

Молдавский Государственный Университет
Факультет Математики и Информатики
Департамент Информатики

Отчет по индивидуальной работе №3 по курсу JavaScript & TypeScript

Проверил: Nartea Nichita
Ответ составил: студент 1-го курса
группы I2302 Slavov Constantin

Кишинев, 2024

Цель индивидуальной работы:

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

Теоретическая часть:

В данной работе было создано простое веб-приложение, которое позволяет пользователю следить за своими финансовыми операциями, добавлять, отображать, редактировать и удалять введенные записи.

Описание цели и основные этапы работы:

- Основной упор делается на учет транзакций, связанных с автомобилями.
- Инициализация массива транзакций: Создание пустого массива `transactions`, который будет содержать информацию о каждой транзакции.
- Добавление транзакции: Создание функции `addTransaction`, которая обрабатывает отправку формы, извлекает данные из формы и создает объект транзакции. Затем транзакция добавляется в массив `transactions`, а также добавляется в таблицу и пересчитывается общая сумма.
- Отображение транзакций: Создание функции `appendTransactionToTable`, которая добавляет переданную транзакцию в таблицу на странице. Также эта функция устанавливает класс строки таблицы в зависимости от значения суммы транзакции (положительная или отрицательная).
- Удаление транзакции: Создание функции `deleteTransaction`, которая удаляет транзакцию по ее ID из массива `transactions`, обновляет таблицу и пересчитывает общую сумму.
- Расчет общей суммы: Создание функции `calculateTotal`, которая рассчитывает и отображает общую сумму всех транзакций.
- Отображение подробностей о транзакции: Создание функции

showTransactionDetails, которая отображает подробное описание выбранной транзакции при клике на соответствующую строку таблицы.

- Добавление обработчика событий: Установка обработчика события для отправки формы, чтобы вызывать функцию addTransaction при отправке формы.

Краткое описание особенностей реализации:

Исходя из условия задания, я создал 3 файла, каждый из которых имеет свое назначение:

- HTML-файл: Используется для создания “скелета” страницы, таблиц, форм, полей для ввода данных, кнопок.

- CSS-файл: Используется для стилизации всех элементов страницы, для того, чтобы интерфейс данной программы был понятен для использования юзером и в целом имел приятное оформление.

- JS-файл: Используется для реализации логики программы, обновление содержимого страницы и обработку всех изменений, который ввел пользователь.

- Используются функции для добавления, удаления и отображения транзакций, что обеспечивает чистоту и модульность кода.

Пример использования проекта:

Финансовый учет автомобиля

01.05.2024 14:45

1300

Возврат средств от дилера

Поддержка дилера

Добавить транзакцию

ID	Дата и Время	Категория	Описание	Действие
1	2024-05-01T14:30	Maintenance	Сменили масла и фильтра	Удалить
2	2024-05-01T14:45	Fuel	Заправка автомобиля бензином	Удалить
3	2024-05-01T14:45	Sponsorship	Возврат средств от дилера	Удалить

Итого: -700.00

ID: 1

Дата: 2024-05-01T14:30

Сумма: 1500

Категория: Maintenance

Описание: Сменили масла и фильтра

Вывод:

В целом, разработанное веб-приложение представляет собой удобный инструмент для учета финансов, особенно связанных с автомобилями. Оно позволяет пользователям добавлять новые транзакции, просматривать уже существующие и удалять ненужные, а также отслеживать общую сумму всех транзакций. Использование HTML, CSS и JavaScript в сочетании позволило создать динамичное и интуитивно понятное приложение. HTML использовался для разметки структуры страницы, CSS - для стилизации элементов, а JavaScript - для реализации логики приложения. Каждый из этих языков программирования выполнял свою функцию, совместно создавая полноценное

веб-приложение. В результате получилось приложение, которое может быть использовано для ведения учета финансов автомобиля, обеспечивая простоту и удобство в использовании для пользователя.

Ссылка на репозиторий GitHub:

https://github.com/kraaddys/JS_and_TS/tree/main/LI3_JS

Ответы на контрольные вопросы:

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

- JavaScript позволяет получить доступ к элементам веб-страницы через Document Object Model (DOM), используя методы такие как getElementById или querySelector. Эти методы позволяют выбирать элементы по их идентификатору или CSS-селекторам соответственно.

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

- Делегирование событий — это метод управления событиями, при котором обработчики событий устанавливаются на родительские элементы вместо отдельных дочерних элементов. Это повышает производительность и упрощает управление динамическими элементами, так как не требуется устанавливать обработчики для каждого отдельного элемента.

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

- После выборки элемента с помощью JavaScript, его содержимое можно изменить с использованием свойств innerHTML или textContent. innerHTML позволяет изменять

HTML-структуру внутри элемента, в то время как `textContent` предназначен для работы с текстовым содержимым элемента.

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

- Для добавления нового элемента в DOM сначала используется метод `createElement` для создания элемента. Затем его можно настроить по необходимости и, наконец, добавить в DOM с помощью методов `appendChild` или `insertBefore`. Это позволяет вставлять элемент в указанное место в структуре DOM.