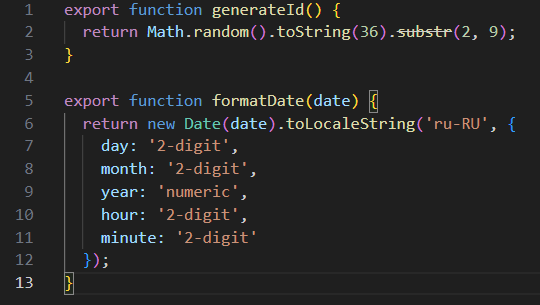
Реализованы функции для:

Отрисовки транзакций в таблице

Обработки формы добавления

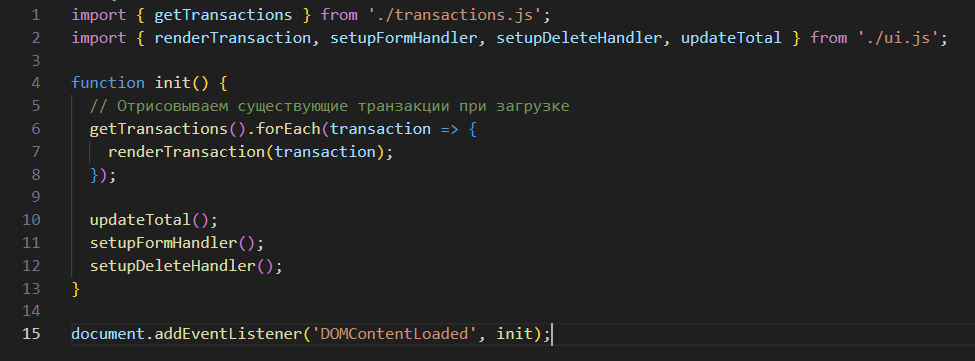
Удаления транзакций через делегирование событий

2.3. Модуль utils.js



Вспомогательные функции для генерации ID и форматирования даты

3.Инициализация приложения (index.js)



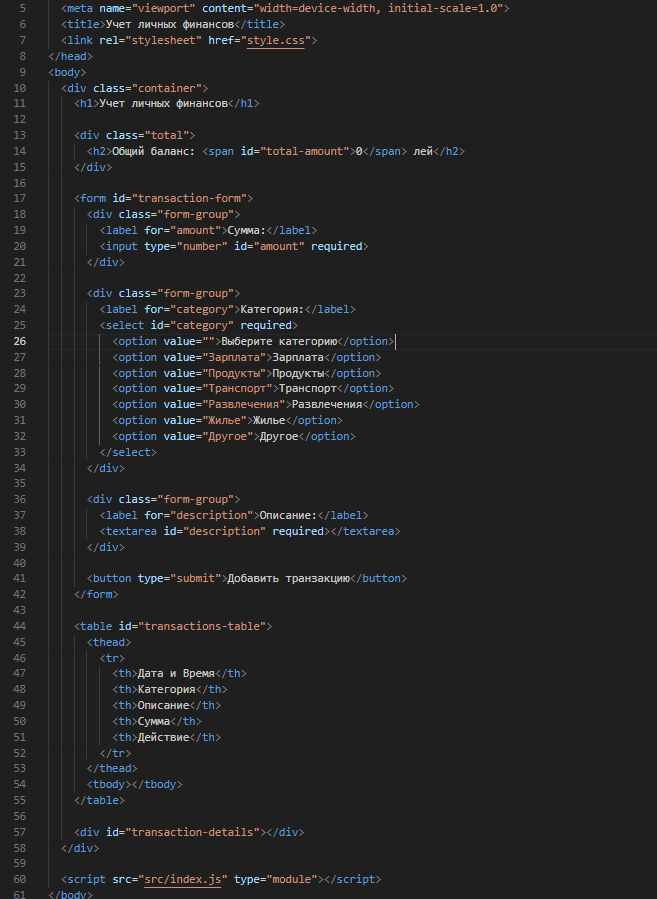
При загрузке страницы:

