**Executive Summary**

**Mengyang Liu**

**Ziyun Weng**

**Wenqin Cheng**

**Ling Wang**

To implement various functions, the smart contract(Adoption.sol) is updated with functions to handle transactions and emit events. The HTML file adds buttons, while the appjs to process payments via MetaMask and update blockchain statuses. Deployment scripts are modified to redeploy the updated contract. These changes ensure smooth integration and seamless user functionality.

No.2: Voting

Function: A way of voting for the best pets (or liking a pet) in the Petshop DApp

transferred from Elections.

* File app.js:

Binding event（click） to vote button and retrieving the vote counts for all candidates from the smart contract and updates the corresponding UI elements (.pet-votes) for each candidate in the webpage.

* File index.html

Adding Vote button after adopt button

* File Adoption.sol:

Mapping addresses to boolean and adding “vote” and “getVotes” functions ensures each address can vote only once for a candidate numbered 0 to 15 and allows querying the vote counts for all candidates.

No.5 and 6: Filter

Function: A way of filtering for a list of pets (available pets not adopted already or already adopted pets) of a specific breed, age and location.

Modification:

* File app.js:
* Adding new features “populateFilters”, “filterPets” and “resetFilter” and they are used to form the filter condition, performing filtering and reset data respectively.
* Adding a few lines in “bindEvents” to keep the clicking-on acting well for filter and reset.
* File index.html
* Adding a new feature “<!-- Filter Form -->” under “<container>” to decorate the front so that the function in app.js can be used practically.

No.10: Track users’ adoption history

Function: A way of keeping track of a user's adoption history is added. In this function, users can view their own adoption history by clicking a button “View My Adoption History”. The pop up window shows the ID (0, 1, 2, 3, etc.) of the pet that they have adopted. This window can be closed by clicking “OK”.

* File app.js:
* Adding new features “handleViewHistory” which is used to record the pet adoption history.
* Adding a line in “bindEvents” to keep the clicking-on acting well for the buttons “View Adoption History”.
* File index.html
* Adding a new feature “View Adoption History” under petTemplate and make sure that this button appears after the adoption is completed.
* File Adoption.sol
* Adding the feature “Retrieving the adoption history of a specific pet”.

No.11: Track pets’ adoption history

Function: A way of keeping track of a pet's adoption history is added. Under each picture of the pet, there is a button “View Adoption History”. After adopting a pet, users can click the button under this pet, and the adoptor’s Metamask account address will be shown below this button. Click the “View Adoption History” button again can make the adoption history disappear.

* File app.js:
* Adding new features “handleUserViewHistory” which is used to record the user’s adoption history.
* Adding a line in “bindEvents” to keep the clicking-on acting well for the buttons “View User Adoption History”.
* File index.html
* Adding a new feature <!-- Button for Viewing User’s Adoption History”
* File Adoption.sol
* Adding the feature “Retrieving adoption history for a specific user”.

No.15: Vaccination Registry

Function: a way of keeping a vaccination registry for the pet. Before vaccination: the button says "Register". After entering vaccine details and completing registration: the button changes to "Show". Clicking "Show" displays the stored vaccination details in read-only mode. “Cancel Vaccination” button to revert vaccination status.

Modification:

* File app.js:
* Adding new features “openVaccinationModal”, “handleVaccinationRegistration” and “markVaccinatedLocally” for the purpose of triggering the vaccine form, storing the vaccine information locally and button logic manipulation.
* Adding a few lines in “bindEvents” to keep the clicking-on acting well for the key of “register”, “Compete” and “Cancel”.
* File index.html
* Adding new feature “<!-- Vaccination Modal -->” to decorate the front website so that the function in app.js can be used practically.
* File Adoption.sol
* Adding three features “registerVaccination”, “cancelVaccination” and “getVaccinationStatus”. Make sure to compile it before accessing the website.

No. 17: Buy Product & Book Service

Another critical functionality is to buy product and book service features through Pet Shop. Users can purchase pet-related products or book services through dedicated "Buy" and "Book" buttons. Payments are securely processed via MetaMask and recorded on the blockchain for transparency. Upon successful transactions, users receive real-time confirmations.

No. 21: Donation

The donation function lets users send ETH to the dog shelter via MetaMask. The smart contract processes donations, tracks totals. Users click "Donate", enter an amount, and approve the transaction. Upon success, the platform confirms the donation with a thank-you message.

In conclusion, petshop DApp combines blockchain technology with various app functions to provide transparency and security for each transaction.