APS1050 Individual Project 2

Executive Summary

The project are about implementing new features and UI enhancements on top of an existing Pet's Shop DApp. For the second individual project, I am going to continously add two new features and make UI enhancements based on the first project.

New Feature: Pet's Market

I transfer the marketplace feature learning from Marketplace DApp to the Pet's Shop DApp, where users are allowed to trade pet in the page. First, each user is allowed to put any number of pet into market and each user is also allowed to buy any number of pet from the market as long as it has sufficient fund in their account. Users are not allowed to buy pets that registered by themself. Each pet is only allowed to be traded once, which means that a pet is not able to traded in between different accounts multiple times if it has been sold.

New Feature: Pet's Forum

I create this feature of forum from the scratch. The forum feature is a page that allow users to post pet related contents and reply to other user's post. Users are allowed to post any number of comments, and the post could be specified as a new post or a reply to a certain post. Time and owner of the post could be explicitly viewed from the page.

UI Enhancements

Similarly, I transform the original marketplace UI to fit the theme of the first project. The home page contains two more buttons to route to the two new features, and the page of marketplace and forum adapt the same theme as the election page.

Dependencies

- lite-server: ^2.3.0
- truffle: ^5.3.0
- chai: 4.2.0
- chai-as-promised: 7.1.1
- chai-bignumber: 3.0.0
- web3: 1.0.0-beta.55
- solidity: ^0.5.12
- Node.js: ^14.16.0
- Ganache: ^2.5.4
- OS: macOS ^11.2.3

Installation

Step 1. Install Dependencies

Install all packages specified in the `package.json`

```
npm install
```

Step 2. Install MetaMask

Install MetaMask

Step 3. Open Ganache

Open the Ganache, and click on `QUICKSTART` to obtain a new workspace for etherum

Step 4. Truffle Migrate and Deploy

Migrate and deploy the contracts to our local network

```
truffle migrate --reset
```

Step 5. Run

Run our application on localhost

```
npm run dev
```

Testing

There are also unit testing for each contracts. We are able to run the unit tests by truffle testing command:

```
truffle test
```

Contract test files are located under `test` folder, and testing result is presented by `p2_contract_testing.png`.

Demo

A video recording of project 2 demo `p2_demo.mov` has been placed inside the submission folder.