Cookie jar v3 project layout description/Project brief prompt:

I am creating the ui mvp for an app that acts like a virtual petty cash drawer and i need your help to create a project brief that I will give to vercel's v0 ui creation tool to help create the ui. i am calling it a cookie jar instead of a petty cash drawer. this version of the app needs to be as simple as possible. I have provided a rough overview of the app layout/structure. Please ask any clarifying questions you may have that are necessary to create the project brief. Do not ask any questions about the smart contract structure etc. Right now we are only concerned with creating a project brief that will be used to create a simple dummy project layout using V0. Do not make any assumptions. If you don't know something tell me you don't and ask clarifying questions.

Landing page

the landing page of the app has 3 buttons. 1 button allows users to search for cookie jars they have access to. the 2nd button allows them to create their own cookie jar and the 3rd button allows them to manage jars they have already created.

Jar search page

This page has a search field where a user can enter an evm compatiable address. When they click the search button it will search all whitelists to find jars that the address is present in. there will be another button that allows them to connect their wallet. When they have connected their wallet the app will use that provided address to search for that address in all jar whitelists.

Jar creation page

When jars are created the following information needs to be specified by the jar creator: Jar owner (needs to be an evm compatiable wallet address), jar title, cool down period (how long users must wait between fund withdrawals. Eg if the cooldown period is 60 min then funds can only be withdrawn every 60 min), cookie token contract address (this is the token that will fill the cookie jar that eligible users will be able to withdraw. It can be any ERC20 taken or the native token for whatever chain it is deployed on.), network selection(this will be to select the network the cookie jar is deployed on it will be a drop down menu that only allows one network to be selected) maximum withdrawal amount (this sets the maximum amount of funds a user can withdraw per claim period. The app will enable users to

withdraw a variable amount of funds up to the maximum amount. The formatting on this field may be a bit tricky because of how erc20 token contracts can have different token decimals. Ideally I'd like this field to autoformat for different token decimals based on the cookie token address that has been entered but this may be too complex for consideration for this mvp. If you know of any simple solutions for this issue please suggest them), whitelisted addresses (these are addresses that have the ability to withdraw funds from the jar. there should be built in validation to confirm each address entered is a valid evm compatible address. All addresses should be separated by a space or a comma or both. There should be an ability to upload addresses in bulk via a csv file), initial funding amount (this is the amount of tokens the creator will initially send to the newly created cookiejar. The amount can be zero.), donate to the fund the cookie jar project radio button (this will be auto selected by default), amount to donate (this can be either a % of the initial funding amount or it can be a fixed amount. It will default to 5% of the initial funding amount. It will use the cookie contract address as the default donation token but can also have the option for jar creators to choose a different token to donate). This page will also have a create jar button that mints the cookiejar, funds the jar (if there was an amount entered for the initial funding amount field and sends donation funds to support the project if that option was selected). After the button is hit, the user will be prompted to sign transactions using their connected wallet.

A bonus feature is confetti after the jar is successfully created but instead of traditional confetti it is cookies coming out of a cookie jar in the same way that the confetti comes out of the confetti cannon.

View eligible cookie jars page

This is where the user will see all jars that the provided evm address has access to. Each cookie jar will show up as a card that has all of the information provided in the jar creation page except it won't show: all the whitelisted addresses, initial funding amount, if the jar creator donated to the cookie jar project.

and it will additionally show the contract address for the deployed jar (when the user creates the jar a factory contract mints a new contract for each cookie jar), the current balance of the cookie jar, # of total claims made on the jar,

if the address entered doesn't appear on any cookie jar whitelists then we will display this message "looks like the provided address doesn't have access to any cookie jars :crying_face" it will then have 2 inputs. 1). Search for a new address (enables them to input

a new address to search for and pulls up the results after they input the address and hit the search button). 2). Create a cookie jar. This will bring them to the jar creation page.

Cookie jar detail page

This page will display all the relevant information for a selected jar. It will show the same information as listed on the cards of the find a cookie jar page but it will display it in a more expanded manner. It will also show the claim history for that jar.

It will also give users the ability to click on the contract address so they can open it in the relevant block explorer(which would be based on the network it is deployed on). If a user's connected wallet provides an evm address that is on the jar's whitelist, and the claim period is open (EG the cooldown period has passed) they will also have the option to withdraw a variable amount of cookies from the cookie jar. When the user is eligible to withdraw they will need to provide a text description for why they are withdrawing funds (claim reason). The claim reason needs to be at least 20 characters.

This page will also provide an option to add funds to the cookie jar. If a user wants to add funds there will be a flow for them to provide details for how much they want to fund the jar. The funds they add must be in the same token the jar already contains. The flow will provide them the option to donate funds to the cookiejar project when they add funds to an existing cookie jar. The donation option will give them the option to donate a percentage of the jar funding amount. They will have the option to opt out (eg a radio/select button that says "Donate funds to the cookie jar team to maintain this project"). The option to donate will be selected by default and will be set to 5%. They should have the ability to adjust this amount and opt out of the donation if they wish.

Manage all my cookie jars page

The first stage of this page has the same view as the search for a jar page but instead of checking to see if the provided address is on a whitelist it checks to see if it is a cookie jar owner. If the provided address is a cookie jar owner it brings up all the jars it is an owner of. This page shows the same cookie jar info in the same layout as the view eligible cookie jars page except this time when the user clicks on the jar it brings them to the manage my individual jars page

Manage my individual jars page

in order to access any of the features that adjust any aspect of a cookie jar (modifying the whitelist, adjusting the maximum withdrawal amount etc), except sending funds to it, the user needs to connect their wallet to the app and the address provided by the wallet needs to match the address of the jar owner. This page displays the same content as the cookie jar detail page but it has the following fields as modifiable: white list (the jar owner can modify the white list of which addresses have access to withdraw funds from the jar), maximum withdrawal amount, add funds, add owners, remove all funds from jar.

When I call something a page, it doesn't mean it needs to have its own page. Let's do whatever makes the most sense here/follow best practices. I am just using the page term to signify a different stage in the user flow.

Extra info:

The cookie jars have an access control mechanism that uses a white list of EVM wallet addresses to determine who is eligible to withdraw funds from them.

In the description make sure you incorporate cookie metaphors eg creating a jar would be like baking cookies.