

Table of Contents

Table of Contents	1
Login	2
Dog Dashboard	2
View Dog Details	6
Add Expense	8
Add Adoption	9
Add Dog	10
Add Adoption Application	12
Adoption Application Review (Admin Only)	14
Reports (Admin Only)	15

Login

Abstract Code

- User enters *email* ('\$email'), *password* ('\$password') input fields
- If both *username* and *password* input fields are filled:
 - When **Enter** button is clicked:

```
SELECT password
FROM `User`
WHERE email = '$email'
```

- If User record is NOT found
 - Go back to Login form with error message
- If User record is found but `User.Password != '$password'`:
 - Go back to Login form with error message
- Else:
 - Store login information as session variable '\$email'.
 - Go to Dog Dashboard Page
- Else *Email* and *password* input fields are NOT filled, display Login form with an error message

Dog Dashboard

Abstract Code

- Upon successful login:
 - Display all dogs currently in the shelter. For each dog, show the following information:
 - `Dog.name`
 - `Dog.age`
 - `Dog.sex`
 - `Dog.alteration_status`
 - `DogBreed.breed_name`
 - `Dog.adoptability_status`. This is derived from `Dog.alteration_status`.
 - If `Dog.alteration_status` is "altered", then the system automatically populates "adoptable".
 - If `Dog.alteration_status` is "unaltered", then the system automatically populates "not adoptable".

```
SELECT Dog.name, Dog.age, Dog.sex, Dog.alteration_status,
CASE
    WHEN Dog.alteration_status = 'altered' THEN 'adoptable'
    ELSE 'not adoptable'
END AS adoptability_status,
ConBreed.ConcatBreeds AS breed_name
FROM Dog INNER JOIN
(SELECT dogID, GROUP_CONCAT (DISTINCT breed_name ORDER BY breed_name
SEPARATOR '/') as ConcatBreeds
FROM DogBreed GROUP BY dogID) AS ConBreed
ON Dog.dogID = ConBreed.dogID
INNER JOIN Surrender
ON Dog.dogID = Surrender.dogID
WHERE Dog.dogID not in (SELECT dogID FROM Adoption)
ORDER BY Surrender.date;
```

- Display the **number of available spaces**. The information is derived from counting the number of dogs currently in the shelter.
 - The number of currently available spaces in the shelter = 15 - the number of dogs currently in the shelter
 - At any time, $0 \leq \text{the number of available spaces} \leq 15$
 - If the number of available spaces = 0, the “**Add Dog**” button will be disabled.
 - If the number of available spaces > 0, the “**Add Dog**” button will be clickable.

```
SELECT 15-((SELECT COUNT(*) FROM Dog) - (SELECT COUNT(*) FROM Adoption)) AS
Difference;
```

- Show “**Filter Dogs**”, “**Sort Dogs**”, “**Add Dog**”, “**Add Adoption Application**”, “**Adoption Application Review (Admin Only)**” and “**Reports (Admin Only)**” tabs.
- Upon:
 - Click on a **Dog** in the dog list - Jump to **View Dog Details** task.
 - Click **Filter Dogs** button -User select an option of **Adoption Status** from dropdown menu: Adoptable / Not Adoptable
 - If ‘\$Adoptable’ is selected, the **Dog Dashboard** will show the dogs whose Adoptability Status is “adoptable”.

```
SELECT Dog.name, Dog.age, Dog.sex, Dog.alteration_status,
CASE
```

```

        WHEN Dog.alteration_status = 'altered' THEN 'adoptable'
        ELSE 'not adoptable'
    END AS adoptability_status,
    ConBreed.ConcatBreeds AS breed_name
FROM Dog INNER JOIN
(SELECT dogID, GROUP_CONCAT(DISTINCT breed_name ORDER BY breed_name
SEPARATOR '/') AS ConcatBreeds
FROM DogBreed
GROUP BY dogID) AS ConBreed
ON Dog.dogID = ConBreed.dogID
INNER JOIN Surrender
ON Dog.dogID = Surrender.dogID
WHERE Dog.dogID not in (SELECT dogID FROM Adoption)
AND alteration_status = 'altered'
ORDER BY Surrender.date;

