

Table of Contents

Abstract Code / Inline SQL	2
Logging in	2
Animal Dashboard	2
Add Animal	5
View Animal Details	6
Add Vaccination	7
View Vaccine Reminder Report	8
Add Adoption	9
Add Adoption Application	10
Adoption Application Review	11
Lookup Volunteer	13
Display Monthly Adoption Report	13
View Animal Control Report	15
Display Volunteer of the Month	17

Abstract Code / Inline SQL

Logging in

Login form

While **login** button not pushed, do nothing

When the **login** Button is pushed, do the following:

username and password = get *username* ('\$UserName') and *password* ('\$Password')
from form

call login function

login('\$UserName', '\$Password'):

if forms are valid:

Authenticate the username ('\$UserName') and *password* ('\$Password')

```
SELECT password FROM Users WHERE UserName = '$UserName'
```

if *username* is found and `Users.password == '$Password'`:

Store Login information as a session variable '\$UserName'

Go to the **Animal Dashboard** for user

else:

Display error message with wrong username/password, display **Login** form

else:

Display error message with wrong username/password, **display login form**

Animal Dashboard

Animal Dashboard

View Animal Dashboard, Populate all animals/animal information

```
SELECT
  Animal.Name,
  Animal.Species,
  GROUP_CONCAT(AnimalBreeds.Breed_Name ORDER BY
  AnimalBreeds.Breed_Name SEPARATOR '/') as Breed_Name,
  Animal.Sex,
  Animal.Alteration_Status,
  Animal.Age,
  AdoptionApplication.State as Adoptability_Status
FROM Animal
INNER JOIN AnimalBreeds
ON Animal.Pet_ID = AnimalBreeds.Pet_ID
INNER JOIN AdoptionApplication
```

```
ON Animal.Adoption_Application_Number = AdoptionApplication.Application_Number
GROUP BY
  Animal.Name,
  Animal.Species,
  Animal.Sex,
  Animal.Alteration_Status,
  Animal.Age,
  AdoptionApplication.State as Adoptability_Status
```

While no buttons are pressed, do nothing

When a button is pressed, do the following:

if **species filter** is pressed:

```
SELECT DISTINCT Name FROM Species;
```

Populate filter with species

'\$Species' = *Select species* to filter

```
SELECT
  Animal.Name,
  Animal.Species,
  GROUP_CONCAT(AnimalBreeds.Breed_Name ORDER BY
  AnimalBreeds.Breed_Name SEPARATOR '/') as Breed_Name,
  Animal.Sex,
  Animal.Alteration_Status,
  Animal.Age,
  AdoptionApplication.State
FROM Animal
INNER JOIN AnimalBreeds
ON Animal.Pet_ID = AnimalBreeds.Pet_ID
INNER JOIN AdoptionApplication
ON Animal.Adoption_Application_Number = AdoptionApplication.Application_Number
WHERE Animal.Species = '$Species';
GROUP BY
  Animal.Name,
  Animal.Species,
  Animal.Sex,
  Animal.Alteration_Status,
  Animal.Age,
  AdoptionApplication.State
```

Update dashboard with selected species

View Animal Dashboard

if **adoptability status filter** is pressed:

```
SELECT DISTINCT `State` FROM AdoptionApplication;
```

Populate filter with adoptability status

'\$Adoptability_Status' = *Select adoptability status* to filter

```
SELECT
  Animal.Name,
  Animal.Species,
  GROUP_CONCAT(AnimalBreeds.Breed_Name ORDER BY
  AnimalBreeds.Breed_Name SEPARATOR '/') as Breed_Name,
  Animal.Sex,
  Animal.Alteration_Status,
  Animal.Age,
  AdoptionApplication.State
FROM Animal
INNER JOIN AnimalBreeds
ON Animal.Pet_ID = AnimalBreeds.Pet_ID
INNER JOIN AdoptionApplication
ON Animal.Adoption_Application_Number = AdoptionApplication.Application_Number
WHERE AdoptionApplication.State = '$Adoptability_Status'
GROUP BY
  Animal.Name,
  Animal.Species,
  Animal.Sex,
  Animal.Alteration_Status,
  Animal.Age,
  AdoptionApplication.State
```

