Best Friends Company Project Report

Content

Motivation

In the past decade, more and more people choose pets as their close companions. Statistics show that of all the types, the majority like to choose cats and dogs as their pets. In recent years, the number of pets in China has increased greatly, and the growth rate of pet cats is higher than that of pet dogs. Statistics show that in 2019, 23% people in China raised pets, while the rate in the US is 56% in 1988. Thus, the overall pet economy still has a lot of room for growth, which is one of the reasons why pet track entrepreneurship and financing have been relatively busy in recent two years.

However, at present, there are few websites dedicated to selling pet supplies on the market, and most of the pet supplies are sold on mainstream websites such as Taobao and JD. These channels have different commodity prices, one-side commodity information, and other problems. Pet owners can't obtain the details of the product quickly and need to spend a lot of time finding the most suitable product for their pets.

Therefore, we would like to establish a website that can easily search for information on pet daily necessities, providing a convenient online purchase channel for pet owners, and helping pet owners to purchase high-quality and cost-effective products and services quickly. At the same time, it is convenient for store owners to know the needs of the public and product feedback from customers. According to the feedback, store owners can change the supply quantity of their own products, and constantly improve their own quality of services.

Assumptions

The goal of our project is to establish a website for e-commerce, our targeted users are three groups of people, the "customer", the "store owner" and "manager". To improve our service, we made some assumptions:

1. All the users should login to our websites using only one identity so that we can provide better services to our users. Thus, the entities under person are disjoint.
2. Our customers are different from the visitors. The former has its own accounts and orders in our database, the latter does not. But the visitors can create an account to visit the websites.
3. To prevent excessive stockpiling, we set the limitations on the volume of the order, that is each time for the same each product, the customer can only buy five of it.

Conceptual Design

We sum up our targeted users to the person entity, and all the products into the product entity. In the next sections, we are going to introduce the end design, that is the design for our database of the project. We will construct the ER diagram by two steps. First, we mainly introduce the main entities of our model. Secondly, we will introduce some relationship sets of our database. The picture 1 shows the ER diagram of our database.

1. **Person:** the entity set product is a super set of managers, customers and store owners. They are all strong entities. Each person has an ID, a password and a name.

   1.1 Manager: Each manager is described by a unique manager ID, which uses the character ‘m’ at the beginning in order to distinguish from other person id. The manager also has a password; They work for the platform. The manager can manage all the orders and check the information for each order.

   1.2 Customer: Each customer is described by a unique customer ID, which uses the character ‘c’ at the beginning in order to distinguish from other person id. The customer also has a password, a name, an address and the credit card id; Each customer who buys a product will have an order, which contains a unique order ID.

   1.3 Owner: Each store owner is described by a unique owner ID, which uses the character ‘ow’ at the beginning in order to distinguish from other person id. The owner also has a password, the store name, an owner name, an address. And owners can check the order information, the inventory and purchase information of the store, which means they can check the suppliers’ information and the products.

   1.4 Visitor: No information, they are the potential customers who use the website to search for information. And they need to sign up for an account in order to login to

the websites. Thus, for this type of person, we will not include any information in the database system until they create an account.

2. **Supplier:** Each supplier is described by a unique supplier ID, which uses the character 'su' at the beginning in order to distinguish from other person id. The supplier also has a name, an address, establishment time and a legal person. The information of each supplier is also provided to stores.
3. **Product:** The entity set product is a super set of pet food, cat litter, health care, toy, and daily necessities. They are all strong entities. Each product has a unique ID, the price, and a name.

   3.1 Pet food: Each pet food is described by a unique pet food ID, which uses the character 'pf' at the beginning in order to distinguish from other person id. And the pet food entity also has the food price, the pet food name, the applicable target, the specification, the usage, the ingredient, the product categories, the place of origin, and the brand and a food expiration date.

   3.2 Car litter: Each cat litter has a unique cat litter ID, the cat litter price, the cat litter name, the type, the smell, the cat litter weight.

