NFT Auction Contract Deployment

JS + + =

FOLLOW @BMSCIS 195

> Web3 contract deployment with Reach and JS

This is a repo to work with, upload and Auction NFTs smart contracts using the svelte framework and Reach. Web 3 NFT Market Place with **Reach AWS** and **Svelte**. ...*This was tested on a linux platform with vs-code.

Prerequisites

Please install or have installed the following:

- VsCode
- NodeJs
- Algo Sandbox

Installation

Installing Algorand Sandbox

This is a fast way to create and configure an Algorand development environment with Algod and Indexer.

Docker Compose *MUST* be installed. Instructions.

On a *Windows* machine, **Docker Desktop** comes with the necessary tools. Please see the Windows section in getting started for more details.

Warning: There are known issues with running the Auction on a *Windows* machine. A linux setup would be preferable.

Getting Started with Algorand Sandbox

Ubuntu and macOS

Make sure the docker daemon is running and docker-compose is installed.

Open a terminal and run:

```
git clone https://github.com/algorand/sandbox.git
```

In whatever local directory the sandbox should reside. Then:

```
cd sandbox
./sandbox up
```

Once the sandbox is running, you can get default accounts with:

```
cd sandbox
./sandbox goal account list
```

In order to create a new wallet with MyAlgoWallet, you need to get the `mnemonic` and `seed` from the `account list` output.:

```
cd sandbox
./sandbox goal account export --account-id <account-id>
```

Clone the repository

```
sudo git clone https://github.com/BMscis/reach-tutorial.git
```

...*Make sure you are on the **Nft-Websocket** branch.

Install dependancies

Install all npm dependancies:

```
sudo npm install
```

Start dev environment

1. Run localhost

```
Run npm run dev
```

2. Open the localhost link on the browser.

3. Sign up and wait for authentication code on your email.

> Make sure your password includes digits.

> Once logged in please reload the screen if the NFT's do not apper.

4. Connect to the wallet you created above.

5. Create NFT.

> Alternatively go to NFTea to see the web version.

STACK

NODEJS

Nodejs is a JavaScript runtime environment that executes JavaScript code outside of a browser.

ROLLUP

Rollup is a module bundler for JavaScript which compiles small pieces of code into something larger and more complex, such as a library or application. It uses the standardized ES module format.

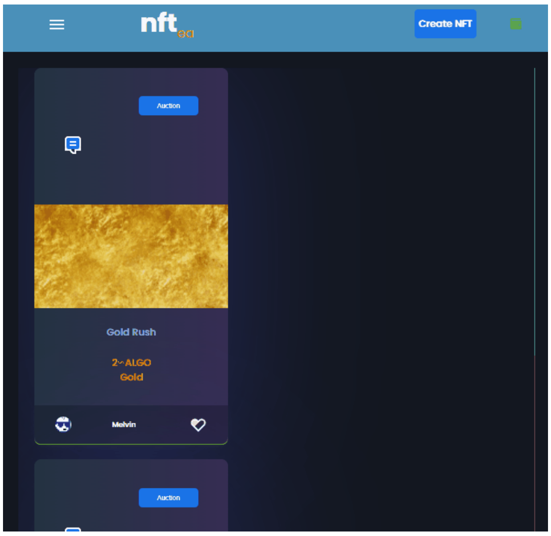
JAVASCRIPT

JavaScript is a dynamic programming language that's used for web development, in web applications, for game development, and lots more.

SVELTE

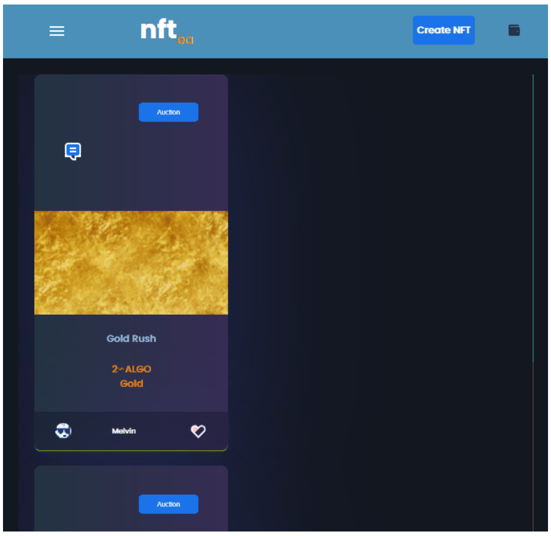
Svelte is a framework for building web applications just as React or VUE. We used svelte because with svelte we could develop components which functions as a html, has attributes,can be reused and are singularly responsive.

