Brief:

As you are applying for a senior role in EraMeta, we expect that you have an excellent understanding of Angular and can design small-scale applications, which are able to communicate with the backend.

Please view this video of our launchpad MVP:
https://twitter.com/EraMetaDAO/status/1513534454454697998?s=20&t=LG4I9V5i62Q3JRnSp2aI0Q

TASK 1: Arweave Cost Calculator:

Design:

As a user who wants to deploy my candy machine through EraMeta, I would like to know how much SOL to have in my wallet in order to cover all of the uploading costs. Storing image files permanently through Arweave is not free and this cost should be calculated for the user before they decide to upload their Solana NFTs through EraMeta. The uploading costs include Arweave storage cost calculated in USD per MB.

Task:

Implement a front-end app that calculates the Arweave permanent storage cost. First, the user is prompted to connect their Solana wallet which should redirect them to a separate page upon a successful connection. On this next page, users are asked to select or drop as many image files as they would like. Once the user generated files have been stored in memory, the user should be able to click a “calculate” button. Upon clicking “calculate”, the program should fetch relevant data from Arweave in order to calculate the permanent storage cost. Once the cost has been calculated, the program should present a pop-up modal that displays the total storage cost as well as whether or not the user has enough funds in his wallet.

Please make the UI as beautiful as you can, using your own design guidelines. Any animation or unique design is a bonus.

Implementation:

Front-end: Angular

Back-end: ExpressJs (if needed)

Storage: Firebase (if needed)

References:

Arweave - https://www.arweave.org/

Arweave calculator - https://arweavefees.com/

TASK 2: Draggable stepper:

Example: https://miro.medium.com/max/1112/1*hRIcb4Cq4QviANwGYywQKw.gif

Design:

As a user, I would like to order my layer groups after I upload all my images. The layer group order is important because this is the order in which my final NFT will be layered. (ie. in a pfp collection the background must come before the character body, and the character body must come before the character's eyes and so on). If by accident the user has ordered the body layers before the background layers, then they should be able to drag the background step to the top of the stepper meaning that the backgrounds will be the first layer group in the layer order.

Task:

Create a stepper where each step component contains a textfield input, a "save" button and a "add layers" button. Upon clicking the "save" button, the selected step label should be renamed to the text input. Upon clicking the "add layers" button, a new step should be appended to the stepper. The main objective of this exercise is making the stepper's steps *draggable* so that the steps can be re-ordered at the user's demand. Every time the user drags to re-order the stepper, the program must print each step's label in the new order of the stepper.

Please make the UI as beautiful as you can, using your own design guidelines. Any animation or unique design is a bonus.

Implementation:

Frontend: Angular

References:

Stepper: https://material.angular.io/components/stepper/overview