Отрисовываются существующие транзакции

Обновляется общая сумма

Настраиваются обработчики событий

4.Реализация интерфейса (index.html)

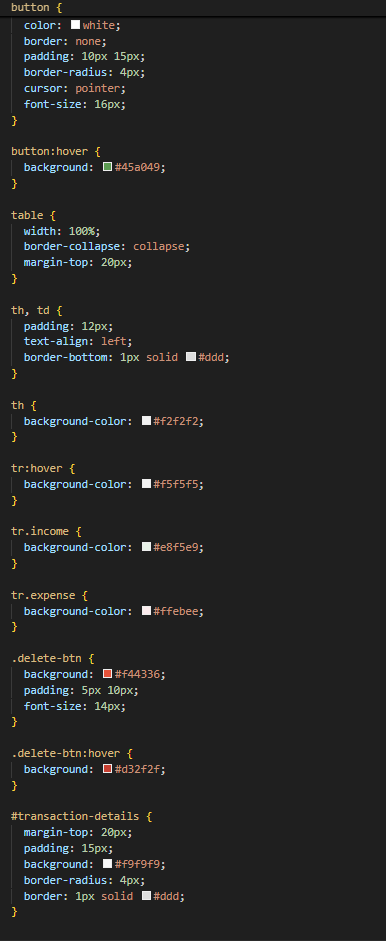


Создана форма для добавления транзакций

Реализована таблица для отображения транзакций

Добавлен блок для деталей транзакции

5.Стилизация (style.css)



Оформлены:

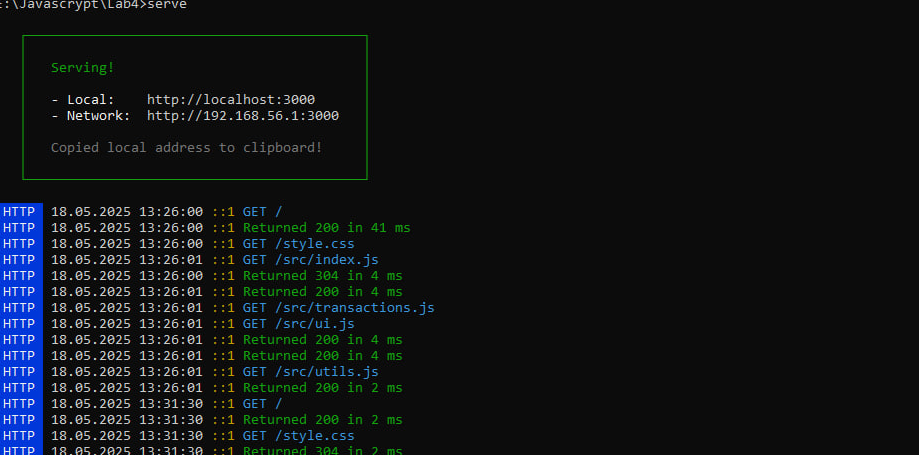
Форма добавления

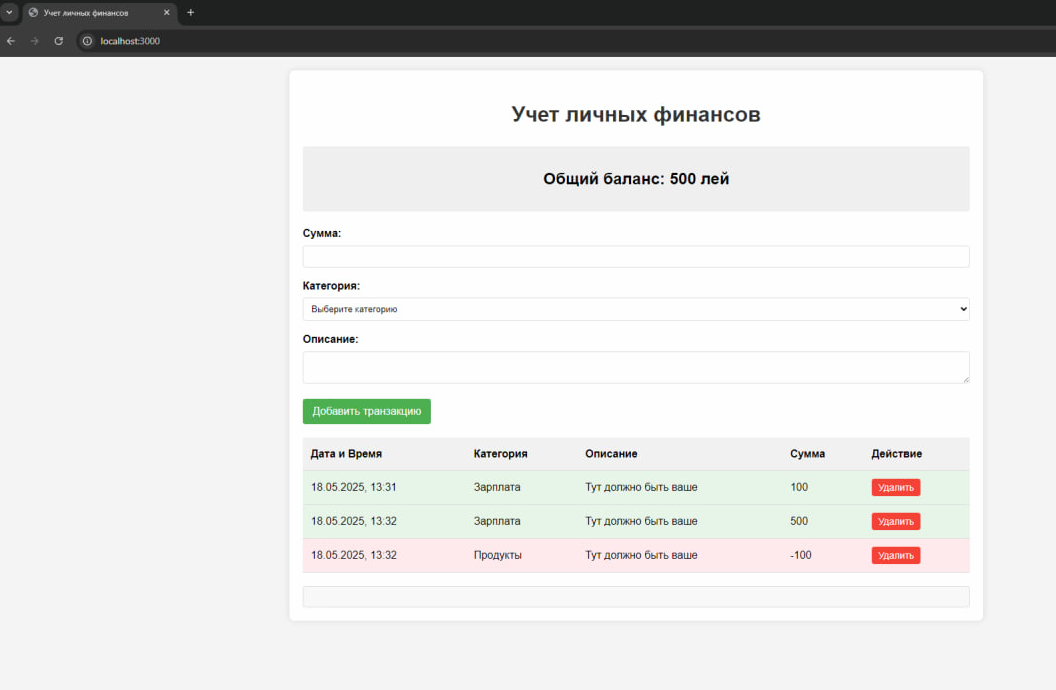
Таблица транзакций

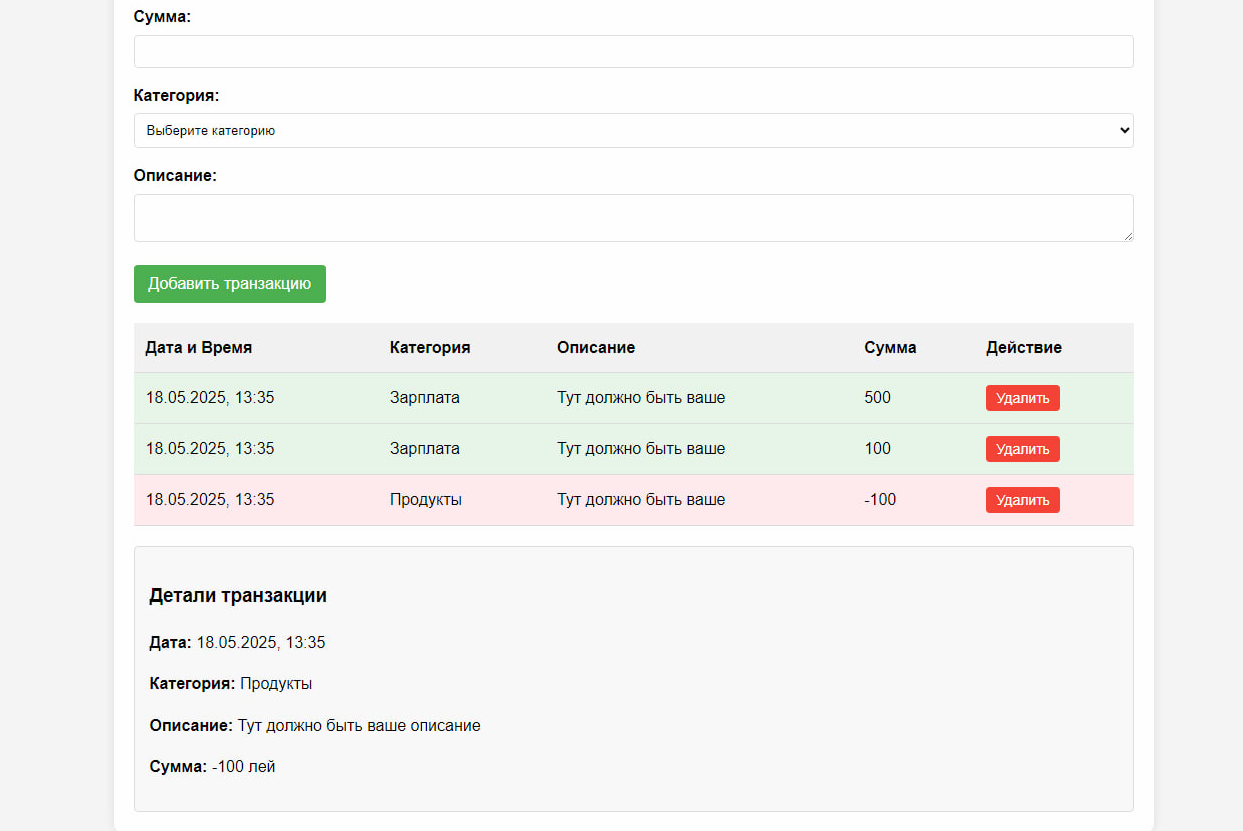
Элементы управления

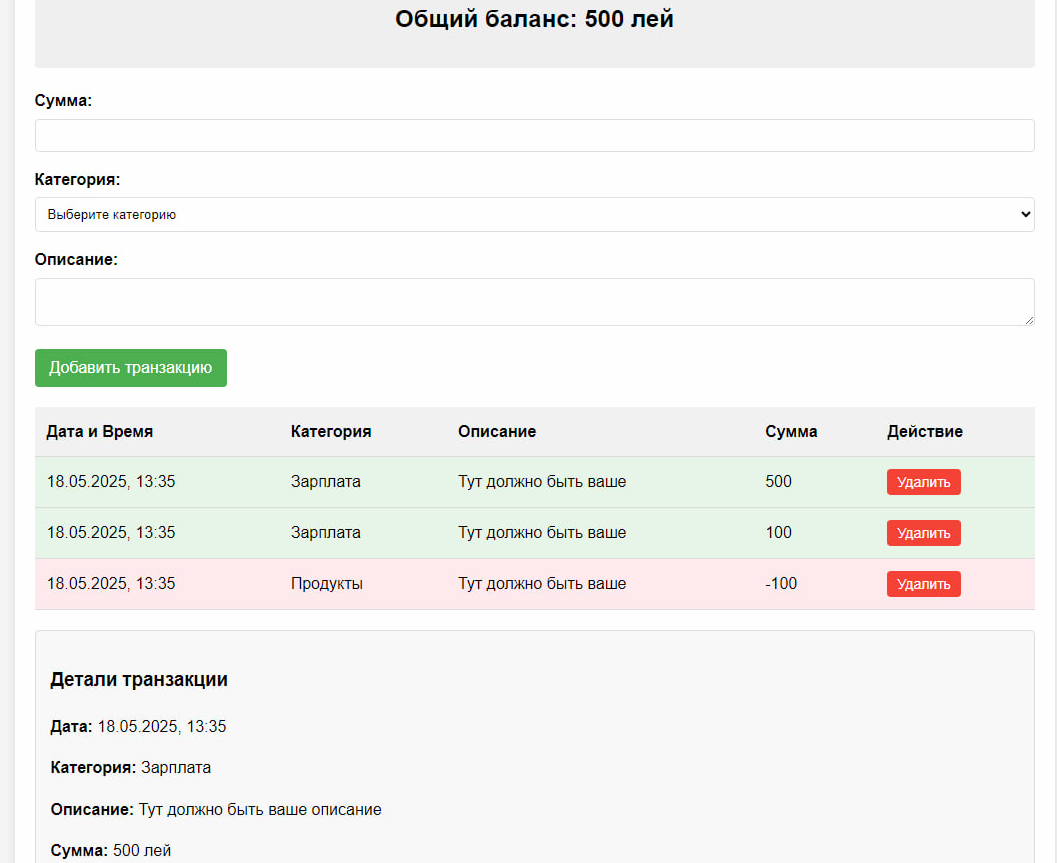
Реализована подсветки доходов/расходов

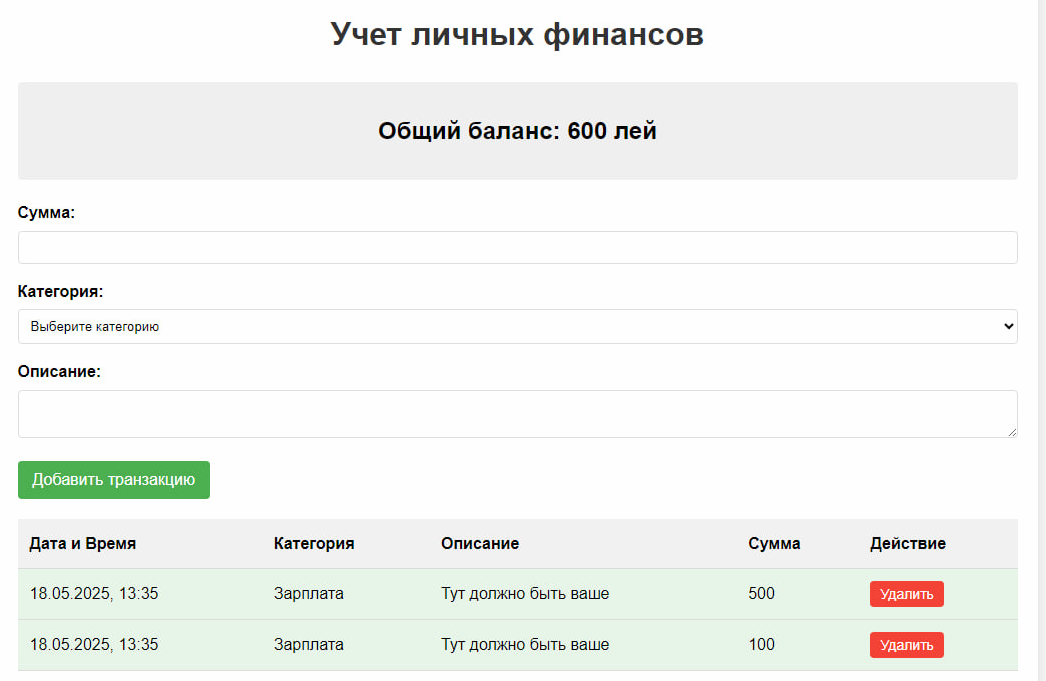
Результат при запуске:











**Выводы**

В ходе выполнения лабораторной работы №4 я изучил основы работы с DOM-деревом в JavaScript, методы выборки и изменения элементов страницы, обработку событий и делегирование событий и работу с формами и валидацию данных. Приобретенные навыки позволяют создавать интерактивные веб-приложения с динамическим обновлением контента.

**Ответы на контрольные вопросы**

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

document.getElementById(id)

document.querySelector(selector)

document.querySelectorAll(selector)

