UNSW
THE UNIVERSITY OF NEW SOUTH WALES

Decentralised Puppy Shop
Final Report

Group: Snake Squad
Due Date: 25th May
Members:

Name:	Wenjie Gao	Yanjie Hu	Minzhen Ye	Yufei Zhang
ID:	z5045361	z5097732	z5042116	z5121128
Email:	qq471849587@gmail.com	kelseyan0305@gmail.com	minzhen.ye@student.unsw.edu.au	yufei.zhang@student.unsw.edu.au
Role:	Developer	Developer	Developer	Scrum Master

Abstraction

Decentralized applications(dapps)*, a new model for building massively scalable and profitable applications, is emerging. They are a type of software program designed to exist on the Internet in a way that is not controlled by any single entity and they are more flexible, transparent, distributed, resilient, and have a better incentivized structure than current software models.*

Introduction

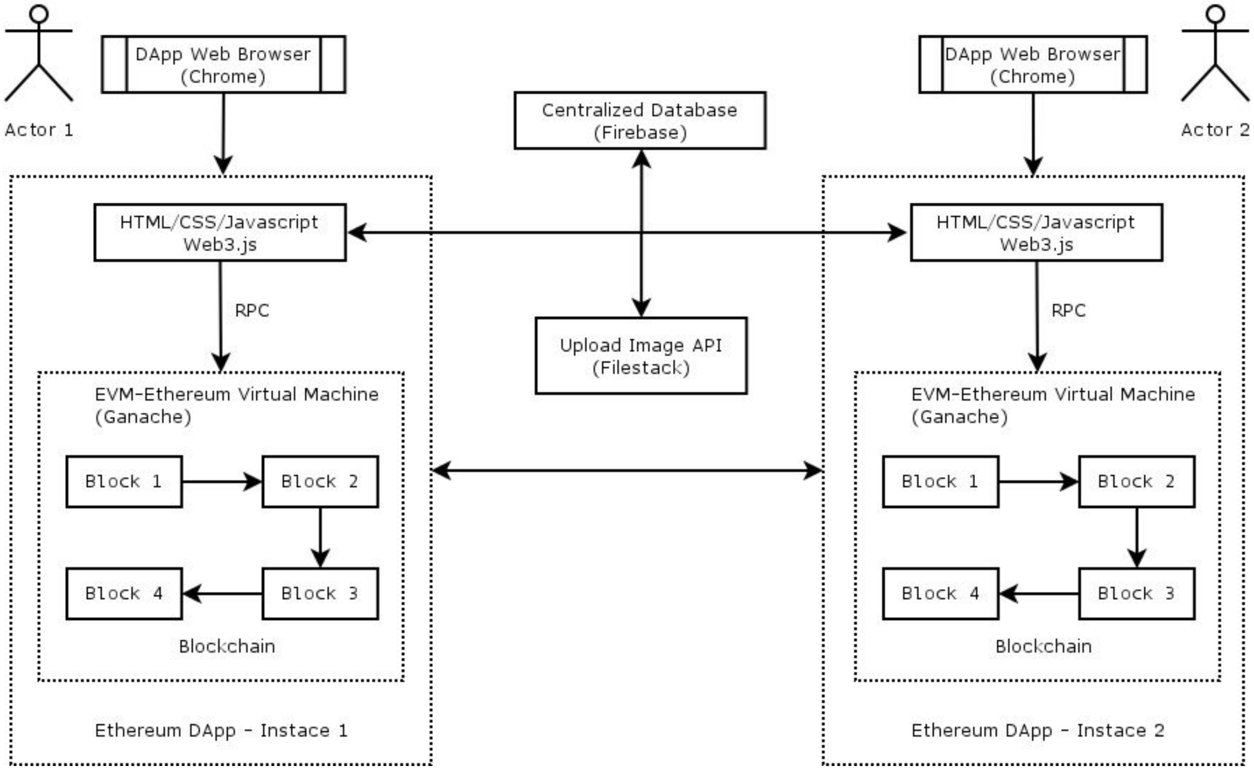
For this Project, our aim is to implement a decentralised shopping cart. Similar to a classical shopping cart application, the decentralised shopping cart supports operations such as adding items, removing items, calculating prices, searching items, purchasing etc. However, typical online shopping systems may contain extra charge from sellers. More than that, traditional online shopping applications suffer security issues such as DDoS, SQL injection and cross-site scripting attack. One solution is to make the application decentralised. In a decentralised shopping application, all trades happen directly between buyers and sellers with no middleman to take a cut from each sale. In terms of security, essential components have been distributed using the blockchain. Such property makes it practically impossible and expensive to attack.