```

- If '\$Not Adoptable' is selected, the **Dog Dashboard** will show the dogs whose Adoptability Status is "not adoptable".

```

SELECT Dog.name, Dog.age, Dog.sex, Dog.alteration_status,
CASE
    WHEN Dog.alteration_status = 'altered' THEN 'adoptable'
    ELSE 'not adoptable'
END AS adoptability_status,
ConBreed.ConcatBreeds AS breed_name
FROM Dog INNER JOIN
(SELECT dogID, GROUP_CONCAT(DISTINCT breed_name ORDER BY breed_name
SEPARATOR '/') AS ConcatBreeds
FROM DogBreed
GROUP BY dogID) AS ConBreed
ON Dog.dogID = ConBreed.dogID
INNER JOIN Surrender
ON Dog.dogID = Surrender.dogID
WHERE Dog.dogID not in (SELECT dogID FROM Adoption)
AND alteration_status != 'altered'
ORDER BY Surrender.date;

```

- Run the **Filter Dogs** task: query for information about dogs based on the selected Adoptability Status.

- Click **Sort Dogs** button - User can select a sorting criterion.
 - By default, sort all dogs by **Surrender.Date**.

```
SELECT Dog.name, Dog.age, Dog.sex, Dog.alteration_status,
CASE
    WHEN Dog.alteration_status = 'altered' THEN 'adoptable'
    ELSE 'not adoptable'
END AS adoptability_status,
ConBreed.ConcatBreeds AS breed_name
FROM Dog INNER JOIN
(SELECT dogID, GROUP_CONCAT (DISTINCT breed_name ORDER BY
breed_name SEPARATOR '/') as ConcatBreeds
FROM DogBreed
GROUP BY dogID) AS ConBreed
ON Dog.dogID = ConBreed.dogID
INNER JOIN Surrender
ON Dog.dogID = Surrender.dogID
WHERE Dog.dogID not in (SELECT dogID FROM Adoption)
ORDER BY Surrender.Date ASC;
```

- Display sorted information based on the selected attribute in ascending order.
- Only can be sorted by one attribute at a time.

```
SELECT Dog.name, Dog.age, Dog.sex, Dog.alteration_status,
CASE
    WHEN Dog.alteration_status = 'altered' THEN 'adoptable'
    ELSE 'not adoptable'
END AS adoptability_status,
ConBreed.ConcatBreeds AS breed_name
FROM Dog INNER JOIN
(SELECT dogID, GROUP_CONCAT (DISTINCT breed_name ORDER BY
breed_name SEPARATOR '/') as ConcatBreeds
FROM DogBreed
GROUP BY dogID) AS ConBreed
ON Dog.dogID = ConBreed.dogID
INNER JOIN Surrender
ON Dog.dogID = Surrender.dogID
WHERE Dog.dogID not in (SELECT dogID FROM Adoption)
ORDER BY '$SelectedAttribute' ASC;
```

- Click **Add Dog** button - Jump to the **Add Dog** form.
- Click **Add Adoption Application** button - Jump to **Add Adoption Application** form.

- Click **Adoption Application Review (Admin Only)** button - Jump to **Adoption Application Review (Admin Only)** task.
- Click **Reports (Admin Only)** button - Jump to **Reports (Admin Only)** task.
- Click **Log Out** button - Invalidate login session and go back to the **Login** form.

View Dog Details

Abstract Code

- User clicks on **View Dog Details** button from **Dog Dashboard**. The **View Dog Details** screen is divided into 3 sections: **Dog Details**, **Expense Details** and **Add Adoption**.
- Run the **Dog Details** task: query for information about the **Dog** that was clicked on. Display the following information for the selected **Dog**:
 - **Dog.dogID**
 - **Microchip.microchipID**
 - **Dog.description**
 - **Dog.name**
 - **Dog.age**
 - **Dog.sex**
 - **Dog.alteration_status**
 - **Dog.adoptability_status**. This is derived from **Dog.alteration_status**.
 - If **Dog.alteration_status** is “altered”, then the system automatically populates “adoptable”.
 - If **Dog.alteration_status** is “unaltered”, then the system automatically populates “not adoptable”.

```
SELECT Dog.dogID, microchipID, description, name, age, sex, alteration_status,
CASE
  WHEN alteration_status = 'altered' THEN 'adoptable'
  ELSE 'not adoptable'
END AS adoptability_status
FROM Dog, Microchip
WHERE Dog.dogID = '$dogID'
and Dog.dogID = Microchip.dogID ;
```

- **DogBreed.breed_name**

```
SELECT breed_name
FROM DogBreed
WHERE dogID = '$dogID'
ORDER BY breed_name;
```

- **Edit** button is shown in the **Dog Details** section that allows a user to edit the following attributes of the selected **Dog**:

