МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Математики и Информатики

Департамент Информатики

Лабораторная работа № 3

Учет личных финансов с использованием JavaScript и DOM

Проверил: профессор, др. N.Nartea

Выполнил: Киорогло Никита IA2304

Кишинев, 2022

**[1.Теоритическая часть…………………………………………………2](#teo)**

**[2.Практическая часть…………………………………………………..2](#prak)**

**[3.Вывод и ссылка на GIT………………………………………………5](#vivod)**

**[4.Ответы на контрольные вопросы…………………………………..5](#kontr)**

**[5.Источники………………………………………………………………5](#isto)**

Теоретическая часть

Формулировка задачи:

Цель работы состоит в создании веб-приложения для учета личных финансов с использованием JavaScript для взаимодействия с DOM-деревом. Задачи включают в себя создание структуры проекта, реализацию функциональности добавления, удаления и отображения транзакций, а также подсчет общей суммы транзакций.

Описание цели и основные этапы работы:

Цель работы заключается в ознакомлении студентов с основами взаимодействия JavaScript с DOM-деревом на примере создания веб-приложения для учета личных финансов. Основные этапы работы включают в себя настройку и структурирование проекта, создание представления транзакций, добавление и удаление транзакций, подсчет общей суммы транзакций, отображение полного описания транзакции, добавление формы для добавления транзакции, а также документирование кода.

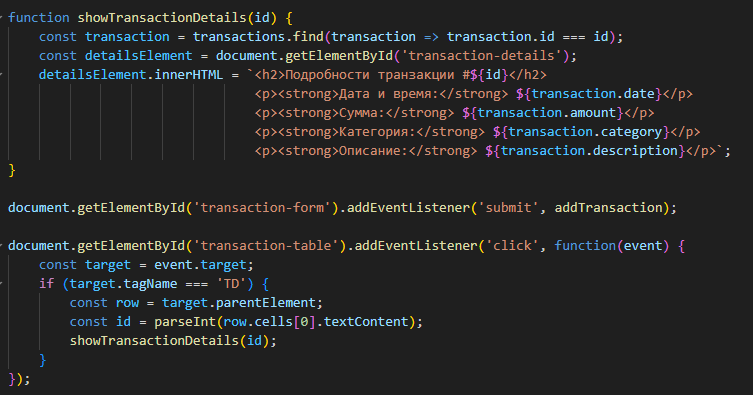
Практическая часть

Краткое описание особенностей реализации:

* Использование HTML, CSS и JavaScript для создания интерфейса и логики приложения.
* Динамическое добавление и удаление строк таблицы с транзакциями.
* Отображение общей суммы транзакций в реальном времени.
* Подробное описание транзакции отображается при клике на соответствующую строку таблицы.



В этом фрагменте кода показана функция **addTransaction**, которая добавляет новую транзакцию в таблицу при отправке формы. Она получает данные из формы, создает объект транзакции с уникальным ID, добавляет транзакцию в массив, создает строку в таблице с данными о транзакции и добавляет обработчик для кнопки удаления. Кроме того, цвет строки в таблице определяется в зависимости от суммы транзакции, и общая сумма пересчитывается с помощью функции **calculateTotal**. Таким образом, эта часть кода позволяет динамически добавлять новые транзакции и обновлять таблицу и общую сумму на странице.



Отображение подробной информации о транзакции:

При клике на строку таблицы с транзакцией, подробная информация о ней отображается в блоке на странице. Это делает интерфейс более информативным и удобным для пользователей.

Вывод и ссылка на репозиторий Git

В ходе выполнения лабораторной работы было создано веб-приложение для учета личных финансов, которое демонстрирует основы взаимодействия JavaScript с DOM-деревом. Репозиторий проекта доступен по ссылке: [https://github.com/ChiorgloNichita/test2]

Контрольные вопросы:

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

Доступ к элементам на веб-странице можно получить с помощью метода document.getElementById(), указав ID элемента в качестве аргумента.

Также можно использовать методы document.querySelector() и document.querySelectorAll() для выборки элементов с использованием селекторов CSS.

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

Делегирование событий - это метод, при котором обработчик события устанавливается на родительский элемент, который существует в момент загрузки страницы. Когда событие происходит на дочернем элементе, событие всплывает до родительского элемента, где и запускается обработчик. Это позволяет эффективно управлять событиями на элементах, особенно если у вас есть множество дочерних элементов с одним типом события.

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

После выборки элемента DOM можно изменить его содержимое с помощью свойства innerHTML, установив новое значение в виде строки HTML. Также можно изменить текстовое содержимое элемента, используя свойства textContent или innerText.

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

Новый элемент можно создать с помощью метода document.createElement(), указав имя тега создаваемого элемента.

Затем этот элемент можно добавить в DOM дерево с помощью методов appendChild() или insertBefore() для добавления в родительский элемент.

Источники

<https://docs.arangodb.com/3.12/develop/transactions/javascript-transactions/>

<https://lazysysadmin.com/js/index.html>