   3.3 Health care: Each health care product is depicted by a unique ID, the price and the name, efficacy, and the expiration date.

   3.4 Toy: Each toy is detailed by a unique toy ID, the price, the name, the texture, and the toy weight.

   3.5 Daily necessities: Each daily necessity is illustrated by a unique ID, the price, and the name.
4. **Store:** Each store is described by a unique ID, an address, establishment time, legal person.

Relationship Set

Considering all the realistic cases, we construct some relationship sets between the entities:

a) Inventory: There are several categories of the products on the website, and different categories have some common attributes such as price so we sum them up into a whole entity, defined as products. For each product it can be offered by many stores. And the store owner can manage the inventory of each product in their store. We allow some stores do not have any products considering that some stores are new-established.

b) Create & Consist: Customers and visitors can search for the products based on the information on the websites. When they make their decision, they can create some order lists. And each order list consists of some product id chosen by the customer.

c) Supply: All the products are supplied by the suppliers. And each supplier will supply multiple products and each product can be supplied by multiple suppliers.
d) Manage: For each order list made by the customer, we would like to arrange a manager to manage the after-sale service if there are any problems that occur.

Logical Design

To better implement all the information into our database system, we check the schema for each table, and also examine the normal form of our logical design. The schema of our table is as follows, including the foreign key constraints of some attributes.

customer=(customer_ID, name,password,address, credit_card_ID)
manager=(manager_ID, name,password)
owner=(owner_ID, name,password,address)
order_list=(order_ID,consumer_ID,manager_ID)
store=(store_ID,address,establish_time,legal_person,owner_ID)
cat_litter=(litter_ID,name,price,type,smell,weight)
supplier=(supplier_ID,name,address,legal_person)
health_care=(care_ID,name,price,efficacy,expiration_date)
toy=(toy_ID,name,price,texture,weight)
daily_necessities=(necessity_ID,name,price,efficacy)
pet_food=(food_ID, ,price,brand,ingredient)
inventory=(product_ID store_ID,inventory_num)
supply=(supplier_ID,product_ID)
consist=(order_ID, product_ID)

For all the entity sets, there is only one candidate key ID and that is primary key. As for the order and store, there are also foreign keys as a reference to the schema

All the relationship sets, which all include foreign key will be explained in data implementation.

$F_1 = \{customer\_ID \rightarrow name, password, address, credit\_card\_ID\}$
$F_2 = \{manager\_ID \rightarrow name, password\}$
$F_3 = \{owner\_ID \rightarrow name, password, address\}$
$F_4 = \{customer\_ID \rightarrow order\_ID, manager\_ID\}$
$F_5 = \{store\_ID \rightarrow address, establish\_time, legal\_person, owner\_ID\}$
$F_6 = \{customer\_ID \rightarrow order\_ID, manager\_ID\}$
$F_7 = \{litter\_ID \rightarrow name, price, type, smell, weight\}$
$F_8 = \{supplier \rightarrow name, address, legal\_person\}$
$F_9 = \{care\_ID \rightarrow name, price, efficacy, expiration\_date\}$
$F_{10} = \{toy\_ID \rightarrow name, price, texture, weight\}$

$F_{11} = \{necessity\_ID \rightarrow name, price\}$
$F_{12} = \{food\_ID \rightarrow name, price, brand, ingredient\}$
$F_{13} = \{product\_ID \rightarrow store\_ID, inventory\_num\}$
$F_{14} = \{\}$
$F_{15} = \{\}$

Since a functional dependency is a generalization of the notion of a key, the functional dependencies are also simple, only one. For supply and consist, the whole schema is a key and there is no functional dependency.

According to the definition, the functional dependencies all ID is a super key for R. Besides, the relation with the whole schema as key is also BCNF or 3NF. Therefore, we can conclude that all the information in the ER diagram is implemented in our logical design and also in the database.