Major features of Dapps:

- Be completely open-source and operate autonomously with no entity in charge of the majority of its currency;
- Have any protocol changes that are designed to make some overall improvement approved by all its users;
- Cryptographically store all of its operation data and records in a public blockchain;
- Use a bitcoin or a currency that is native to its blockchain system so that it can be accessed for use and any future contributions to its value from miners; and
- Generate tokens, or currency, which follows a standard cryptographic algorithm.

Overview

1. Architecture


Actor 1
DApp Web Browser (Chrome)
Centralized Database (Firebase)
DApp Web Browser (Chrome)
Actor 2
HTML/CSS/Javascript Web3.js
HTML/CSS/Javascript Web3.js
RPC
RPC
Upload Image API (Filestack)
EVM-Ethereum Virtual Machine (Ganache)
Block 1
Block 2
Block 3
Block 4
Blockchain
Ethereum DApp - Instace 1
EVM-Ethereum Virtual Machine (Ganache)
Block 1
Block 2
Block 3
Block 4
Blockchain
Ethereum DApp - Instace 2

The architecture is shown as the image. Our APP does not have a backend. We connect the centralized database (Firebase), Uploading Image API (Filestack) and Blockchain all through javascript. As we used truffle, it is recommended to use Ganache as the virtual machine

.

2. Functionalities

As a puppy shop, we supported the following functionalities:

a. Signup & Login & Forget Password (Reset password)
b. Update Personal Profile
c. Add a Puppy & Edit a Puppy
d. Display All Selling Puppies & Search for Puppies
e. Add Puppies to Cart & Delete Puppies from Cart
f. Calculate the Total Price of Puppies in Cart
g. Recharge
h. Checkout
i. Show Transactions & Show the Detail of One Transaction

Descriptions of the functionalities

1. **Signup (Firebase & Blockchain)**

When a user does not have an account, the first step for him/her is to sign up a new account. While signing up, users should fill in personal information, mailing address and security questions for resetting the password in case. For password, we choose md5 protocol to encrypt user's password to prevent others from looking user's password. We store the encrypted password instead of the original one.

2. **Login & Forget Password (Firebase)**

After signing up, users are allowed to login with a valid username and password pair.

System will check the existence of an entered username in firebase, and if it exists, system will then compare the entered password to the stored one. If match, users can successfully login; otherwise, system will through an error and alter it.

If users forget their password, they can use Forget Password to reset their passwords only if the input email addresses are stored in firebase and all of the corresponding security questions are answered correctly.

3. **Update Personal Profile (Firebase)**

After logging in, users are able to update their profiles. The current data will be load from firebase and users can edit it. While click 'Confirm', the data will be updated.

4. **Add a Puppy & Edit a Puppy (Blockchain & Filestack)**

After logging in, users are able to add puppies one by one to the blockchain as each puppy is unique. While adding a puppy, you need to upload an image for that puppy and fill in the Name, Breed, Age, Birth Place and Price for that puppy.

After adding a puppy, users are able to edit those puppies they added. Current data of the target puppy will be load from blockchain and users can do the same thing as it is while adding a puppy. Then the latest information of that puppy will be updated in blockchain.

5. **Display Puppies & Search for Puppies (Blockchain)**

After other users added some puppies, users are able to see all the puppies that are able to be bought on the home page. Users are not able to see those puppies that are added by themselves.

While searching puppies, users can use puppies' name, breed, age, birth place and price as a feature. While searching depends on price, users are allowed to use a comma to separate the lower bound and upper bound.

6. **Add Puppies to Cart and Delete Puppies from Cart (Only Javascript)**

Users can add a puppy to cart by clicking ‘Add’ button to the shopping cart. Users can see how many puppies are currently in the shopping cart. We will use cookies to store the information of puppies which are in the cart.

Delete a puppy from cart, users are allowed to delete puppies from cart if there exist at least one puppy in the cart.

7. **Calculate the Total Price of Puppies in Cart (Only Javascript)**

Once there is a change in the shopping cart, we are going to recalculate the total price that are needed. While calculating the price, we added 5% tax and round it to integer because solidity can only handle with int. In addition, users can see how many seller who sell puppies to them. The number of sellers equals to the number of transactions users will get. After checking out.

8. **Recharge (Blockchain)**

Once users click on ‘Checkout’ system will check the current balance first. If the balance is lower than they are asked, system will ask them to recharge. While recharge, users need to add how much money they want to recharge and click on Confirm. Important notes: as the blockchain need sometime to process, users need to refresh the webpage to make sure your balance is updated.

