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)

- Improve the design of Dapp & handling of some errors
- Install 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 striking colour differential for better UX
- Click outside the modal to clear the screen or interrupt the transaction
- Adjusted settings to optimise look-and-feel for responsiveness on both desktop and mobile, including truncating the owner address to prevent overflow (npm install react-truncate)
- Solidity, Hardhat, React JS, Tailwind

https://github.com/markvelous/botdapp

https://botdapp.netlify.app/

Botdapp2 (agnostic NFT minter)

- Minting of any number of robots (and other 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 (infinite robohash minter)

- Auto-generate robohash bots by factoring in dynamic url using https://robohash.org/{bot}?size=150x150
- Prevent repetition of unique ID, i.e. bot name and allow truncation of long names to prevent text overflow
- Generate and mint robot NFTs on the fly
- Reconfigure setup to deploy contract on Rinkeby 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/markbarkpar

Localhost:7545

Setting up Hardhat

Create a project folder	mkdir <name>
Initialise a new node	npm init -y
Install hardhat	yarn add hardhat npm install --save-dev @nomiclabs/hardhat-waffle ethereum-waffle chai @nomiclabs/hardhat-ethers ethers
Initialise new project	npx hardhat
Accept default settings in hh	
Open Visual Studio	code .
Testing	npx hardhat test
Setting admin permission (if nec)	Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
OR clone from Github	git clone https://github.com/yehjxraymond/hh-default.git npm i npx hardhat test code .

Running Hardhat

Running scripts	**npx hardhat run scripts/<name>.js**
Access wallet	**npm i dotenv –save**
Update hardhat.config.js	

```
require("@nomiclabs/hardhat-waffle");
require("dotenv").config();

task("accounts", "Prints the list of accounts", async () => {
const accounts = await ethers.getSigners();
for (const account of accounts) {
   console.log(account.address);
  }
});
module.exports = {
  solidity: "0.8.4",
  networks: {
    hardhat: {},
    rinkeby: {
      url: process.env.ENDPOINT_RINKEBY,
      accounts: [process.env.ACCOUNT_1],
    },
  },
};
```

Install OZ library	npm i @openzeppelin/contracts --save
Create .env	ENDPOINT_RINKEBY <Alchemy>, ACCOUNT_1 <wallet private key> Update .env in .gitignore
Deploy on Rinkeby Etherscan	**npx hardhat run scripts/sample-script.js --network rinkeby**

NFT – Writing ERC721 Contracts

Install contract library	npm i @openzeppelin/contracts –save
Create a .sol file	

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.4;
```

```
import "@openzeppelin/contracts/token/ERC721/presets/ERC721PresetMinterPauserAutoId.sol";

contract Robots is ERC721PresetMinterPauserAutoId {
   constructor(
      string memory name,
      string memory symbol,
      string memory baseTokenURI
   ) ERC721PresetMinterPauserAutoId(name, symbol, baseTokenURI) {}
}
```

Creating Dapp

Create React App

```
npx create-react-app garage
cd garage
yarn start
```

Set up free web hosting
Set up project repository in Github
Create new site in Netlify

Set up Tailwind CSS

```
npm install -D tailwindcss@latest postcss@latest autoprefixer@latest
<npm install -D tailwindcss@npm:@tailwindcss/postcss7-compat postcss@^7 autoprefixer@^9>
npm install -D @craco/craco
```

Replace package.json

```
{
        "start": "craco start",
        "build": "craco build",
        "test": "craco test",
}
```

Create craco.config.js

```
// craco.config.js
module.exports = {
  style: {
    postcss: {
      plugins: [
        require('tailwindcss'),
        require('autoprefixer'),
      ],
    },
  },
}
```

Generate tailwind config

```
npx tailwindcss-cli@latest init / npx tailwindcss init
```

Replace tailwind.config.js

```
module.exports = {
  purge: [],
  darkMode: false, // or 'media' or 'class'
  theme: {
    extend: {},
  },
  variants: {
    extend: {},
  },
  plugins: [],
}
```

Replace src/index.css

```
/* ./src/index.css */
@tailwind base;
@tailwind components;
```

```
@`tailwind utilities;
```

Run development server

```
yarn start
```

Loading data from smart contracts

Install packages

```
npm i ethers axios dotenv --save
```

Create .env (with ALCHEMY_KEY=MY_API_KEY_GOES_HERE)

Add .env to .gitignore

Create config.js to read .env

```
import { config as setup } from "dotenv";
setup();
export const config = {
  alchemyKey: process.env.ALCHEMY_KEY,
};
```

Committing progress in Github

Saving progress to Github

```
git init
git add .
git commit -am "<comments on changes made>"
git push
```

Change origin

```
git remote set-url origin <target>
```

```
git remote add origin https://github.com/markvelous/botdapp.git
git branch -M main
git push -u origin main
```