Data Implementation

Firstly, a new database named '**project1**' should be added to your server. Then you can import the 'data_generation.sql' from our files. The data in our database has total four types: string, int, date and float. And inside each type, there are some details inside, we will claim them clearly in the following section.

ID attribute (string type):

In order to display some functions on our websites, we generate some data in the database system to do some tests. For all the 'id' attribute in each entity, we use some specific characters at the beginning to distinguish one type from the others. To be more precise, we list the 'id' attributes in the table below. And to ensure that each id is unique within each table, we use the increasing integers combined with the characters to set the id to be the unique among all the tables.

ID for customer	Begins with 'c'
ID for manager	Begins with 'm'
ID for store owner	Begins with 'o'
ID for pet food	Begins with 'pf'
ID for health care	Begins with 'h_c'
ID for cat litter	Begins with 'c_l'
ID for daily necessities	Begins with 'd_n'
ID for toy	Begins with 'toy'
ID for supplier	Begins with 's'

Address attribute (string type):

To be closer to the realistic cases, we generate the address using two steps. We constantly print the 'road' and 'street' but generate the name of the road and street

randomly using some characters. And the attribute is not a key, so it can have some replications in the database. The specific code can be found in the road_gen function and street_gen function above main function in data_gen.cpp

Efficacy/smell/type/texture/ingredient/brand attribute (string type):

For these attributes, we create a data pool and randomly pick the items from the data pool, and the data pools are based on the realistic information searching online. In the table below, we summarize all the data pools we used to generate the data.

type attribute of cat litter	bentonite, tofu, pine, crystal, mixed, paper, zeolite, hay, corn cob, activated carbon, grain
smell attribute of cat litter	green tea, cherry-blossom, peach, no flavor, pine wood, milk, mint
efficacy attribute of health care	Oral hygiene, anti-inflammatory, cleaning, health care, deworming, hair care, disinfection, nutrition product
efficacy attribute of daily necessities	Hair cream, lecithin, probiotics, catnip, deworming
texture attribute of toy	Paper products, chemical fiber, plastic, cotton, resin, metal, silicone, wood and bamboo, water paint
ingredient attribute of pet food	Chicken breast, eggs, salmon, vegetables and fruits, dairy products, beef, pumpkin, broccoli, cereals, bean products, fruits, vitamins, oils
brand attribute of pet food	Royal, Bernard Pure, Crazy Puppy, Jenzy, McFaddie, Wanpy, SOLID GOLD, Pedigree, Origen
name of supplier	Royal, Bernard Pure, Crazy Puppy, Jenzy, McFaddie, Wanpy, SOLID GOLD, Pedigree, Origen

Name/legal person attribute (string type):

For these attributes, we generate some random characters to represent. In the reality, some customers may not want to release their private information, so it is reasonable to use some random string to protect the information of our users. And the name of the supplier is overt and generated from the same data pool as the brand of the pet food. Because from our investigation, we find out that the popular supplier all supply some pet food in the whole pet industry. Additionally, for the legal persons in store and supplier, we also generate some strings consisting of characters. But to your notice, in the real world, the legal person must use some real name. Thus, for application we can update the real information into legal person.

Weight attribute (string type):

For the cat litter and toy products, we will provide some information about the weight of the product in order to give our customers a more precise intuition about the product they brought. Thus, for the weight attribute we use ‘gram’ as our unit and generate some random integers from 30-1000.

Price (float type):

For all the price attributes in our products, like cat litter, health care and so on, we use some float to represent them. Considering the reality, we choose floating point to be one. And numbers are randomly generated from the range 50-900.

Password/credit card id attribute (int type):

The password is important for our users and is used when they need to login to the websites. Regarding both safety and convenience, we choose the six digits number as the password for our customers, store owners and managers. The credit card id is provided by the customers in case that he/she wants to use the credit card to purchase our products. And we use the integers ranging from 20000-30000 to represent and in the real replication, we can update some real information to replace the random numbers.