9. **Checkout (Blockchain)**

System will retrieve data from webpage including seller ID, buyer ID, trading puppies’ IDs, price, trading date, and mailing address. System will cost money from buyer and recharge balance for seller. Besides, system will generate a transaction ID for each seller, and charge the corresponding price for corresponding puppies. Each transaction will be recorded, as well as the trading date.

10. **Show Transactions & Show the Detail of One Transaction (Firebase & Blockchain)**

For each transaction, users can read simple information which tells the user about sell/buyer, trading date and price. Users can click on ‘Details’ to check the detail of that specific transaction. Detail including transaction ID, detail of each puppy, detail of cost, and mailing address. The mailing address does not stored on blockchain.

Implementation challenges

1. **The use of Firebase**

Firebase is a platform which allows us to build web and mobile applications without server side programming language. We do not need to write APIs. Firebase uses Google's server; acted as an API and a database. We chose the realtime firebase database as our centralized database and we only need to use javascript to connect firebase to our APP. However the difficulty is that firebase is a brand new thing for us, we need to learn its structures, its grammar to let it works.

2. **The use of Solidity**

Solidity is also a brand new thing for us during the developmentation. The grammar of it is similar to java. It is, however, weaker than java; for example, solidity does not support float. Hence when we need to store some data; for instance, price, we have to round to an integer.

In addition, when we use javascript and web3.js to call the solidity function, we need to convert for loop to a recursion because solidity requires a function-then style to get or set data as a loop. This problem also occurs when we use firebase. If we do not use 'then', there is not enough time for firebase or solidity to process the mission.

3. **The use of MetaMask**

This is the most annoying stuff that we need to use. Because there are so many strange bugs that can cause the whole project stack even though all of the codes are hundred percent right. Bugs such as "block number in MetaMask and testrpc does not match.", "GAS is beyond limit" and "GWEI is too low" are so annoying. Sometimes, if GWEI equals to 1, the program is good, but it need such a long time to process. The most unobservable bug is that, MetaMask cannot read "https://127.0.0.1:7545" while we copy it from Ganeche, we need to type it to fix that bug.

In general the project is not that hard because a pile of thing are redundant. Once we learned to code the first webpage from html to blockchain, we can use it as a template. There are a pile of work, and some of the tasks have a pile of steps (checkout); however, the structure is exactly the same as those simple pages.

User documentation

As prerequisites, you need install or apply all of the following stuff. You can go to Installations for for more details.

1. Install Node.js

As we decided to use web3.js, we are going to use npm to install web3.js, which need Node.js. Here we give you guys three installation guides on *Ubuntu*, *Mac* and *Windows*.

- Install on Ubuntu

  Install Node.js

  ```
  sudo apt-get install nodejs
  ```

  Install npm

  ```
  sudo apt-get install npm
  ```

- Install on Mac

  Install Node.js

  ```
  brew install node
  ```

  Install npm

  ```
  brew install npm
  ```

- Install on Windows

  you can find npm and Node.js download instructions from here.

- Check installations

  After finish the previous steps, you can check the version of node.js and npm by

  ```
  $ node --version
  v9.8.0
  $ sudo npm --version
  5.6.0
  ```

2. Install Truffle and web3.js

To install truffle, you need to type the following code in your terminal

```
npm install -g truffle
```

Then you can clone our code using

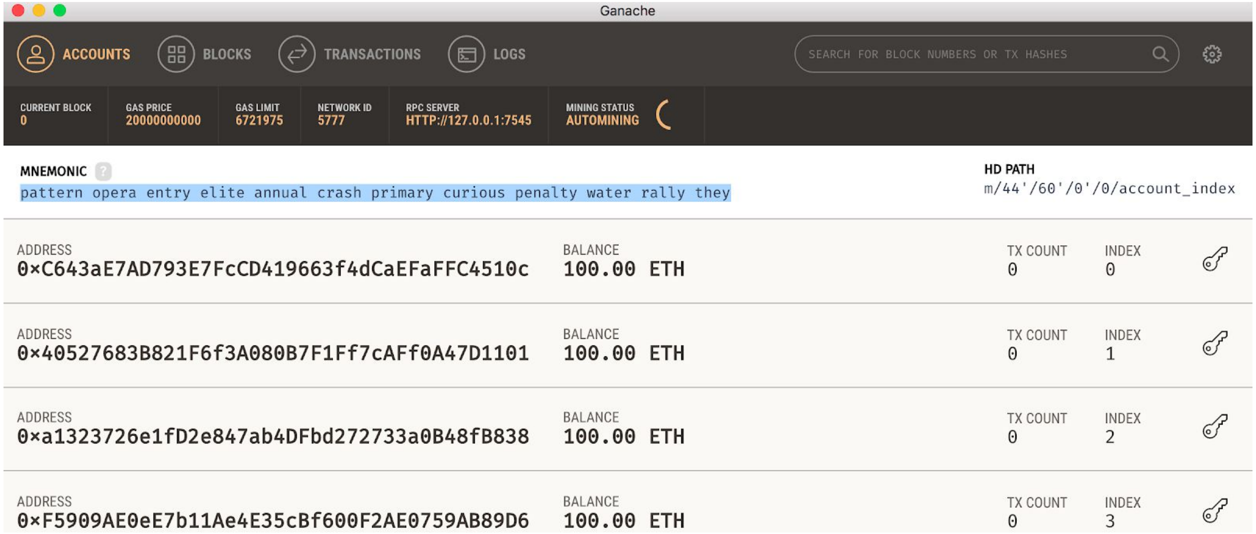
```
git clone https://github.com/Snake-Squad/Decentralised-Shopping-Cart.git
```