Update dashboard with selected adoptability status

View Animal Dashboard

if **clear filter** is pressed:

Clear filters

View Animal Dashboard

if **animal name button** is pressed:

Go to Animal Detail Screen

Get user information

```
SELECT is_Admin FROM Employees WHERE UserName = '$UserName';
```

If `Employees.is_Admin == True`:

retrieve number of spaces in adoption center

```
SELECT
    Species.Name,
    s.Species_in_Shelter
FROM Species
INNER JOIN (
    SELECT
        Species,
        Count(Species) as Species_in_Shelter
    FROM Animal
    GROUP BY Species
) s
ON s.Species = Species.Name
```

Display number of spaces in adoption center

View Animal Dashboard

if `Employees.is_Admin` != Null:

display add animal button

display add adoption button

View Animal Dashboard

if **add animal** button is pressed:

go to **add animal** form

if **add adoption** button is pressed:

go to **add adoption** form

Add Animal

Add animal form

Display **add animal** form to fill in all information about -
animal(name, species, breed, sex, alteration status, age,and adoptability status)

While **submit animal** button is not pushed, do nothing

When **submit animal** button is pushed, do the following:

if forms are and complete:

`$Species` = Get species from Form

Check number of species to see if space available

```
SELECT
    Species.Name,
    IF(s.Species_in_Shelter<Species.Max_Per_Shelter, "1", "0") as ROOM_IN_SHELTER
FROM Species
INNER JOIN (
    SELECT
        Species,
        Count(Species) as Species_in_Shelter
```

```
FROM Animal
GROUP BY Species
) s
ON s.Species = Species.Name
WHERE s.Species = '$Species';
```

```
if s.ROOM_IN_SHELTER == 1:
    Tell user that information is correct
    Go to Animal Detail Screen
else:
    Tell user that space is full
    go to Add animal form screen
else:
    Display error message with invalid form, display animal form again
```

View Animal Details

Abstract Code - View Animal's Detail Dialog

View animal details upon clicked, will display a list of the animals details including all the fields mentioned.

While Dialog is open:

Display all Animal information included, adoption status, vaccinations and animal details

```
SELECT Pet_ID, Name, Description, Age, Microchip_ID, Sex, Alteration_Status,
Surrender_Reason, Surrender_By_Animal_Control, Surrender_Date, Adoption_Date,
Adoption_Fee, Adoption_Application_Number, Species
FROM Animal
WHERE Pet_ID = '$PetID'
```

- Read Animal results row into *\$Animal*
- Display fields from *\$Animal* on screen
- Execute query to display list of Animal Breeds

```
SELECT Name
FROM AnimalBreeds
WHERE Pet_ID = '$PetID'
```

- For each row returned from query, display fields in a table
- Execute query to return list of vaccines administered to animal

```
SELECT
  Vaccine_Type, Vaccination_Number, Date_Administired,
  Expiration_Date, Vaccine_Submitter
FROM VaccineAdministration
WHERE Pet_ID = '$PetID'
```

- For each row returned from query, display fields in a table
- if user has appropriate permissions

```
SELECT e.Username
FROM Employees AS e
WHERE e.UserName = '$UserName' AND e.is_Admin = TRUE
```

- if *\$animal.Adoption_Date*==NULL (animal can be adopted):
 - Display button to adopt animal
- if does not have any vaccinations:
 - Add option to add vaccination (displayed below)
 - If nothing is pressed do nothing
 - If button is pressed, redirect to add Vaccination page

```
SELECT a.Pet_ID, COUNT(va.Vaccine_Type)
FROM Animal AS a INNER JOIN VaccineAdministration AS va on a.Pet_ID =
va.Pet_ID
GROUP BY a.Pet_ID
HAVING COUNT(va.Vaccine_Type) = 0
```

- Else:
 - Display Information mentioned above

Add Vaccination

Abstract Code - Add Vaccination form

Located in the Animal Details Page, a button with 'add vaccine' when pushed will display Display Dialogue box with a list of Vaccinations corresponding to that breed of animal coming from the Vaccine Appendix in the Database