document.getElementsByClassName(className)

document.getElementsByTagName(tagName)

Пример:

const btn = document.getElementById('myButton');

const paragraphs = document.querySelectorAll('p');

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

Делегирование событий — это приём, при котором обработчик события вешается не на каждый элемент отдельно, а на их общего родителя. Событие "всплывает" вверх по дереву DOM, и родительский элемент обрабатывает его через event.target. Это эффективно, потому что не нужно вешать обработчики на каждый дочерний элемент и работает даже с элементами, созданными динамически после загрузки страницы.

Пример:

document.getElementById('parent').addEventListener('click', function(e) {

if (e.target.classList.contains('delete-btn')) {

console.log('Удалить элемент с id:', e.target.closest('tr').dataset.id);

}

});

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

element.innerText — текстовое содержимое (без HTML).

element.innerHTML — содержимое с HTML-тегами.

element.textContent — аналогично innerText, но чуть быстрее.

Примеры:

const p = document.getElementById('message');

p.innerText = 'Новый текст';

p.innerHTML = '<strong>Жирный текст</strong>';

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

// 1. Создание элемента

const newElement = document.createElement('div');

// 2. Настройка элемента

newElement.textContent = 'Новый элемент';

// 3. Добавление в DOM

document.body.appendChild(newElement);

**Использованные источники**

Документация MDN Web Docs - <https://developer.mozilla.org/ru/>

Современный учебник JavaScript - <https://learn.javascript.ru/>

Документация W3C по DOM - <https://www.w3.org/DOM/>

Курс MOODLE - <https://github.com/MSU-Courses/javascript>