- If **Dog**.sex is “unknown”, the **Dog**.sex field is editable.
- If **Microchip**.microchipID is null, the **Microchip**.microchipID field is editable.
- If **Dog**.alteration_status is “unaltered”, the **Dog**.alteration_status field is editable.

```
UPDATE Dog
SET sex = '$sex', alteration_status = '$alteration_status'
WHERE dogID = '$dogID';

INSERT INTO Microchip VALUES( '$microchipID', '$dogID');
```

- If **DogBreed**.breed_name is “unknown” or “mixed”, the **DogBreed**.breed_name field is editable from a list from **Breed**.breed_name.

```
SELECT breed_name
FROM Breed;

UPDATE DogBreed
SET breed_name = '$breed_name'
Where dogID = '$dogID';

-- If a dog has more than 1 breed, add the other attributes to the table.
INSERT INTO DogBreed (dogID, breed_name)
VALUES ( '$dogID', '$breed_name');
```

- **Save** button is shown in the **Dog Details** section that allows a user to save the changes for the selected **Dog**.
- Run the **Expense Details** task: query for information about the expenses associated with the **Dog** that was clicked on. Under the **Expense Details** section, for each expense, show the following information:
 - **Expense**.date
 - **Expense**.description
 - **Expense**.amount
 - **Vendor**.name

```
SELECT date, description, amount, vendor_name
FROM Expense
WHERE dogID = '$dogID';
```

- Display **Add Expense** button that allows a user to add an [Expense](#) under the **Expense Details** section.
 - Click on the **Add Expense** button - jump to **Add Expense** form.
- **Add Adoption** button is shown at the bottom of the screen.
 - If Adoptability Status is “not adoptable”, the **Add Adoption** button is disabled.
 - If Adoptability Status is “adoptable”, clicking on the **Add Adoption** button will open up the **Add Adoption** form.

Add Expense

Abstract Code

- User clicks on **Add Expense** button from **View Dog Details**:
- User enters the following information on the form:
 - Automatically populates (`'$Dog.dogID'`) input field with the selected dog's ID.
 - Enter (`'$Vendor.name'`) input field.
 - Enter (`'$Expense.date'`) input field.
 - Enter (`'$Expense.description'`) input field.
 - Enter (`'$Expense.amount'`) input field.

```
INSERT INTO Expense (dogID, vendor_name, date, amount, description)
VALUES ('$Dog.dogID', '$Vendor.name', DATE '$Expense.date', '$Expense.amount',
'$Expense.description');
```

- Show **Save** button on the form.
 - If `'$Dog.Id'`, `'$Expense.date'` and `'$Vendor.name'` match with one of the expenses that has been recorded in the database, clicking on the **Save** button will display an error message “Duplicated expense - a dog can incur only one expense associated with one vendor at a given date.”
 - Else, clicking on the **Save** button will add this new expense into the database.
 - If `'$Vendor.name'` doesn't match with any [Vendor.name](#) in the [Vendor](#) table, clicking on the **Save** button will add `'$Vendor.name'` to the [Vendor](#) table with a message “A new vendor has been added.”

```
SELECT name
FROM Vendor;

INSERT INTO Vendor (name)
VALUES ('$Vendor.name');
```


Add Adoption

Abstract Code

- User clicks on **Add Adoption** button from **View Dog Details**:
- Run the **Add Adoption** task: query for information about the application and the associated applicant information.
 - Find eligible adoption application by searching applicant's last name and the co-applicant's last name (case insensitive, partial match allowed):
 - Display the following information for each applicant shown in the search result:
 - `Applicant.last_name`
 - `Applicant.first_name`
 - `Applicant.phone_number`
 - `Applicant.email`
 - `Applicant.street`
 - `Applicant.city`
 - `Applicant.zip_code`
 - `Applicant.state`
 - Display co-applicant's Last Name if match is made by co-applicant last name.

```
SELECT DISTINCT Applicant.last_name, Applicant.first_name,
Applicant.phone_number, Applicant.email,
Applicant.street, Applicant.city, Applicant.zip_code, Applicant.state,
Application.co_applicant_last_name
FROM Applicant, Application
WHERE Application.applicant_email = Applicant.email
AND ((Applicant.last_name like '%$last_name%')
OR (Application.co_applicant_last_name like '%$last_name%'))
AND Application.applicationID in (SELECT applicationID FROM Approved);
```