NFT COMPONENTS



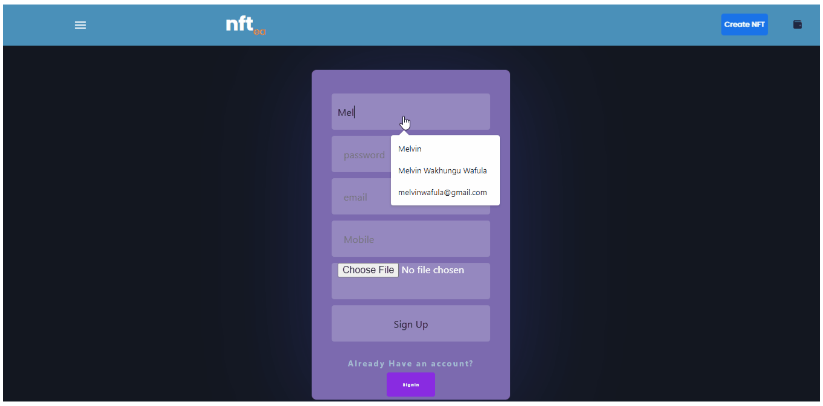
With this design, we were able to define one component and reuse it for all possible nfts. In this sense, the application is a Single Page Application. Meaning the page is separated into components that reload individually and don't require a full page reload.

SVELTE STORES



Svelte also allows writables called stores that can be used to share data between components and can be used to trigger a signle component update when the store value changes. Watch as the wallet value changes once we connect to the wallet.

AUTHENTICATION AND STORAGE



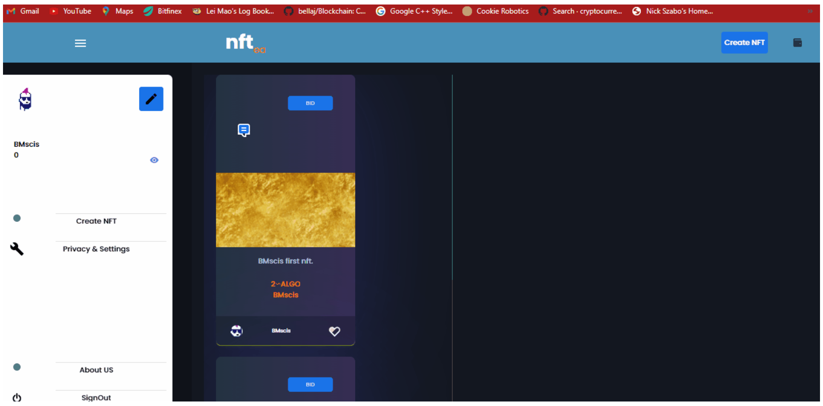
Storage and authentication are handled by AWS Amplify. NFTea uses Amplify to for backend queries. NFTea uses Amazon s3 to store images. NFTea uses Amazon Authenticate to authenticate users.

REACH

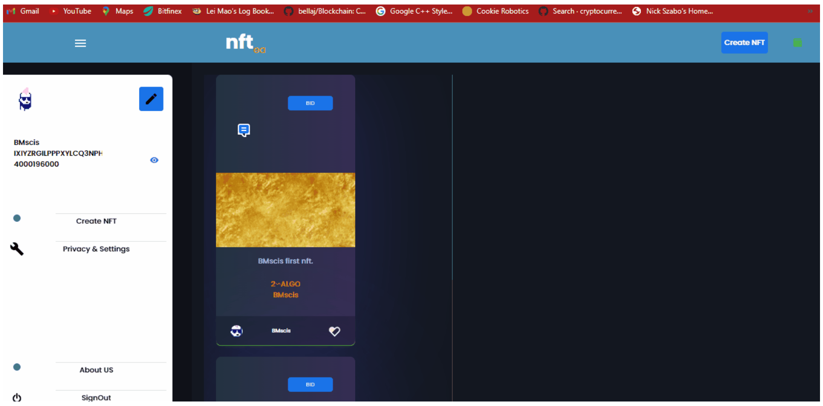
Reach is a blockchain development program that allows us to create and manage smart contracts on the blockchain. Reach compiles to a javascript module that can be imported into the project just like any other javascript module.

1. Reach smart contract code is written in a javascript like syntax and can be easily tested.
2. Reach allows the programmer to write single code in .sh and it will compile it to mjs.
3. Reach handles the backend deployment of the contract and therefore it can be used simultaneously with multiple blockchains.

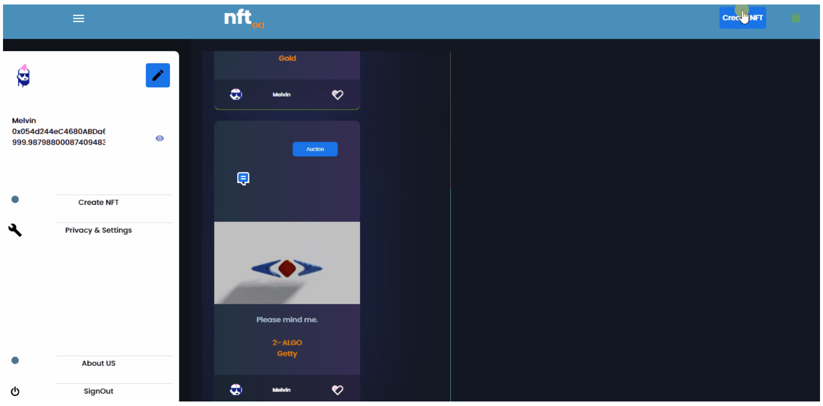
CONNECT WALLET OPTION



UPLOAD NFT WITH ALGORAND



UPLOAD NFT WITH ETHERIUM



Melvin
0x054d244eC4680ABDa6
999.98798800087409483