Once you cloned our code, you no not need to install web3.js because we already have it in our repo.

3. Download and Install Ganache

We used Ganache, a personal blockchain for Ethereum development you can use to deploy contracts, develop applications, and run tests. You can use the super link to download it.

Once you launch it, you will see



CURRENT BLOCK	GAS PRICE	GAS LIMIT	NETWORK ID	RPC SERVER	MINING STATUS
0	20000000000	6721975	5777	HTTP://127.0.0.1:7545	AUTOMINING

MNEMONIC
pattern opera entry elite annual crash primary curious penalty water rally they

HD PATH
m/44'/60'/0'/0/account_index

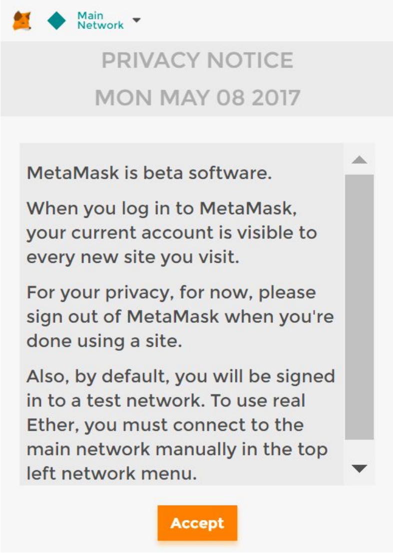
ADDRESS	BALANCE	TX COUNT	INDEX
0×C643aE7AD793E7FcCD419663f4dCaEFaFFC4510c	100.00 ETH	0	0
0×40527683B821F6f3A080B7F1Ff7cAFf0A47D1101	100.00 ETH	0	1
0×a1323726e1fD2e847ab4DFbd272733a0B48fB838	100.00 ETH	0	2
0×F5909AE0eE7b11Ae4E35cBf600F2AE0759AB89D6	100.00 ETH	0	3

4. Install MetaMask

The easiest way to interact with our dapp in a browser is through MetaMask, a browser extension for Chrome.

1.Install MetaMask in your browser.

2.Once installed, you'll see the MetaMask fox icon next to your address bar. Click the icon and you'll see this screen; click Accept to accept the Privacy Notice.



PRIVACY NOTICE

MON MAY 08 2017

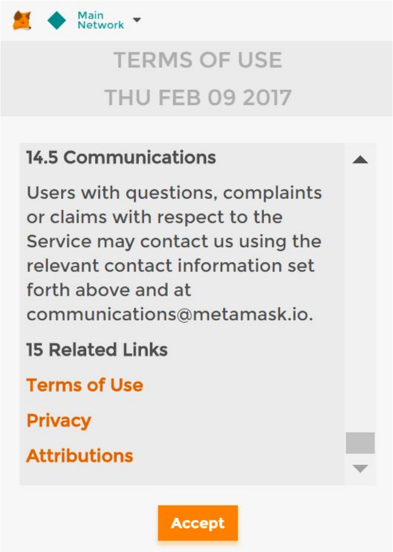
MetaMask is beta software.

When you log in to MetaMask, your current account is visible to every new site you visit.

For your privacy, for now, please sign out of MetaMask when you're done using a site.

Also, by default, you will be signed in to a test network. To use real Ether, you must connect to the main network manually in the top left network menu.

3. Then you'll see the Terms of Use. Read them, scrolling to the bottom, and then click Accept there too.



TERMS OF USE

THU FEB 09 2017

14.5 Communications

Users with questions, complaints or claims with respect to the Service may contact us using the relevant contact information set forth above and at communications@metamask.io.