- Choose an adopter; display that adopter's most recent approved **Adoption Application**.

```
SELECT Application.applicationID, Application.applicant_email, date,
co_applicant_first_name,
co_applicant_last_name, 'Approved' AS status
```

```
FROM Application, Applicant, Approved
WHERE Application.applicant_email = Applicant.email
AND Approved.applicationID = Application.applicationID
AND Application.applicant_email = '$Email'
ORDER BY date DESC
LIMIT 1;
```

- Calculate adoption fee for Applicant; display Adoption Fees.
 - If Surrender.animal_control is “yes”, then Adoption.fee equals the sum of Expense for this dog multiplied by 1.15.
 - If Surrender.animal_control is “no”, then Adoption.fee equals the sum of Expense for this dog multiplied by 0.15.

```
SELECT animal_control,
CASE
    WHEN animal_control = true THEN (SELECT SUM(amount)*1.15
FROM Expense WHERE dogID = '$dogID')
    ELSE (SELECT SUM(amount)*0.15 FROM Expense WHERE dogID =
'$dogID')
END AS Fees
FROM Surrender
WHERE dogID = '$dogID';
```

- Add Adoption Date
- **Submit Adoption** Button will save the entered information to the Adoption table.

```
INSERT INTO Adoption (dogID, applicationID, date, fee, admin_email)
VALUES ( '$dogID', '$applicationID', DATE '$date', '$fee', '$admin_email');
```

Add Dog

Abstract Code

- User clicks on **Add Dog** button from **Dog Dashboard**:
- The following information of a dog need to be entered:
 - System automatically populates ('\$Dog.dogID') in the input field.

- The assigned '\$Dog.dogID' equals the last dogID saved in the Dog table plus 1. The value will be saved as the new Dog.dogID.
- Enter ('\$Microchip.microchipID') input field only if a dog has a microchipID and it will be saved as Microchip.microchipID.
 - Only when the Microchip ID is provided and the value doesn't exist in the Microchip table, the value will be added into the Microchip table.
- Enter ('\$Dog.description') input field which will be saved as Dog.description.
 - String limited to 200 characters.
- Enter ('\$Dog.name') input field which will be saved as Dog.name.
 - String limited to 20 characters.
- Enter ('\$Dog.sex') input field which will be saved as Dog.sex.
 - User must choose from one of the 3 options: "male", "female", or "unknown".
- Enter ('\$Dog.alteration_status') input field which will be saved as Dog.alteration_status.
 - User must choose from one of the 2 options: "unaltered" or "altered".
- Enter ('\$Dog.age') input field which will be saved as Dog.age.
 - The age must be recorded in months.

```
INSERT INTO Dog (name, sex, alteration_status, age, description)
VALUES ('$Dog.name', '$Dog.sex', '$Dog.alteration_status', '$Dog.age',
'$Dog.Description' );

SET @dogID = LAST_INSERT_ID();

INSERT INTO Microchip VALUES (@dogID, '$Microchip.microchipID');
```

- Enter ('\$DogBreed.breed_name') input field where multiple breed names can be selected from the Breed table. They will be saved as DogBreed.breed_name.
 - The system populates available options to choose from.
 - If "unknown" or "mixed" is selected, none of the other options can be selected.
 - If "unknown" or "mixed" is not selected, one or multiple values from the predefined Breed table can be selected.
 - If multiple breeds are selected, they should be concatenated into a single value, delimited with a forward slash (/). The combination of breed names should be in alphabetical order.

```
INSERT INTO DogBreed VALUES
(@dogID, '$DogBreed.breed_name1'), (@dogID, '$DogBreed.breed_name2');
```

- Enter ('**\$Surrender.reason**') input field which will be saved as **Surrender.reason**.
 - String limited to 200 characters.
- Enter ('**\$Surrender.date**') input field which will be saved as **Surrender.date**.
 - Enter in date time format.
- Enter ('**\$Surrender.animal_control**') input field which will be saved as **Surrender.animal_control**.
 - User must choose from one of the 2 options: "true" or "false".
- System automatically populates ('**User.email**') in the input field based on the User Login information.

```
INSERT INTO Surrender (dogID, email, date, reason, animal_control)
VALUES (@dogID, '$User.email', DATE '$Surrender.date', '$Surrender.reason',
'$Surrender.animal_control');
```

- For auditing purposes, the system needs to keep track of which user originally entered the dog's information.
- Show **Save** button on the form.
 - If '**\$DogBreed.breed_name**' is "bulldog" and '**\$Dog.name**' is "Uga", clicking on the **Save** button will display an error message "Please enter a different name."
 - If any fields other than '**\$Microchip.microchipID**' is null, clicking on the **Save** button will display an error message "Please enter the required information."
 - Else, clicking on the **Save** button will add this new dog into the database.
- Show **View Dog Details** button on the form.
 - Upon click, jump to the **View Dog Details** screen.
- Show **Dog dashboard** button on the form.
 - Upon click, jump to the **Dog Dashboard** screen.

