Coin Flip Project Plan - Josh Codrescu

Deliverables:

- Coin Flip website/game
- How-to blog
    - Summaries about the things I learn + all the resources I used to learn them
    - Examples
    - Walk throughs
    - Etc.

My Goals:

What I want to learn:

- Rust
- Solana smart contract dev (even if it's just simple stuff)
- Connecting a website to a blockchain
- Real time connections (real time multiplayer) (client-server using websockets)

What I want to practice/get better at

- Front end dev
- Web design (if I have time)
- Simpler back end web stuff
    - serving up static pages/images
    - Databases? (can't see a need right now but may realize one as I get further in)
    - Routing between pages

Needs to be done:

Frontend:

- Join screen
    - logo/site theme
    - Name field
    - Amount field
    - Wallet connect button
    - join/flip button
- Coin flip screen
    - Coin flip animation
    - You won/you lost alert
    - Play again
    - Wallet of the other person playing (to verify a real transaction)
    - logo/site theme
- Loading screen
    - Site will need to be able to connect two users so while that search is happening I will need a loading screen
    - logo/site theme

    - Loading animation
    - Cancel button

Backend (web2):

- Need the ability to connect two users into the same session
    - Not sure how to do this but will look more into it
    - Most likely something using websockets
    - Probably a packet/lib out there already
- Once a player chooses to join the flip, the money they stated needs to be withdrawn from their wallet to the ‘coin flip bank’ that will hold it until a winner is decided
- Once players have joined a random number needs to be generated to decide the winner this should trigger the money to be sent to the winner
- After a winner and loser are chosen the animations need to be displayed and a play again button needs to pop up that links the player back to the join in screen with the information filled out

Backend (web3):

- Need the ability to have one wallet send the specified amount to the other
    - Or - need a ‘coin flip bank’ wallet that both players will send money to and one will receive money from
- Need the ability for each player to connect their wallets
    - There are definitely libraries for this that have standardized it I will look more into it
- Need the ability to check the balance of both players to make sure they can cover the bet amount

Languages to know:

Frontend:

- JS
- Probably using React/Next JS
- Look into MUI styles to handle website theme (more of a ‘if I have time’ thing)

Backend - web2:

- NodeJS or Java - still deciding Node seems easier, but it would be cool to get a lil Java backend experience

Backend - web3:

- Rust
- Solana blockchain
- If I have extra time I will extend this to Eth using Solidity

Proposed Timeline:

Weeks 1-2:

- Build the front end - 3 pretty simple pages Will help me practice my React
- Complete/Work on ‘the book’ (The book is the documentation for Rust)

- Complete exercises from the book

Weeks 2-4:

- Figure out how the real time multiplayer connection works and implement it
    - Using a library of course
- Make the non web3 part of the website work
    - Two players can join in with their names and a winner is decided
- Complete/Work on the book
- Complete exercises from the book

Weeks 5-6:

- Look into the Solana blockchain/Rust dev on the sol network
- Create a simple contract that sends money from one specified wallet to another
    - Create a simple contract that pulls money out of one wallet and sends it to the 'coin flip bank'
    - Create a simple contract that sends money from the bank to a provided wallet (the winner)
    - Look into security for the second contracts (I'm afraid of exploitation around the 'provided' wallet)

Weeks 7-8:

- Add the 'connect your wallet' button that connects to the user's preferred wallet (will start with just Phantom)
- Add the code needed to interact with the smart contracts
- Add the balance check to make sure both players have the funds they are planning on committing to the coin flip

Weeks 9-10:

- Buffer time
- AWS upload
- If I actually do finish everything above in the specified time frame I will:
    - Work on the website's design and make it look better
    - Look into security
    - Make my writeup cleaner
    - Add an Eth Implementation (only if I finish the project REALLY early)

Finals Week:

- Clean up my writeup and get everything fully ready to submit
- Blog Upload