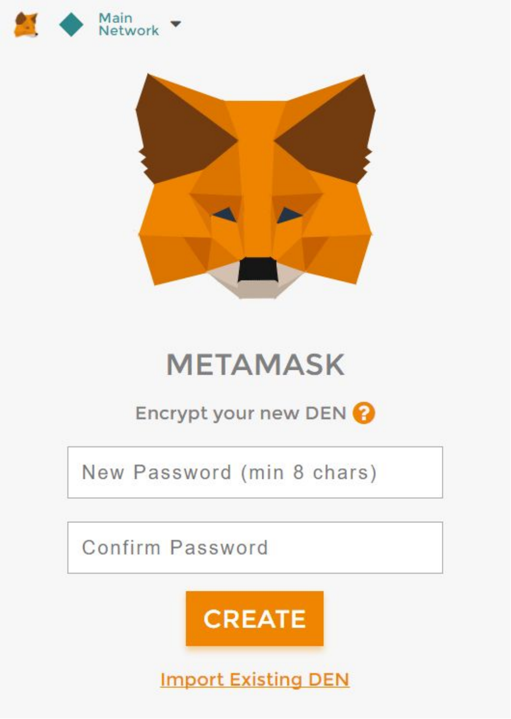
15 Related Links

Terms of Use

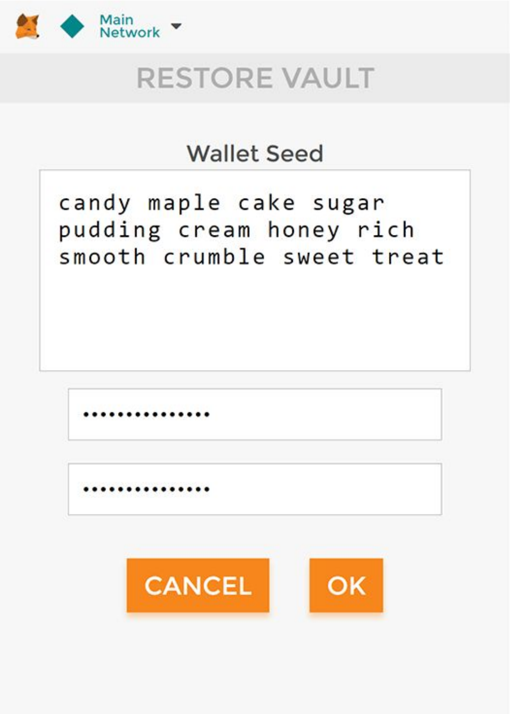
Privacy

Attributions

4. Now you'll see the initial MetaMask screen. Click **Import Existing DEN.**



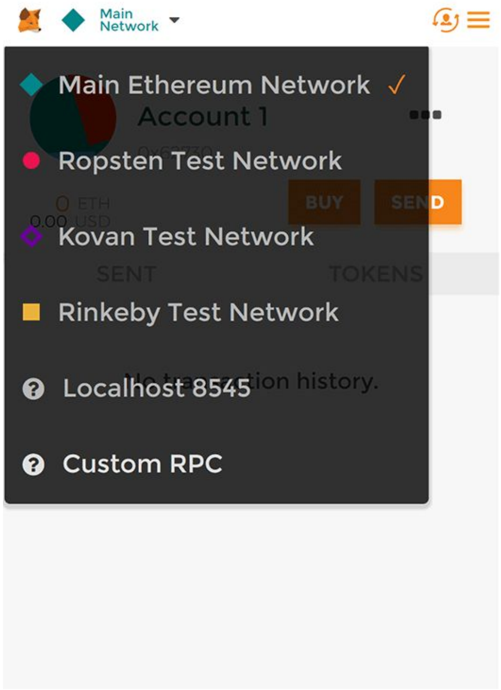
5. In the box marked **Wallet Seed**, enter the mnemonic that is displayed in Ganache.