Establish time/expiration date (date type):

For the pet food products, the customers probably need to get some information about the expiration date of the products. And we randomly generate some date to represent. And the establish time of the stores is important for some customers, so we will also provide that information using the date type in the database. As date type data in SQL is YYYY-MM-DD form of string type, we what we need to do is to specify the year, month, and day of the random number generation range and converted it into a string style.

Extended features (foreign keys/trigger)

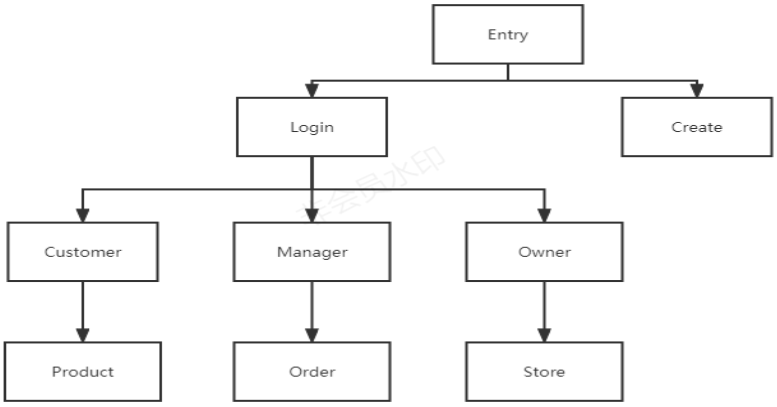
For some relationships, we establish an individual table, such as the table 'consist', 'supply', 'order_list', and 'inventory'. But in these tables, we need to ensure that some attributes are consistent with other tables. Therefore, we use some foreign keys to add some constraints.

Table 'order_list'	The 'customer_id' attribute refers to the 'person_id' attribute in table 'customer'; The 'manager_id' attribute refers to the 'person_id' attribute in table 'manager'; And if the customers delete their accounts, the 'order_list' should also delete the orders of the customers.
Table 'consist'	The 'order_id' attribute refers to the 'order_id' attribute in the 'order_list' table. If an order is deleted from the 'order_list', then the information should also be deleted from the table 'consist'.
Table 'store'	The 'owner_id' refers to the 'owner_id' attribute in table 'owner'. If the store owner deletes her/his account, then the store she/ he manages should also be deleted.

Besides, we notice that the is a supplier is deleted from the 'supplier' table, then all the information of the products supplied by the supplier should also be deleted from the table 'supply'. We use a trigger to complete this function.

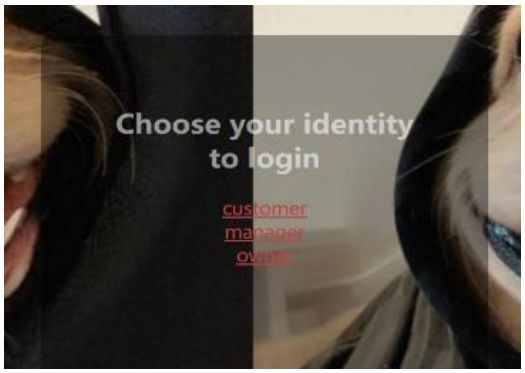
Website Function Designs

In order to improve the user experience, we have designed many functions for our website. And for each type of user, we define some specific functions for them. Next, we will show some details of these functions. Here is our top-to-bottom design:



Identification/Login

When entering into our website, you will choose whether to login (if you have an account) or to create a new account (Only for the visitor). Our targeted users can be divided into three parts, and each time the user can only login using one identification. In order to provide better service for each type of user, we require the user to choose one specific identity, such as ‘manager’, ‘customer’, or ‘owner’. The picture shows the buttons on the websites. This part will introduce the website design by different users.