While **add vaccine** button is not pushed, do nothing

When **add vaccine** button is pushed, do the following:

User will be prompted to enter a vaccine in a text box (optional autocomplete)

- if vaccine does not exist:
 - Will through error saying to pick a vaccine from the appendix

```
SELECT COUNT(*)
FROM Vaccine
FROM Animal AS a INNER JOIN VaccineAdministration AS va on a.Pet_ID =
va.Pet_ID INNER JOIN Vaccine v on va.VaccineType = v.VaccineType AND
va.Species_Name = v.Species_Name
WHERE va.VaccineType = '$Vaccine_Type' and va.Pet_ID = '$PetID'
```

- else:
 - if animal already has that *vaccine*:
 - Error will be thrown saying that the animal already has that *vaccine*
 - Prompt user to try adding a Vaccine from the database
- Else:
 - Database will be updated with animal's new *vaccinations* that the user chose for the animal
 - User will have to enter vaccination date, and next dose date
 - Optional: vaccine/tag number

```
INSERT INTO VaccineAdministration (Pet_ID, Species_Name, Vaccine_Type,
Vaccination_Number, Date_Administired, Expiration_Date, Vaccine_Submitter)
VALUES ( '$PetID', '$SpeciesName', '$VaccineType', '$VaccinationNumber',
'$DateAdministired', '$ExpirationDate', '$VaccineSubmitter')
```

- If user closes box without entering appropriate vaccine
 - Dialogue box will close

View Vaccine Reminder Report

Abstract Code - **View Vaccine Reminder Report**

Seperate page with all the animals in a table with their corresponding vaccines expiration date.

- User clicks the **View Report Button**
- If Vaccination is due in the current and next three months:
 - Display vaccination type
 - Display vaccination due date
 - Display pet ID, species
 - Display the breed - array format just in case there are multiple
 - Display rest of fields such as: sex, alteration status, microID, surrender date

- First and last name of person who recorded the last vaccination of this type for this animal.
- Data will be displayed in ascending order of the vaccination due date && ny pet ID ascending

```
SELECT va.Pet_ID, va.Species_Name, va.Vaccine_Type, va.Vaccination_Number,
va.Date_Adminisistired, va.Expiration_Date, va.Vaccine_Submitter,
GROUP_CONCAT(va.Breed_Name SEPARATOR '/') as Breed_Name, a.Name,
a.Description, a.Age, a.Microchip_ID, a.Sex, a.Alteration_Status, a.Surrender_Reason,
a.Surrender_By_Animal_Control, a.Surrender_Date, a.Adoption_Date,
a.Adoption_Fee, a.Adoption_Application_Number, a.Species
FROM VaccineAdministration AS va
INNER JOIN Animal AS a on va.Pet_ID = a.Pet_ID
WHERE va.Expiration_Date >= DATEADD(month, -3, getdate())
GROUP BY va.Pet_ID, va.Species_Name, va.Vaccine_Type, va.Vaccination_Number,
va.Date_Adminisistired, va.Expiration_Date, va.Vaccine_Submitter, a.Name,
a.Description, a.Age, a.Microchip_ID, a.Sex, a.Alteration_Status, a.Surrender_Reason,
a.Surrender_By_Animal_Control, a.Surrender_Date, a.Adoption_Date,
a.Adoption_Fee, a.Adoption_Application_Number, a.Species
ORDER BY va.Expiration_Date, va.Pet_ID
```

Add Adoption

Abstract Code - **Add Adoption** Form

- User clicks on **Add Adoption** button from **Animal Detail**:
- Prompt **Search Dialog** to look up an approved adopter. User can input *Applicant Last Name* (\$ApplicantLastName) and *Co-Applicant Last Name* (\$CoApplicantLastName)
- While **Search** button is not pressed, do nothing
- When **Search** button is pressed, then:

```
SELECT Application_Number, Applicant_First_Name, Applicant_Last_Name, Street,
City, Adopter.State, ZIPCode, Phone_Number, Adopter.Email_Address,
CoApplicant_First_Name, CoApplicant_Last_Name
FROM Adopter INNER JOIN AdoptionApplication ON Adopter.Email_Address =
AdoptionApplication.Email_Address
WHERE AdoptionApplication.State = 'Approved' AND (Applicant_Last_Name LIKE
'$ApplicantLastName%') AND (CoApplicant_Last_Name LIKE
'$CoApplicantLastName%') AND Application_Number NOT IN
(SELECT Adoption_Application_Number
FROM Animal
WHERE Adoption_Application_Number IS NOT NULL)
```

