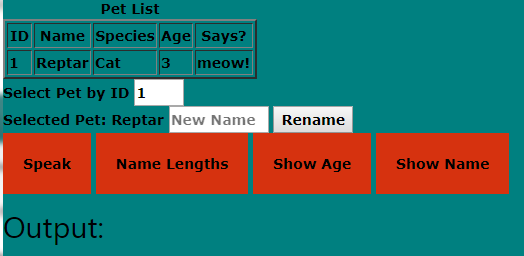
**Pet Tracking Tool**

**Instructions:**



I used a web form to handle pet input and creation. Pet Name is the only required field in order to create a pet, the rest have default values if left blank.

After pet creation a table is shown to display pets added to the list and information entered.



You are now able to select a pet via its corresponding ID and perform the desired action by clicking on the buttons.

**Rename:**

Allows the user to rename the selected pet via the input box and then displays the name change and average name length of all past and current names.

**Speak:**

Makes the selected pet speak , if the speak is a custom sound that doesn’t match the default sound for that species cat/dog it will speak again with the default sound.

**Name Lengths:**

Calculates the average name length of the selected pet including current and past names.

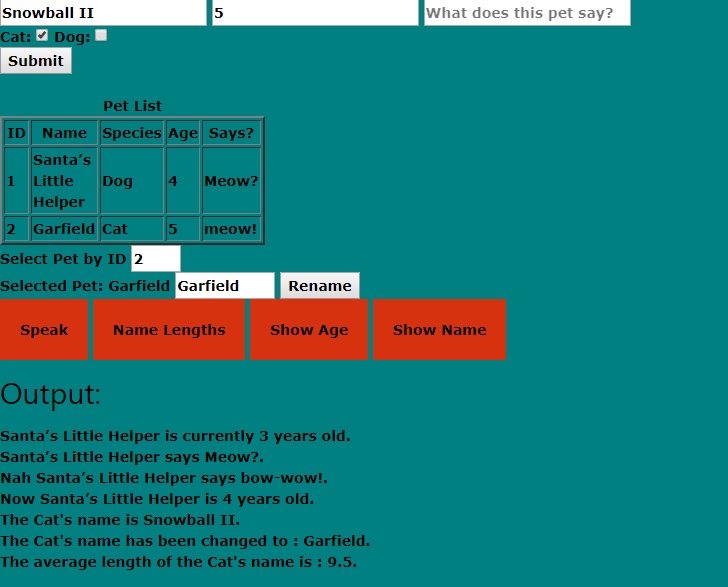
**Show Age:**

Shows age of pet to output.

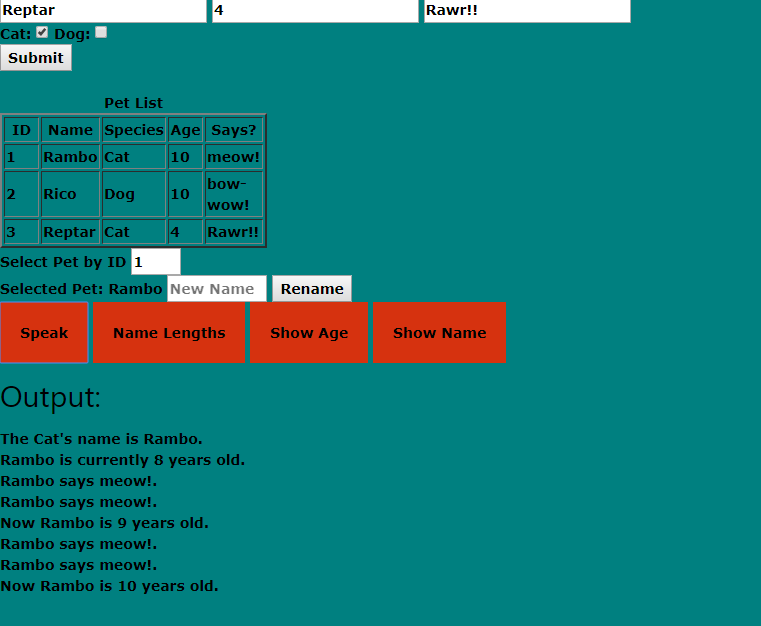
**Show Name:**

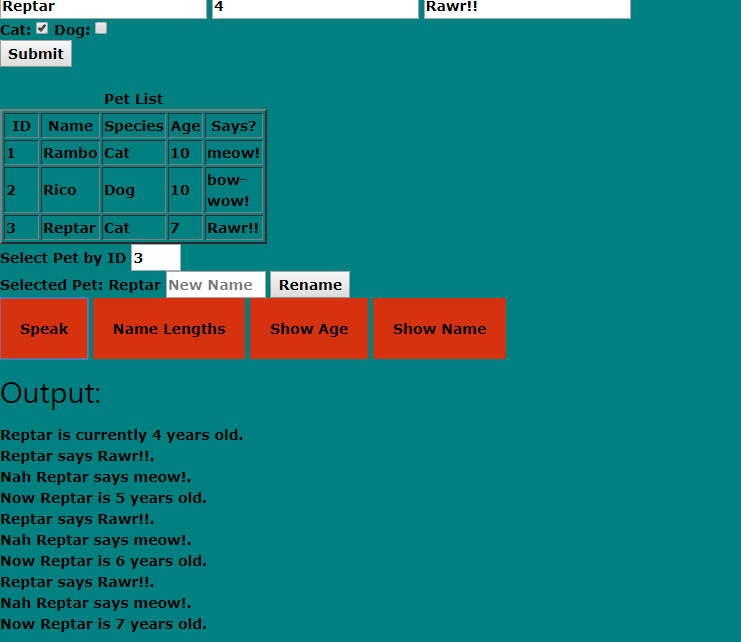
Shows Name of pet to output.

**7) The program needs to do the following things in order, utilizing the pet entities you have created. Any output specified should be on its own line.**

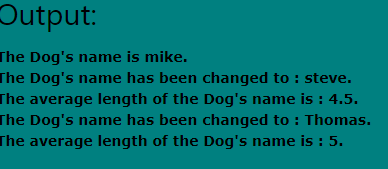
****

**8) Create a test that tests step #5 (the test should verify that each pet increments its age by 1 every 2 times it speaks).**

****

****

**9) Create a test that tests step #6 (the test should verify that we’re able to correctly calculate the average of all names given to the pet).**



**Mike(4)+**

**Steve(5)**

**9/2=4.5**

**+Thomas(6)**

**15/3 =5**

**10) Now we want to persist the state of the cats and dogs to database tables. Write SQL that would create the schema necessary to do so. We need to store not only their current name and age, but also a historical record of previous names for each.**

I would store the pet objects into a table that can be updated and retrieved. Also I would create an additional table to store the collections of previous pet names.

CREATE TABLE Pets (  
   ID int,  
   Name varchar(255),  
   Species varchar(255),  
   Age int,  
   Says varchar(255),

TimesSpoken int,

);

CREATE TABLE PetNames (  
   ID int,  
   Name varchar(255),  
);

**11) List some ways to abstract the dogs and cats in your program if you haven't already implemented those abstractions.**

INSERT INTO Pets  
VALUES (1, “Reptar”, “Cat”, 3, “Meow!”,0);

SELECT Name FROM PetNames

WHERE ID =1;

ORDER BY Name ASC

**12) Explain your data modeling strategy for step #10. What performance optimizations did you make / could you make, and what trade-offs do those optimizations incur?**

I Just wanted to setup a basic table to be able to store the pet data to be persistent between runs and also store all the previous pet names.

To optimize this schema I would spend more time creating smaller relational tables instead of just one “Pets” table.