Тестовое задание

При выполнении использовать JavaScript, Vue.js, HTML, CSS

Задача: создать веб-страницу (имитация личного кабинета банка) и отобразить данные, получаемые от backend. Верстка адаптивная, дизайн на Ваше усмотрение.

Что должно быть на веб-странице:

- Необходимо отобразить список всех счетов (например, счет 1 и сумма).

- Должна быть кнопка для создания нового счета. После создания сразу показать новый счет в списке всех счетов.

- Создать кнопку напротив каждой строки со счетом для пополнения. При нажатии на которую, открывается форма для ввода суммы. После пополнения, данные реактивно обновляются в списке всех счетов. То есть баланс у выбранного счета должен поменяться.

- Сделать кнопку “Потратить”, при нажатии на которую открывается форма для ввода названия (покупаемой вещи или услуги) и суммы. После покупки баланс счета уменьшается и это отображается в списке всех счетов.

- Сделать кнопку удаления счета. Также напротив каждой строки со счетом.

- Реализовать возможность посмотреть транзакции по счету. То есть при нажатии на счет раскрывается список всех пополнений и расходов по данному счету.

- Кнопки пагинации. Если счетов больше 10, то отображать эти кнопки. Пагинация должна работать корректно.

Важно, чтобы все действия взаимодействовали с backend приложением.

Список методов api, которые понадобятся для взаимодействия с backend приложением, ниже:

- **/rest-auth/registration/** - необходимо открыть эту страницу и зарегистрироваться. После отправки запроса на регистрацию, Вы получите токен, который необходимо использовать для авторизации. Все нижеследующие методы api не работают без авторизации. Поэтому для успешного обращения к ним, необходимо в Headers отправлять токен зарегистрированного пользователя. **Важно**: саму логику авторизации и регистрации не надо реализовывать. Вам надо только получить токен единожды и использовать его для обращения к нижеследующим методам api.

- **/api/bank/account/** (GET запрос вернет список всех счетов пользователя, отправка пустого POST запроса создает новый счет. Не забудьте про token в Headers)
- **/api/bank/action/** (GET запрос вернет список всех пополнений у пользователя, POST запрос позволяет пополнить счет. В теле POST запроса должны присутствовать поля account и amount, формат JSON)
- **/api/bank/transaction/** (GET запрос вернет список всех покупок, POST запрос позволяет списать деньги со счета - в теле запроса должны присутствовать поля account, merchant, amount. Если денег на счете не хватает, то получим error)
- **/api/bank/account/{id}/** (DELETE с пустым body удаляет счет с указанным id)

Запуск backend приложения для обращения к методам api

- Для запуска backend приложения у Вас должны быть установлены **docker** и **docker-compose**.
- Скачайте приложение по адресу **https://github.com/andrepopoff/django-rest-framework-bank.git**
- Выполните команду **docker-compose up --build**
- В отдельном окне терминала запустите команду **make make_migr**
- Затем команду **make migr**
- Бекенд приложение запущено по адресу **localhost:8000**
- Теперь методы api Вам доступны и Вы можете написать тестовое задание :)

Успехов!