- Upon selection of Adopter
 - Prompt User to enter *Adoption Date* (\$AdoptionDate) and *Adoption Fee* (\$AdoptionFee).
- When **Submit** button is clicked. The selected Application Number is used in the following query, as well as the PetID from the selected animal.

```
UPDATE Animal
SET Adoption_Application_Number = '$ApplicationNumber', Adoption_Date =
'$AdoptionDate', Adoption_Fee = '$AdoptionFee'
WHERE Pet_ID = '$PetID'
```

Add Adoption Application

Abstract Code - Add Adoption Application Form

- Display Add Adoption Application form, allowing user to fill in the following fields:
 - *Applicant's First Name* (\$ApplicantFirstName)
 - *Applicant's Last Name* (\$ApplicantLastName)
 - *Co-Applicant first name* and *Co-Applicant last name* (Optional) (\$CoApplicantFirstName & \$CoApplicantLastName)
 - Address: *Street, city, state, zip code* (\$Street, \$City, \$State, \$ZIPCode)
 - *Phone number* (\$PhoneNumber)
 - *Email address* (\$EmailAddress)
 - *Date of Application* (\$DateOfApplication)
- While **Submit Adoption Application** button is not pushed, do nothing
- When **Submit Adoption Application** button is pushed, do the following:
 - Verify all required fields are filled in and match their data type.
Co-Applicant first name and *Co-Applicant last name* fields need to be both filled in or both empty for application to be valid.
 - If all fields are valid, then:
 - Verify if the Adopter already exists in the database by attempting to retrieve the email address.

```
SELECT Email_Address
FROM Adopter
WHERE Email_Address = '$EmailAddress'
```

- If adopter does not exist, then insert a new row with its information.

```
INSERT INTO Adopter (Email_Address, Phone_Number, Street, City, `State`,
ZIPCode, Applicant_First_Name, Applicant_Last_Name)
VALUES ( '$EmailAddress', '$PhoneNumber', '$Street', '$City', '$State', '$ZIPCode',
'$ApplicantFirstName', '$ApplicantLastName')
```

- Insert the new row with the information from the application form. Also, the Application Number will be generated automatically because Application_Number in the AdoptionApplication table has AutoIncrement.

```
INSERT INTO AdoptionApplication (Email_Address, Date_Of_Application,
CoApplicant_First_Name, CoApplicant_Last_Name, `State`)
VALUES ( '$EmailAddress', '$DateOfApplication', '$CoApplicantFirstName',
'$CoApplicantLastName', 'Pending Approval')
```

- Retrieve the newly generated Application_Number to display it in the UI.

```
SELECT Application_Number
FROM AdoptionApplication
WHERE Email_Address= '$EmailAddress' AND Date_Of_Application =
'$DateOfApplication' AND CoApplicant_First_Name ='$CoApplicantFirstName' AND
CoApplicant_Last_Name = '$CoApplicantLastName'
```

- Display a Success message; display AdoptionApplication.Application_Number
- Else
 - Display error message with invalid form, keep displaying **Add Adoption Application** form

Adoption Application Review

Abstract Code - **Adoption Application Review Form**

- **Adoption Application Review** button is visible in **Animal Dashboard** only for Admin users
- User clicks **Adoption Application Review** button from **Animal Dashboard**
 - Find each AdoptionApplication where AdoptionApplication.State == "Pending Approval"

```
SELECT Application_Number, Applicant_First_Name, Applicant_Last_Name, Street,
City, Adopter.State, ZIPCode, Phone_Number, Adopter.Email_Address,
CoApplicant_First_Name, CoApplicant_Last_Name
FROM Adopter INNER JOIN AdoptionApplication ON Adopter.Email_Address =
AdoptionApplication.Email_Address
WHERE AdoptionApplication.State = 'Pending Approval'
```