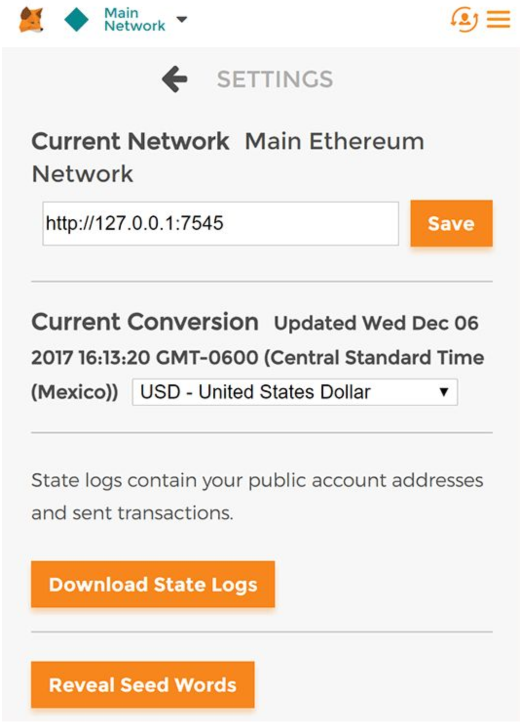
candy maple cake sugar pudding cream honey rich smooth crumble sweet treat

Enter a password below that and click **OK**.

6 Now we need to connect MetaMask to the blockchain created by Ganache. Click the menu that shows "Main Network" and select **Custom RPC**.



7 In the box titled "New RPC URL" enter http://127.0.0.1:7545 and click **Save**.



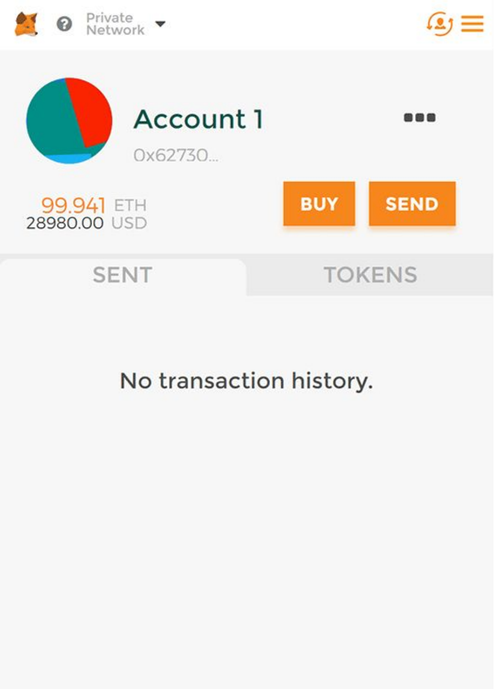
Current Network Main Ethereum Network

http://127.0.0.1:7545

Current Conversion Updated Wed Dec 06 2017 16:13:20 GMT-0600 (Central Standard Time (Mexico)) USD - United States Dollar

State logs contain your public account addresses and sent transactions.

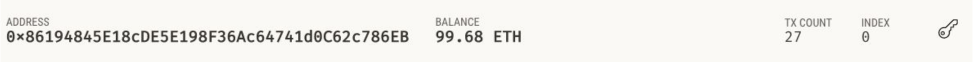
8 Click the left-pointing arrow next to "Settings" to close out of the page and return to the Accounts page.



Account 1
0x62730...
99.941 ETH
28980.00 USD

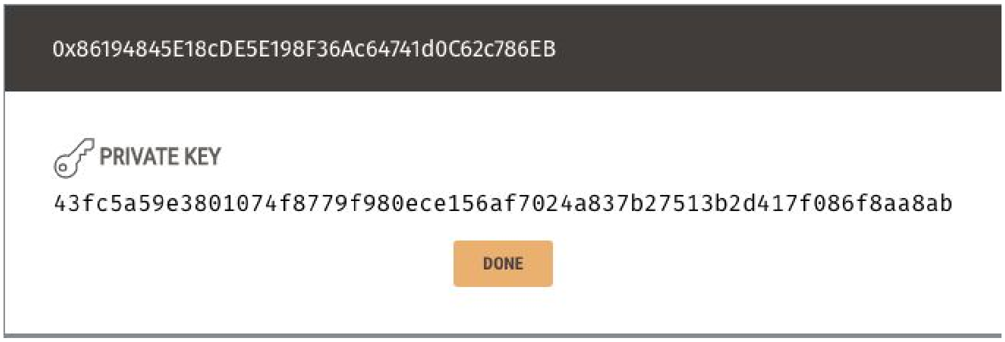
No transaction history.

