Task 1

Group Members: Nolan Spencer

# Project Details

## Purpose

The goal of this database is to create a way store and manage the animals that are put up for adoption in adoption facilities. Since there are many dogs that are constantly being moved around or adopted, etc. It is essential to keep track of all the data that they have associated with the animal so that the owners/facilities will have the correct information with their pet.

To do this, my database will store entries on the animals, their type, their breed, their picture, the facility they are in and its information, as well as if someone wants to adopt them, they will have to provide their information. The benefits to this are that it will keep accurate information on all the animals, and it will make sure that the correct information is being distributed so that the facility can keep operations running efficiently.

## Potential Customers & Users

The potential customers and users for this database would involve the adoption facilities as well as the customers who would like to adopt a new pet. The constant transference of pets causes the need for good infrastructure. They need to be able to care for these pets as well as transfer their data where it is needed. Millions of pets are adopted every year. It is important to make sure all the information is given to the new facility/owner of the pet. This will allow the operation of the facility to be very smooth.

## Data Requirements

For this project, the scope of what I am doing is much smaller than that of the adoption facility. I will be cutting back on some things since I am only one person and not a non-profit organization.

The Primary data requirements are listed and explained below.

## Pet

This data will be the most important since it will be solely about what this website is for. The information that will be pertaining to the pet will be their name, age, a description of the dog and their life, weight, size (one of three S,M,L), if their active or not, the date that the facility received them, if they are spaded/neutered, which facility they are located at, their breed, and whether they are a dog or cat. It is important to be able to reference the data so that we can keep a good record of all the dogs in the system for us and the user.

## Breed

This data will be supporting the pet information. This table will be to help identify what breed the dog is. The information included in this data will be the name of the breed, a description of the breed, and whether the breed is a cat or dog breed.

## Type

This data will be supporting the pet and breed information. This table will be to identify something if it is a dog or cat. The information included will be a description of the animal and an identifier or ID.

## Pictures

This data will be supporting the pet information. This data will be where you store your images for the pet. The information that will be included in this table will be an image path which will just be the name of the picture file, a caption of the file, and the pet’s id that it is linked to. This will ensure that we have the correct photos for the correct pet.

## Location

This table will also be important. The information in this table will be all the information at which facility a specific pet will be held at. The information included in this will be the name of the location, their address, city, state, and zip. It will also include their contact information so, email and phone number.

## Contact

This table will be the user table. This is going to be all the information regarding a user on the account. This is important because the user will be sending information to the website on a dog if they want to adopt or not. The information that will be included will be the users first name, last name, phone number, email, address , city ,state ,and zip. To secure this users information the password will be also in this table. This will ensure users data, and no one will be able to login.

## Adoption

This will be used to store the data of the request to adopt a pet. What will be stored in this will be the adopt date (the date of when you request to adopt), the adoption fee, their contact id so that we can correctly Identify who is making the request, as well as the pet’s id so that we can correlate the pet and new owner.

These are going to be the general requirements of the data, but they are subject to change during the project.

## Functional Requirements

The functional requirements of the database will have to work with that of what I am storing in the database. I will need an interface to display data.

There will be a login feature so that the user may log into their account and send requests to adopt a pet. There will also have to be a signup feature which will create their data and place it into the database. Since there will be a password involved, I am encrypting the password so that the person can have the ease of mind that their data will not be stolen. There will also be a logout feature, where the user may logout of their account.

I will need to implement a way to see all the dogs’ data, including from their supporting tables. I will also need to style it in a way for the user to have a friendly experience looking at the dog and all the information. In addition, I will need to implement a way to see the facilities and their information so that the user may contact them if needed.