Customer

Create accounts

Customers are able to create accounts. The customer creates an account to create a new account that is not in the database, and also the created account must start with 'c'. The user must enter his desired id, his nickname, his password, his address, and his credit card account. The password entered by the user is stored as a password on the web side and cannot be seen in the customer information form in the database, so that the user's credit card information is kept safe. After creating an account, it will

automatically return to the login screen.

Login

When customer needs to enter the website, the account and corresponding password must be correct, and the account information must be stored in the customer sheet in the database, then they can login.

View the product List

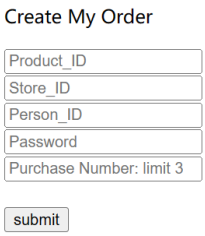
After logging in, the user can see the product category interface, and the user can choose to enter the product category interface they need. We provide five product interfaces.

Search product

Users can search for products. Users can see the product id, product name, store id, price, type, smell and weight of different products in the cat litter interface; see the product id, product name, store id, price, efficacy and expiration date of different products in the health care interface; see the product id, product name, store id, price, texture and weight of different products in the toy interface; see the product id, product name, store id, prices product id, product name, store id, price, ingredient, and brand; in the daily products. When users enter product id in the search box, they can find the information of corresponding products.

Create order

When the user finishes browsing and needs to buy, he can enter the necessary information to create an order in the "create your order" position at the top of the page, in order to update the order in the database. The required information is as follows:



Among this, the product id and store id can be obtained by searching. The user must re-enter his id and the correct password to secure the purchase. The id entered by the user must be the same as the id on their login screen. Also, to prevent hoarding, we set the number of products to be purchased at a time to no more than three.

When the user presses submit, an order id will be automatically generated on the web side, this order id is based on using unique to obtain a unique sequence of non-repeating numbers based on the current microsecond number, and intercepted part of the date combined with the current day as the order number, to ensure that the order number will not be repeated. At the same time, a random manager is assigned to manage the order on the web side, and the manager's information is extracted from the manager table in the database. This information is stored in the order_list, consist, and inventory tables in the database together with the information entered by the user.

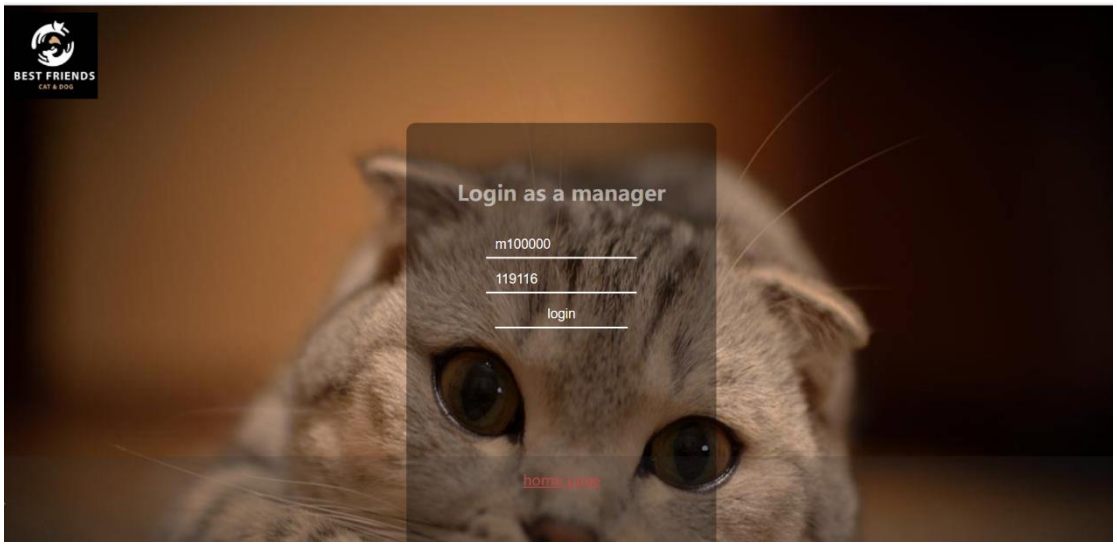
Manager

On the pet store website, managers can login to their accounts and view all the order information including ID of the order(order_id), the customers who bought the good (person_id) and the specific good (product_id). Further, the manager can search the specific order by the order_id io the website. At last, manager can click the footer and then go back to the login page directly.

