## Exercise Prompt

**Add a frontend interface for your smart contract**

## Exercise Concept

Since the last exercise, we've learned:

* How your users fit into our existing blockchain mental model. Specifically, we've learned about
* How the Javascript APIs we get from Web3.js or ethers.js and a cryptowallet like MetaMask connect with the Ethereum network and the ABI generated from the functions in our smart contract code
* How those Web 3 resources connect with the Web 2 resources, like React, to create a blended interface with the user

For this exercise, we'd like you to build the frontend interface for your final project. **While your final project *will* be graded, this specific exercise will not be graded.**

## Exercise Parts

Building a frontend will include building the basic web framework (however you'd like to do that). You'll have to ensure both your interface and the user can connect with the Ethereum network via MetaMask. Your user should be able to submit transactions to your contract. When that transaction is successful, your interface should update with the newly reflected state. [MetaMask docs will be helpful in understanding this workflow.](https://docs.metamask.io/guide/getting-started.html#basic-considerations) [Check out their tutorial here as well.](https://docs.metamask.io/guide/create-dapp.html#basic-action-part-1)

For this exercise, you will need the basic functions for the smart contract(s) underlying your project. You'll use these to construct a transaction on the client side.

**1. Pick a frontend interface you'd like to use for your project.** This will be something like React or the more straightforward HTML/CSS/JS stack.

**2. Integrate Web 3** Pick a Web 3 javascript library you'd like to use. We've gone over [Web3.js](https://web3js.readthedocs.io/en/v1.4.0/) and [ethers.js](https://docs.ethers.io/v5/).

**3.** Build the Javascript methods that allow your interface to **detect the presence of MetaMask,** **connect to the current account**, **display information from your smart contract**, **submit a transaction to your contract,** **monitor the state of the transaction**, and **update the frontend interface** if the transaction is successful or not.

***A note on style:*** *In this course, we emphasize* ***functionality*** *and* ***security*** *over style. This is not a course in interface design, it is a course in Ethereum and decentralized application development. We simply want you to demonstrate what you have learned throughout the course, even if it's very straightforward.*

Once you've pushed the frontend interface to your final project, please share the link to your repository with your study group or on Discord

The link you submit below should look like this:

[**https://github.com/YOUR\_GITHUB\_USERNAME\_HERE/education-dao-final-project**](https://github.com/YOUR_GITHUB_USERNAME_HERE/blockchain-developer-bootcamp-final-project)

**Happy Building!**