- Display an **Approve** and **Reject** buttons for each set of results:
Adoption Application + Adopter
- Upon click of **Approve** button
 - Set AdoptionApplication.State = "Approved" for that application.
The selected Application Number is passed to the query as \$ApplicationNumber

```
UPDATE AdoptionApplication
SET AdoptionApplication.State = 'Approved'
WHERE Adoption_Application_Number = '$ApplicationNumber'
```

- Remove that AdoptionApplication tuple from **Adoption Application Review**. Run previous query to retrieve all applications pending approval.

```
SELECT Application_Number, Applicant_First_Name, Applicant_Last_Name, Street,
City, Adopter.State, ZIPCode, Phone_Number, Adopter.Email_Address,
CoApplicant_First_Name, CoApplicant_Last_Name
FROM Adopter INNER JOIN AdoptionApplication ON Adopter.Email_Address =
AdoptionApplication.Email_Address
WHERE AdoptionApplication.State = 'Pending Approval'
```

- Upon click of **Reject** button
 - Set AdoptionApplication.State = "Rejected" for that application.
The selected Application Number is passed to the query as \$ApplicationNumber.

```
UPDATE AdoptionApplication
SET AdoptionApplication.State = 'Rejected'
WHERE Adoption_Application_Number = '$ApplicationNumber'
```

- Remove that AdoptionApplication tuple from **Adoption Application Review**. Run previous query to retrieve all applications pending approval.

```
SELECT Application_Number, Applicant_First_Name, Applicant_Last_Name, Street,
City, Adopter.`State`, ZIPCode, Phone_Number, Adopter.Email_Address,
CoApplicant_First_Name, CoApplicant_Last_Name
FROM Adopter INNER JOIN AdoptionApplication ON Adopter.Email_Address =
AdoptionApplication.Email_Address
WHERE AdoptionApplication.`State` = 'Pending Approval'
```

- Upon click of **Close** button on the **Adoption Application Review** form, form is closed and user is back in **Animal Dashboard**

Lookup Volunteer

Abstract Code

- Display form with input fields of *first_name* (*\$FirstName*) and *last_name* (*\$LastName*)
- Wait for user to press **Lookup** button
- If **Lookup** button is pressed:
 - If *first_name* and *last_name* form fields are empty:
 - Display message: "Please enter first name and/or last name. You may enter just the beginning of the name as well."
- else:
 - Collect data from form fields into variables *\$FirstName* and *\$LastName*
 - Execute query on RDBMS:

```
SELECT u.First_Name, u.Last_Name, u.Email_Address, v.Phone_Number
FROM Volunteer AS v
INNER JOIN Users AS u on v.Username = u.Username
WHERE u.Last_Name like '$LastName%' OR u.First_Name like '$FirstName%'
ORDER BY u.Last_Name, u.First_Name;
```

- Display a table header with following columns of First name, Last name, Email Address and Phone Number
- For each \$row in rows returned from database, display \$row.FirstName, \$row.LastName, \$row.EmailAddress, \$row.PhoneNumber