9 To import an account, you need to get the private key of an account in Ganache



```
ADDRESS                                      BALANCE       TX COUNT   INDEX
0x86194845E18cDE5E198F36Ac64741d0C62c786EB   99.68 ETH     27         0
```

Click on the button that looks like a key an you will get

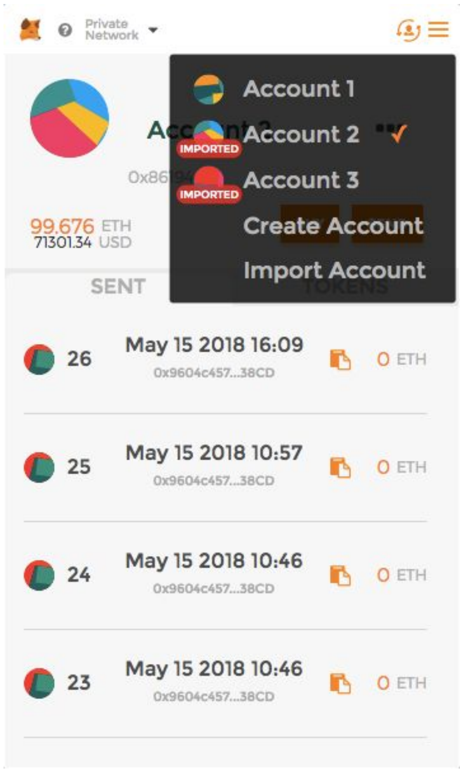


0x86194845E18cDE5E198F36Ac64741d0C62c786EB

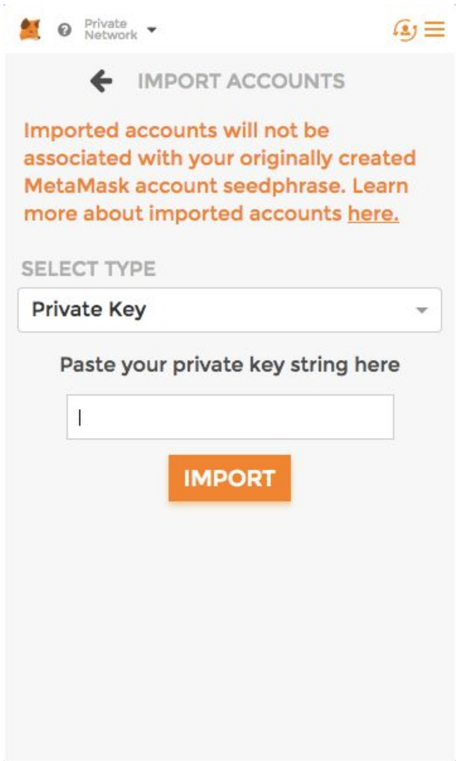
PRIVATE KEY

43fc5a59e3801074f8779f980ece156af7024a837b27513b2d417f086f8aa8ab

Copy the Private Key and click in MetaMask and click on Import Account



Paste the Private key and click Import



Imported accounts will not be associated with your originally created MetaMask account seedphrase. Learn more about imported accounts here.

SELECT TYPE

Private Key

Paste your private key string here

5. Apply a Filebase Account

If you are a new user, you can use https://firebase.google.com/ to apply a new account. However, if you cloned our code, you do not need to apply it.

6. Apply a Filestack Account

If you are a new user, you can use https://www.filestack.com/ to apply a new account. However, if you cloned our code, you do not need to apply it.

7. Start Using the APP

cd to the PetShop folder in your terminal, and type

```
npm run dev
```

You can combine with How to Run the Project if you cloned the code.

References

- pet_shop
- Developing Ethereum Smart Contracts for Beginners
- Installation Details
- Project Contribution

1.1. The useful features of firebase :

- **Real-time database:** Real-time data is the way of the future. Most databases require you to make HTTP calls to get and sync your data. Most databases give you data only when you ask for it. When you connect your app to Firebase, you’re not connecting through normal HTTP. You’re connecting through a Web Socket. Web Sockets are much, much faster than HTTP.

- **File Storage:** Firebase Storage provides a simple way to save binary files — most often images, but it could be anything — to Google Cloud Storage directly from the client. Firebase Storage has its own system of security rules to protect your GCloud bucket from the masses, while granting detailed write privileges to your authenticated clients.

- **Authentication:** Firebase auth has a built in email/password authentication system. It also supports OAuth2 for Google, Facebook, Twitter and GitHub. We’ll focus on email/password authentication for the most part. Firebase’s OAuth2 system is well-documented and mostly copy/paste.

- **Hosting:** Firebase includes an easy-to-use hosting service for all of your static files. It serves them from a global CDN with HTTP/2.

- **Full featured App Platform:** The Firebase team has integrated a bunch of new and existing Google products with Firebase. A bunch of these features apply to iOS and Android but not to web.

Filestack is an API which can upload the image from the local computer and store it in the cloud database. The database will return a URL for users to access the image conveniently. It can be used in different language such as python, php and JavaScript. In this project, because the function of uploading the image should run in the website, we added API in the JavaScript file. Before getting start with Filestack, it is necessary to sign up in the Filestack and there is 500 free pictures for users to use. You can choose plan in the Filestack to improve your storage. After create a new application and the API Key will be generated. This key is very important for us to access the API.