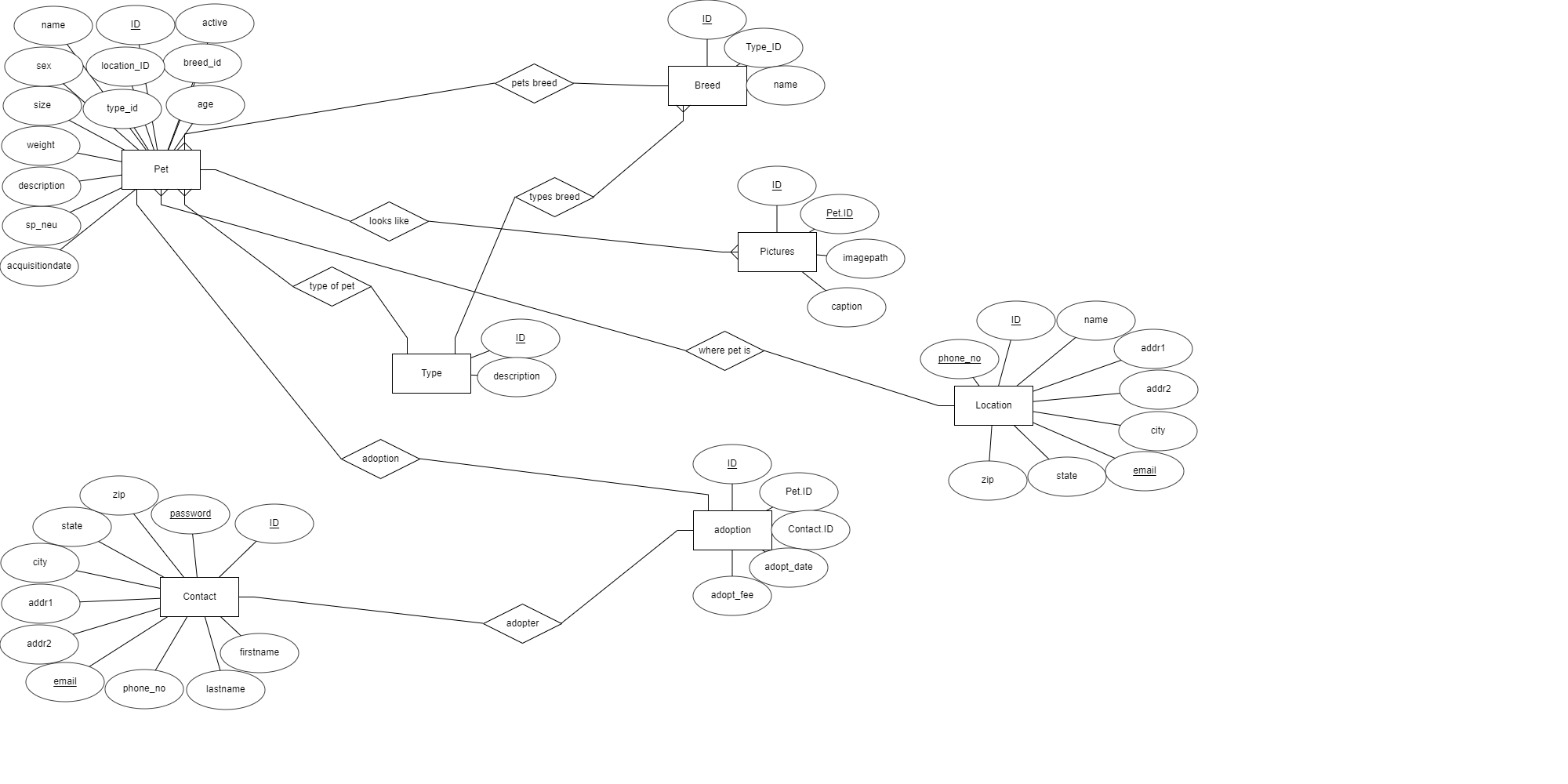
END OF TASK 1

TASK 2

## Conceptual design

There are many relationships between the various entity sets such as what the pets breed is, what the pets’ type is, and many more. They are all shown below.