Display Monthly Adoption Report

Abstract Code

- Execute query on RDBMS:

```

SELECT DISTINCT Yr_Month, Species, Breed_Name, Adoption_Count, Surrender_Count
FROM (
    SELECT
        dates.Yr_Month, Species, Breed_Name,
        IFNULL(SUM(Adoption_Count),0) as Adoption_Count,
        IFNULL(Sum(Surrender_Count),0) as Surrender_Count
    FROM
        (
            -- Based on concepts from:
            -- https://stevestedman.com/2013/06/recursive-cte-for-dates-in-a-year/
            WITH RECURSIVE dates as (
                SELECT date_add(CURDATE(), interval -12 month) as dt -- start
                UNION ALL
                SELECT date_add(dt, interval 1 month) as dt
                FROM dates
                WHERE dt < CURDATE() -- end
            ) SELECT EXTRACT(YEAR_MONTH FROM dt) as yr_month
            FROM dates
            WHERE dt between date_add(CURDATE(), interval -12 month) and date_add(CURDATE(), interval -1 month)
        ) dates LEFT JOIN
        (
            SELECT Yr_Month, Species, Breed_Name, Adoption_Count, Surrender_Count
            FROM
                (SELECT
                    Yr_Month, Species, Breed_Name,
                    0 as Adoption_Count, count(*) as Surrender_Count
                FROM (SELECT
                    EXTRACT(YEAR_MONTH FROM a.Surrender_Date) as Yr_month,
                    Species,
                    GROUP_CONCAT(Breed_Name
                        ORDER BY Breed_Name SEPARATOR '/') as Breed_Name
                    FROM Animal AS a
                    INNER JOIN AnimalBreeds breeds on a.Pet_ID=breeds.Pet_ID
                    WHERE a.Surrender_Date is not null
                    GROUP BY a.Pet_ID, Species, EXTRACT(YEAR_MONTH FROM a.Surrender_Date)
                ) Animals
                GROUP BY Yr_Month, Species, Breed_Name
            ) Surrender UNION
            (
                SELECT
                    Yr_Month, Species, Breed_Name,
                    count(*) as Adoption_Count, 0 as Surrender_Count
                FROM (
                    SELECT
                        EXTRACT(YEAR_MONTH FROM a.Adoption_Date) as Yr_month,
                        Species,
                        GROUP_CONCAT(Breed_Name
                            ORDER BY Breed_Name SEPARATOR '/') as Breed_Name
                        FROM Animal AS a
                        INNER JOIN AnimalBreeds AS breeds on a.Pet_ID=breeds.Pet_ID
                        WHERE a.Adoption_Date is not null
                        GROUP BY a.Pet_ID, Species, EXTRACT(YEAR_MONTH FROM a.Adoption_Date)
                    )
                )
            )
        )
    )

```

```

) Animals
GROUP BY Yr_Month, Species, Breed_Name
)
) t2 on dates.yr_month=t2.yr_month
GROUP BY Yr_Month, Species, Breed_Name WITH ROLLUP
ORDER BY Grouping(dates.Yr_Month), dates.Yr_Month, Grouping(Species),
Species, Grouping(Breed_Name), Breed_Name
) report;

```

- Display table header with following columns of Month, Species, Breed, Number of surrenders, and Number of adoptions
- For each \$row in rows returned from database, display \$row.Yr_Month, \$row.Species, \$row.Breed_Name, \$row.Surrender_Count, and \$row.Adoption_Count

View Animal Control Report

Abstract Code - View Animal Control Report

Will display one of two things, the number of animals surrounded by them in a given month, and also any animals adopted in that month that were in the rescue for more than 60 days.

Goal is to meet auditors qualifications

When Report Active:

- Include: Current Month and Previous 6 months
- Execute query on RDBMS:

```

SELECT
  dates.Yr_Month,
  SUM(CASE
    WHEN Surrender_By_Animal_Control=1 AND
      EXTRACT(YEAR_MONTH FROM a.Surrender_Date)=dates.yr_month THEN 1
    ELSE 0 END) as Surrender_By_Animal_Control_Count,
  SUM(CASE
    WHEN EXTRACT(YEAR_MONTH FROM a.Adoption_Date)=dates.yr_month THEN 1
    ELSE 0 END) as Rescue_Over_60_Count
FROM
(
  -- Based on concepts from:
  -- https://stevestedman.com/2013/06/recursive-cte-for-dates-in-a-year/
  WITH RECURSIVE dates as (
    SELECT date_add(CURDATE(), interval -6 month) as dt -- start
    UNION ALL
    SELECT date_add(dt, interval 1 month) as dt
    FROM dates
    WHERE dt < CURDATE() -- end
  )
  SELECT EXTRACT(YEAR_MONTH FROM dt) as yr_month

```

```

FROM dates
WHERE dt between date_add(CURDATE(), interval -5 month) and CURDATE()
) dates
inner join Animal a on
  (EXTRACT(YEAR_MONTH FROM a.Surrender_date)=dates.yr_month AND
   a.Surrender_By_Animal_Control=1)
OR
  (EXTRACT(YEAR_MONTH FROM a.Adoption_Date)=dates.yr_month AND
   DATEDIFF(a.Adoption_Date,a.Surrender_Date)>=60)
group by dates.Yr_Month;