Login as a manager

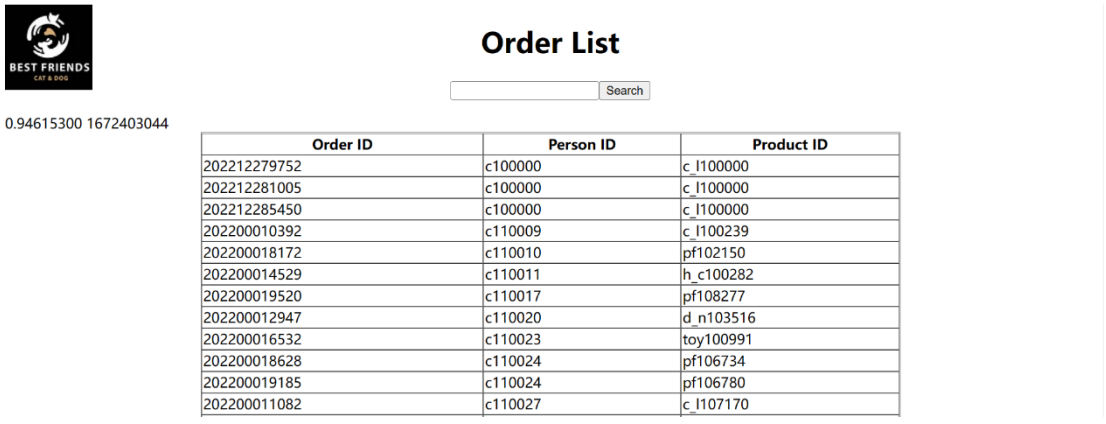
Managers are required to input the manager ID and corresponding password to login their own website.

Sample: manager_id: m100000, password: 119116



View the order

Manager can view the order list which contains the order_id, person_id and the product_id. The top 50 items will show on the order list.

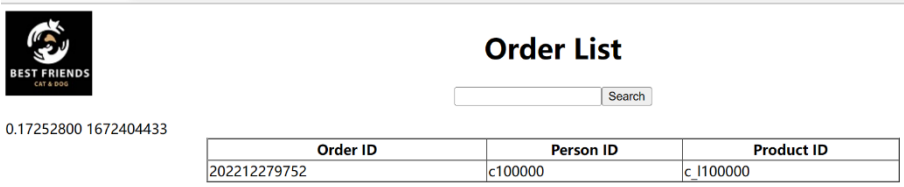


Order ID	Person ID	Product ID
202212279752	c100000	c_l100000
202212281005	c100000	c_l100000
202212285450	c100000	c_l100000
202200010392	c110009	c_l100239
202200018172	c110010	pf102150
202200014529	c110011	h_c100282
202200019520	c110017	pf108277
202200012947	c110020	d_n103516
202200016532	c110023	toy100991
202200018628	c110024	pf106734
202200019185	c110024	pf106780
202200011082	c110027	c_l107170

The order_id and the person_id are selected from the table order_list. The product_id is selected from table inventory.

Search the order

If the manager wants to find the specific order, they can search for it by the product_id. Manager input the whole id of product and click “search” button, then the corresponding information of order will show on the website. Clean the search box and click the “Search” button again will show all the orders again on the website.



For the database level, the variable will be got from the search box and be compared with all the order_id in the database.

