**Тестовое Задание для Фронтенд-Разработчика**

***Описание Задачи***

Вам необходимо реализовать функционал для динамического обновления суммарного баланса по всем счетам на веб-странице. Суммарный баланс отображается в элементе <p> с идентификатором id="overall-balance". Значение баланса обновляется при успешной отправке любой из форм создания, редактирования или удаления операций, влияющих на баланс счетов.

***Технические Детали***

*Backend-запрос:* для получения текущего значения суммарного баланса необходимо отправить запрос к {% url "overall\_balance\_get" %} (Django-аналог ендпоинта <http://localhost:8000/overall_balance_get/>). Backend вернёт актуальное значение баланса в JSON формате, например: {"overall\_balance": 124.0000000022701}.

*Frontend-реализация:* при успешной отправке формы (создание/редактирование/удаление операции), должен происходить запрос к указанному backend-сервису для получения обновленного значения баланса.

*Форматирование баланса:* Суммарный баланс должен быть отформатирован в человекочитаемом виде (например, число 134875.1314500022701 будет отформатировано в виде 134 875,13).

*Обработка Ошибок в Запросах:*

* Предусмотрите механизм обработки ошибок для сценариев, когда запрос к {% url "overall\_balance\_get" %} возвращает не валидное значение или не возвращает данных вообще.
* Реализуйте соответствующие пользовательские уведомления или логирование ошибок для обеспечения надежности интерфейса.
* Убедитесь, что при возникновении ошибки в запросе, пользовательский интерфейс остаётся функциональным и информирует пользователя о проблеме, не вызывая полного сбоя страницы.

***Примерный план выполнения задачи***

*Реализация запроса для обновления баланса:*

* Напишите функцию для отправки запроса к {% url "overall\_balance\_get" %} и получения актуального значения баланса.
* Убедитесь, что запрос отправляется после успешной отправки любой из форм.

*Динамическое обновление элемента с балансом:*

* Используйте полученные данные для обновления содержимого элемента с идентификатором overall-balance.
* Примените форматирование для корректного отображения суммы баланса.

*Проверка и тестирование:*

* Убедитесь, что запрос на обновление баланса отправляется и обрабатывается корректно.
* Проверьте, что суммарный баланс отображается в корректном формате и обновляется в реальном времени.

**Критерии Оценки**

*Функциональность:* Полное и корректное выполнение задачи.

*Техническая реализация:* Эффективность и качество написанного кода.

*Использование дополнительных библиотек:* Использование htmx или Alpine.js для улучшения решения будет оценено дополнительно.

*Код:* Чистота, читаемость и структурированность кода.

*Обработка ошибок:* Корректная обработка возможных ошибок при запросах к серверу и форматировании баланса.

**Время на реализацию:**

*На выполнение задания отводиться:*

* 48 часов

**Дополнения к Тестовому Заданию**

*Важно:* В данном задании мы хотим оценить ваши навыки работы непосредственно с чистым JavaScript и HTML. Вы должны воздержаться от использования сторонних библиотек и фреймворков, таких как React, Vue, Angular, jQuery и других. Все решения должны быть реализованы с использованием нативного JavaScript (ES6+) и стандартного HTML.

*Требования к Использованию Технологий:*

* *JavaScript*: Используйте современные стандарты JavaScript (ES6+). Допускается использование таких возможностей, как fetch для запросов к серверу, async/await для асинхронного кода и т.д.
* *HTML:* от вас ожидается чистая и структурированная разметка HTML. Убедитесь, что ваша разметка семантически корректна и доступна.