# Summary

Originating from the foundational "pet-shop-box-master-completed" project that primarily allowed users to adopt pets, we've embarked on a mission to make pet adoption and management more interactive, transparent, and feature-rich using blockchain.

Here's a snapshot of our enhancements:

1. **Pet Filtering**:
   * At **unadopt.html (**modified from index.html), users can now narrow down their pet choices by filtering available pets based on their breed, age, or location. It load all pets by triggering function getPets form **app.js**.
2. **Pet Deletion**:
   * The **delete.html** page (modified from index.html) is introduced, where users can remove pets by simply entering their names. This action incurs a small fee. Implementation required changes across the front end, the handleDelete function from **app.js**, and the deletePet functon from **adoption.sol**.
3. **Pet Return**:
   * With the **adopt.html** page (modified from index.html), users who've previously adopted a pet can make a decision to return them. Just like pet deletion, this feature also attracts a fee. This involved modifications in the front end, the handleReturn functon from **app.js**, and the readopt function from **adoption.sol**.
4. **Pet Registration**:
   * Users can introduce a new pet into the system through **register.html (**modified from index.html). Essential details like an online image URL for the pet need to be provided. Registering a new pet is a paid feature, necessitating changes across the front end, the handleRegister function from **app.js**, and the register function from **adoption.sol**.
5. **Pet voting:**

* Users can vote for their favorite pets (Note: one user can only vote a pet once) At **unadopt.html (**modified from index.html), users can click “vote” button. This feature is paid and required alterations in the front end, the handleVote function from **app.js**, and the voteForPet function from **adoption.sol**.

1. **Website Statistics**:

* Offering a deeper insight into the adoption trends, users can now view statistical data regarding the distribution of adopted pet breeds and ages and vote results (please adopt after voting to see the result, because it only displays result of all adopted pets), ensuring a more informed adoption decision. It retrieves all adopted pets through the getPets function in **app.js** and filters them by front end.

1. **Pet Detail Update**:

* Our system now accommodates modifications to pet details. At **update.html (**modified from index.html), users can input a pet’s name, breed, age, location, and image URL for any updates. This feature is paid and required alterations in the front end, the handleUpdate function from **app.js**, and the updatePetDetails function from **adoption.sol**.

In conclusion, our enhancements to the original pet shop platform not only amplify its utility but also bolster transparency and user engagement by leveraging blockchain technology. As DApps become increasingly prominent, our upgraded pet shop serves as a testament to the transformative potential of decentralized applications in real-world scenarios.