Course Project for Blockchain Beginner (due 28th Jul 2021)

Implement at least one of the ideas below to pass the beginner course. You need to submit:

- Github URL
- Netlify URL

Improving design of Dapp

Do you think the current design for the app is ugly? Me too! Try making the design better.
You may be interested to learn more about the following:

How to be awesome frontend developer:

- https://github.com/kamranahmedse/developer-roadmap/blob/master/img/frontend.png?year-2021-2
- https://frontendmasters.com/guides/learning-roadmap/

Tailwind CSS:

- https://tailwindcss.com/

Tailwind Component (free):

- https://tailwindcomponents.com/

Sending Robots from the Dapp

Don't you wish that your users never leave your application? How about letting them do more within your application. Try adding a function that allow owners of the Robots to send their robots to someone else within the application!

Support multiple Robots contracts

User has been growing and you are thinking about expanding businesses? How about we transform this little shopfront to an entire marketplace? Consider allowing users to interact with multiple contracts to buy not just Robots from you but also other NFTs from other NFT creators!

Handling errors & edge cases

Are you more concerned about polishing the user experience before expanding to more feature scope? Perhaps you've noticed that if you were to cancel the transaction midway, the modal never goes away? Perhaps you've noticed that when a transaction happen while you're on the page the owner isn't updated live? Perhaps you've noticed that the application crashes when the 6th robot is bought? Fix them!

More Robots

Right now you've got 5 robots for sale. Could you think of a way to quickly generate thousands of robots easily? Or even infinitely many robots, as long as people can afford to pay? Or even in a way that even you do not know how the robots look like until it's being created?

Options

- Convert the Pet Shop (Lab 5) into an e-commerce website (to accept ETH/token into transactions)
- Totally different implementations, with DAPP (web-based) and smart contract involving sending transactions and reading the state of the contracts

Botapp (extension of Lab 4 tutorial)

- Improved the app design & handling of some errors
- Installed elevations into Tailwind to improve visual impact of images
- Added hero patterns svg to spruce up background
- Moved mint button up and used hover with contrasting colour for better UX
- Enabled click outside of the modal to clear screen or interrupt transactions
- Adjusted settings to optimise look-and-feel for responsiveness on different devices
- Added truncate function to prevent long owner address from overflowing
- Solidity, Hardhat, React JS, Tailwind

https://github.com/markvelous/botdapp | https://botdapp.netlify.app/

Botdapp2 (Agnostic NFT Minter)

- Minting of any number of robots (or digital assets) using IPFS link, name and metadata provided by the minter
- Solidity, Hardhat, React JS, Tailwind

Rinkeby Etherscan Contract:
https://rinkeby.etherscan.io/address/0x9049a2C93E7cf4d5aaF2baA9e1f404a0d766de1F

https://github.com/markvelous/botdapp2 | https://botdapp2.netlify.app/

Botdapp3 (Unique Robohash Mass Minter)

- Generate and mint robohash NFTs on the fly
- Prevented ID repetition (bot name)
- Added dynamic url using https://robohash.org/{bot}?size=150x150
- Added truncation to prevent text overflow
- Reconfigured setup to deploy contract on Rinkeby (instead of Ganache) using hardwallet
- Solidity, React JS, Truffle, Ganache

Rinkeby Etherscan Contract:
https://rinkeby.etherscan.io/address/0xB7d67fa0B552105c3Bcc7e15374Ea26B67A3b5A6

https://github.com/markvelous/botdapp3 | https://markobots.netlify.app/

Mark Bark Park (extension of Lab 5 tutorial)

- Pet shop with payable function for the adoption fee of 0.1 ether
- Solidity, JS, HTML, Truffle, Ganache

https://github.com/markvelous/markbarkpark | Localhost:7545