Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

**Отчет по индивидуальной работе №3 по курсу JavaScript & TypeScript**

Проверил: Nartea Nichita

Ответ составил: Pitropov Alexandr

Кишинев, 2024

Цель индивидуальной работы:

Целью данной работы было создание веб-приложения для учета транзакций игрового персонажа с использованием JavaScript для взаимодействия с DOM-деревом.

Теоретическая часть:

Было разработано веб-приложение, предназначенное для учета финансовых операций. Пользователь может добавлять, просматривать, редактировать и удалять записи о транзакциях.

Основные этапы работы:

Инициализация массива транзакций: Создание пустого массива transactions для хранения информации о каждой транзакции.

Добавление транзакции: Создание функции addTransaction, которая обрабатывает отправку формы, извлекает данные из формы и создает объект транзакции. Далее транзакция добавляется в массив transactions, а также отображается в таблице, где пересчитывается общая сумма.

Отображение транзакций:

Создание функции appendTransactionToTable для добавления транзакции в таблицу на странице. Каждой строке таблицы присваивается класс в зависимости от значения суммы транзакции (положительная или отрицательная).

Удаление транзакции: Создание функции deleteTransaction для удаления транзакции по ее ID из массива transactions, обновления таблицы и пересчета общей суммы.

Расчет общей суммы: Создание функции calculateTotal для расчета и отображения общей суммы всех транзакций.

Отображение подробностей о транзакции: Создание функции showTransactionDetails для отображения подробного описания выбранной транзакции при клике на соответствующую строку таблицы.

Добавление обработчика событий: Установка обработчика события для отправки формы с целью вызова функции addTransaction при отправке формы.

Особенности реализации:

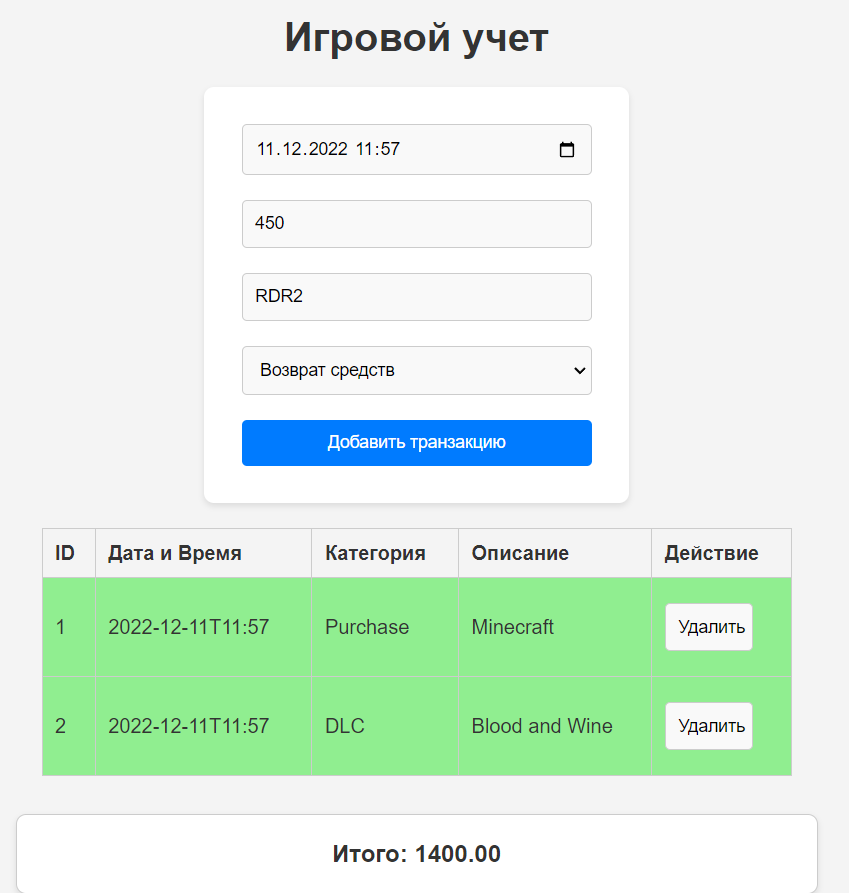
Для реализации функционала были созданы три файла:

HTML-файл: Используется для создания основной структуры страницы, форм, таблицы и блоков отображения информации.

CSS-файл: Применяется для стилизации элементов страницы, обеспечивая удобство использования приложения и приятный внешний вид.

JS-файл: Используется для программной логики приложения, обновления содержимого страницы и обработки событий, внесенных пользователем. Код разбит на функции, что обеспечивает его чистоту и модульность.

Пример использования проекта:



***Вывод:***

В целом, разработанное веб-приложение представляет собой удобный инструмент для учета финансов, связанных с гейм-контентом.. Использование HTML, CSS и JavaScript в сочетании позволило создать динамичное и интуитивно понятное приложение.

***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.

1. 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.
2. 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.
3. 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.

***Ссылка на репозиторий GitHub:***

https://github.com/alexunderpitropov/individualka