## ER Diagram



## Explanation

In this ERD the square boxes represent the tables, and the circles attached are the various entity’s (columns) in that table. The diamonds are the relationships between each table. If the entity is underlined, then that means that the entity is unique. All the IDs of each table are the PRIMARY KEYS. I was going to bold them however I was unable to. The foreign keys are represented by their main table followed by .id or \_id.

## Relationships

## Pet and Location

The relationships between pet and location is very simple. Pet will have a location id linked to the one of the facilities ids. This is important because we will need to know where the pet is located as well as know all the details of that location so that the user may contact the location if they wish to visit the pet or get anymore information about the pet.

## Pet, Breed, Pictures

The relationships between pet breed and pictures is also fairly simple. Pet has the id located with the type id. This will help identify what type of pet it is. Then Pet also has the id corresponding to an id of a breed. There can be many pets to a breed or different breeds. They will correlate by comparing their respective id’s. This process of having support tables for the pet using separate tables for breed and pictures will help make accessing data much easier.

## Relational Schema Diagram

## Relational Schema

# Project Implementation

## Introduction

To implement the project I used node.js, express, niginx, vue, and mysql. Node.js and Express were used to implement the backend of the project. Vue was used to implement the front end. Lastly, NGINX was used as a way to host the server.

## Features

For the features I was not able to get all of the features I wanted to include since the learning curve of how to make a webapp was so large. I implemented most of the features. However, there was not a feature to create and delete a pets data. However, the functions mostly revolve around the contact table and manipulating and creating data from its data.

## SQL Implementation

I used SQL in many ways in this project.

First off, I used a select statement to select all of the data of the pets that I wanted to be sent from the database to the front end. I used this to show the data on the adopt page. I also used select statement to show the facilities that were under my adoption center. I also used select statement to implement my login feature. I also use express-session to keep the user logged in (unless they logout). Lastly, I used select to show the records of requests for adoption of a specific dog.

I used Insert statements multiple times to create new data. I used it so that I could create a new user in the contact table. I also used bcrypt to hash the password that was given. This way the user would have protection in for their account if someone tried to open the database and get their password. This will also prevent SQL injection. I also used an INSERT statement for the adopt button. This button will create a “request” and create data in the adoption table. It will put the current date and the fee for every dog which I specified to be $250.

Lastly, I used the Delete statement when deleting your account. This will take the email and password that is entered and crosschecks them to make sure they are in the system. Then it will delete the users data that is stored in the database.

# Conclusion

## Source code

Source code for the web-app: <https://github.com/NolSpencer/db-project>

Source code for database: <https://github.com/NolSpencer/db-project/blob/main/export.sql>

## Current Problems

There are a few problems with the website that I would like to discuss. Unfortunately, I was not able to fix in time.

First problem: There is nothing that shows up on your screen if you logged in, logged out or signed up properly. It also will not tell you if you deleted your account properly on your screen.

Second problem: If you try to click the signup button with the same email, it will crash the server since the email in contact was declared as a UNIQUE, therefore it cannot create more data if an account with that email already exists.

Third problem: If you try to delete your account it could say it that your account was deleted in the console log, however, it cannot delete your data if the email associated with your account has any requests to adopt a pet. This is because of the contact\_id being a foreign key. Therefore, if you want to delete an account that has records you must be able to delete the records.

## Future Implementation

If I were to work on this project in the future, I would first solve all the problems that I have addressed in the current problems section. Next, I would make it so that not just anyone can create or delete data. It would have to be a specific account that has access to those features. I would also make it so that this admin is the only one to be able to create facility/pet/picture data. I would also update the UI to make it more user friendly. In the future I would like to automate the process of add/deleting pets data by using an API to get the information of the nearby shelters.

## Closing Remarks

I learned a lot from this project. I had to learn how to do frontend and backend code. I am glad that I was able to take this on. I will be updating this in the future to make it even better and cooler. I learned a lot from the errors that I made.

Live-site of project: {{PUT LINK HERE}}

If site is broken or not running, please email me nspence9@kent.edu