In order to integrate Filestack API into our Add Puppy website and Edit Puppy website, the first thing to do is include the Filestack Javascript library. We copy and paste this script into our Add Puppy and Edit Puppy HTML file. The code is shown in the Figure 1.

```
<script src="https://static.filestackapi.com/v3/filestack.js"></script>
```

Figure 1 Paste this in the HTML

And then in the JavaScript File, the API key will be used as argument for filestack.init(API_KEY). After initial function, the Javascript file can access Filestack. Sometimes the uploading images cannot satisfy the requirement of users or our projection, Filestack provides a pick function to edit uploading image.

In the pick function, I selected the image resources at first and the users can upload image from their computer or cloud storage such as Google dive, Facebook and so on. And then, because the size of some images' file is too large, which may increase the pressure for the system, I choose two ways to decrease the quality of the image. First method is to rule the maximum size of the image. If the size of the image is beyond the rule, the image will be resized to the maximum allowed size. Second way is to decrease the quality of image by decreasing the resolution ratio. Because the uploaded image will be shown in small size, it is not big influence when the resolution ratio of image is decreased. Finally, I add function to clip, circle and rotate images.

When users want to add image in the web site, they will click the add puppy button at first and then they will see the interface shown in the Figure 2. They can drag or click button in the middle of the interface to upload the image. They can choose files from Facebook, Instagram, Google Dive as well. After that, it will jump to the edit interface(Figure 3). In the interface, the users can crop, circle and rotate their image, which is shown in the Figure 3. Clicking save and upload button to upload the image, the users finish their uploading procedure.

For our website, we will store the URL of image with the puppy information and then store the URL in the blockchain. When the website want to show the puppy's information, the puppy's image will be shown by using the URL of the images.

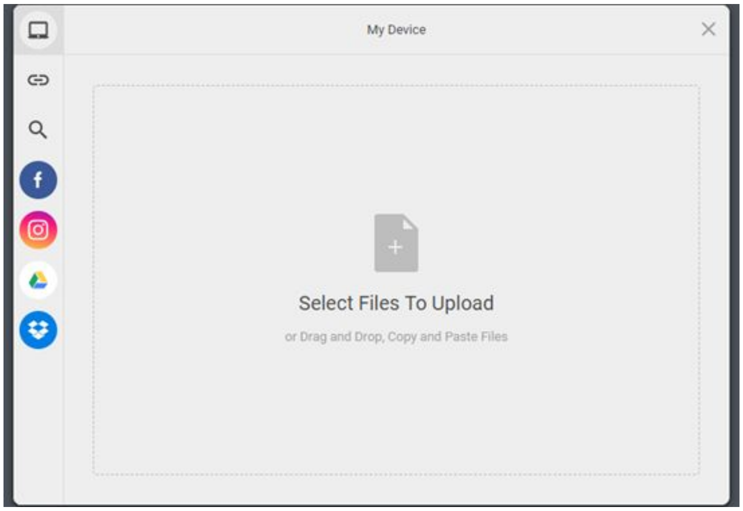


Figure 2 Selecting file interface

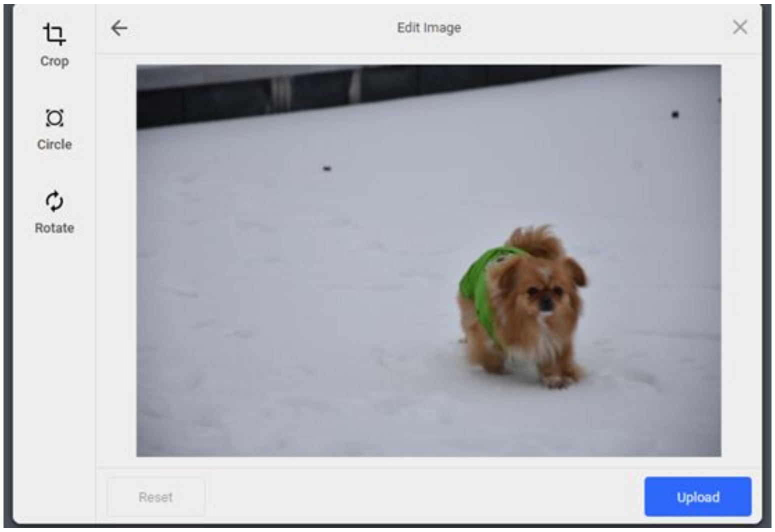


Figure 3 Editing interface