```

- For each row, display \$row.Yr_Month,
\$row.Surrender_By_Animal_Control_Count, and \$row.Rescue_Over_60_Count
 - Each row will have a drill down button
 - Month will be in a variable called \$Yr_Mon
- When the drill down button is selected,
 - Execute the following query on the RDBMS for the adopted animals waiting 60 or more days for rescue within the given month.

```

SELECT
  a.Pet_ID, Species,
  GROUP_CONCAT(Breed_Name ORDER BY Breed_Name SEPARATOR '/') as Breed_Name,
  Alteration_Status, Microchip_ID, Sex, Surrender_Date, Adoption_Date,
  DATEDIFF(a.Adoption_Date,a.Surrender_Date) as Days_To_Rescue
FROM Animal AS a
INNER JOIN AnimalBreeds AS breeds on a.Pet_ID=breeds.Pet_ID
WHERE
  DATEDIFF(a.Adoption_Date,a.Surrender_Date)>=60 AND
  EXTRACT(YEAR_MONTH FROM a.Adoption_Date)='$Yr_Mon'
GROUP BY
  a.Pet_ID, Species, Alteration_Status, Microchip_ID, Sex,
  Surrender_Date, Adoption_Date
ORDER BY DATEDIFF(a.Adoption_Date,a.Surrender_Date) DESC;

```

Execute the following query on the RDBMS for the surrendered by animal control within the given month.

```

SELECT
  a.Pet_ID, Species,
  GROUP_CONCAT(Breed_Name ORDER BY Breed_Name SEPARATOR '/') as Breed_Name,
  Alteration_Status, Microchip_ID, Sex, Surrender_Date
FROM Animal AS a
INNER JOIN AnimalBreeds AS breeds on a.Pet_ID=breeds.Pet_ID
WHERE
  Surrender_By_Animal_Control=1 AND
  EXTRACT(YEAR_MONTH FROM a.Surrender_Date)='$Yr_Mon'
group by a.Pet_ID, Species, Alteration_Status, Microchip_ID, Sex, Surrender_Date
order by a.Pet_ID;

```


- Information that will be presented will include
 - PetID, species, breed (alphabetically listed array), sex, alteration status, microchip ID, surrender date.
 - If animal has been in rescue for more than 60 days or more before adoption:
 - Display number of days they have been in rescue
- Information will be sorted based off the following criteria
 - If “animal control surrenders”:
 - sort by pet ID ascending
 - If “listing of animals over 60 days before being adopted”:
 - Sort by number of days in rescue descending

Display Volunteer of the Month

Abstract Code

- Find distinct month and year combinations with volunteer hours in database

```
SELECT DISTINCT EXTRACT(YEAR_MONTH FROM hours.Date)
FROM Volunteer AS v
INNER JOIN VolunteerHours AS hours on hours.Username=v.Username;
```

- For each row returned from the database, populate Drop Down List.
- Make most recent month the current selection in Drop Down List.
- When **Run** button is pressed, collect \$selectedMonth
- Execute query on RDBMS:

```
SELECT
  u.First_Name, u.Last_Name, u.Email_Address, sum(Hours) as Hours
FROM Volunteer AS v
INNER JOIN Users AS u on v.Username = u.Username
INNER JOIN VolunteerHours AS hours on hours.Username=v.Username
WHERE
  EXTRACT(YEAR_MONTH FROM hours.Date) = '$selectedMonth'
GROUP BY
  u.First_Name, u.Last_Name, u.Email_Address
ORDER BY
  u.Last_Name, u.First_Name;
```

- Display table header with following columns of First Name, Last Name, Email, and Number of Hours Volunteered
- For each \$row of the rows returned from database, display \$row.First_Name, \$row.Last_Name, \$row.Email_Address, and \$row.Hours volunteered
 - If sequential rows have the different \$Hours, \$TopCount ++
 - If \$TopCount == 5 then exit