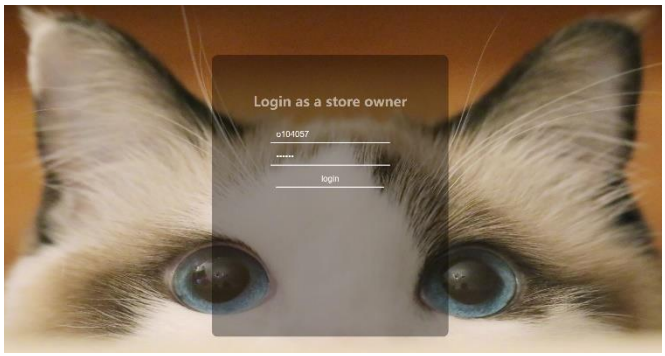
Homepage

If the managers want to go back to the login page to login in other accounts, they can click the “homepage” button and go back directly.

Owner

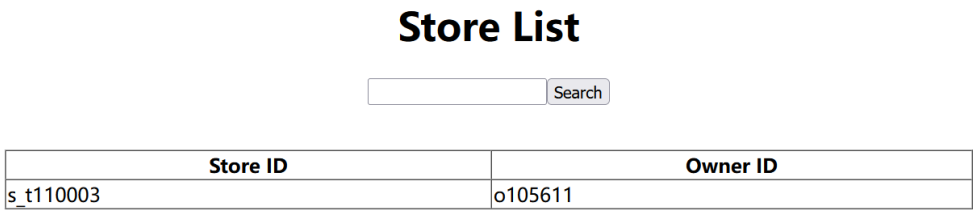
Login

After one owner inputs into his or her ID and password, this page will jump to another page which shows all the stores he or she owns.



Search store

After entering this page, if the owner manages a lot of stores and wants to find a specific one, she/he can search for the store by inputting the store ID. In the database, we only use some test data, so it is very easier for the owner to find the store she/he wants. But if more and more data is input, then this function will be really useful.

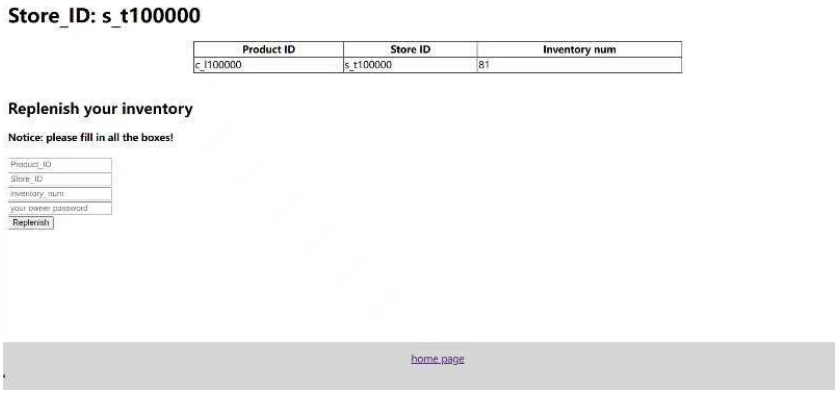


After the research returns its results, the owner can click on the store in which she/he is interested. (For the detailed information is extremely complicated and considerable, we just create three stores for the owner with ID o104057, password is 110361.)

Enter the store

After entering the web page of the store, the owner can see the types of products

and then should input the corresponding product ID, store ID, number of inventory and the password to replenish the inventory.



Summary

From both the database part and the website-design part, we introduce you a very convenience and useful program. We might encounter some problems if we implement some realistic data into our system, but we are confident to face all these challenges. Our object is to provide more efficient and intellectual services to our users and we will continue to improve our program based on all the comments. In the future, more participants might be entered the business, such that we are considering cooperating with some animal hospitals. And we will update and complete our database and websites to a higher level.

All your support and suggestions will be great contributions to us. Thanks as always.

Best Friends Company
December 31^{th}, 2022

Appendix:

The file ‘data_generation.sql’ can be imported to the database ‘project1’.

The file ‘data.cpp’ is the code for generating the queries.

The file ‘html’ contains all the websites files.