Add Adoption Application

Abstract Code

- User clicks on **Add Adoption Application** button from **Dog Dashboard**:
- Display **Applicant Form**;
- Enter ('**\$email**') input field in **Applicant Form**;
- Search the applicant by matching the '**\$email**' with the **Applicant.email** in **Applicant** table.

```
SELECT first_name, last_name, street, city, state, zip_code, phone_number
```

```
FROM Applicant
WHERE email = '$email';
```

- If '\$Email' matches one of the Applicant.email in Applicant table:
 - Autofill **Applicant Form** and display the following information for the existing applicant:
 - Applicant.first_name
 - Applicant.last_name
 - Applicant.street
 - Applicant.city
 - Applicant.state
 - Applicant.zip_code
 - Applicant.phone_number
- If '\$Email' does not match with any Applicant.email in Applicant table:
 - User needs to create a new applicant by entering the following information on the **Applicant Form**:
 - Enter ('\$Applicant.email') input field.
 - Enter ('\$Applicant.first_name') input field.
 - Enter ('\$Applicant.last_name') input field.
 - Enter ('\$Applicant.street') input field.
 - Enter ('\$Applicant.city') input field.
 - Enter ('\$Applicant.state') input field.
 - Enter ('\$Applicant.zip_code') input field.
 - Enter ('\$Applicant.phone_number') input field.

```
INSERT INTO Applicant (email, first_name, last_name, phone_number, street, city, zip_code, state)
VALUES ('$Applicant.email', '$Applicant.first_name', '$Applicant.last_name',
'$Applicant.phone_number', '$Applicant.street', '$Applicant.city', '$Applicant.state',
'$Applicant.zip_code');
```

- Show **Save** button on the form.
 - Click on **Save** button to add the new applicant's information into the Applicant table.
- Display **Application Form** after the applicant's information is found or entered.
- User enters the following information on the **Application Form**:
 - Enter ('\$Application.date') input field.
 - Enter ('\$Application.co_applicant_first_name') input field. Can be null.
 - Enter ('\$Application.co_applicant_last_name') input field. Can be null.
- Show **Submit** button on the form.
 - Click on **Submit** button will add the new application into the Application table.

- System automatically populates ('\$Application.application_number') in the input field and displays it on the screen. The value will be saved as Application.application_number in the Application table.

```
INSERT INTO Application (applicant_email, date,
co_applicant_first_name,co_applicant_last_name, user_email)
VALUES ('$Application.applicant_email', DATE '$Application.date',
'$Application.co_applicant_first_name', '$Application.co_applicant_last_name',
'$Application.user_email');

SELECT LAST_INSERT_ID() AS application_number;
```

Adoption Application Review (Admin Only)

Abstract Code

- User clicks on **Adoption Application Review (Admin Only)** button from **Dog Dashboard:**
- Display all **Application** not in **Approved** and **Rejected** table with following information:
 - Application.applicationID
 - Application.date
 - Application.co_applicant_first_name
 - Application.co_applicant_last_name
 - Applicant.email
 - Applicant.first_name
 - Applicant.last_name
 - Applicant.street
 - Applicant.city
 - Applicant.state
 - Applicant.zip_code
 - Applicant.phone_number

```
SELECT Application.applicationID, Application.date, Application.co_applicant_first_name,
Application.co_applicant_last_name, Applicant.email, Applicant.first_name,
Applicant.last_name, Applicant.street, Applicant.city, Applicant.state, Applicant.zip_code,
Applicant.phone_number
FROM Application, Applicant
WHERE Application.applicant_email = Applicant.email
AND Application.applicationID NOT IN (SELECT applicationID FROM Approved)
AND Application.applicationID NOT IN (SELECT applicationID FROM Rejected);
```

- Admin must choose from one of the 2 options: “Approve” and “Reject”. If “Approve” is selected, add applicationID into [Approved](#) table. If “Reject” is selected, add applicationID into [Rejected](#) table.

```
INSERT INTO Approved VALUES ('$applicationID');  
  
INSERT INTO Rejected VALUES ('$applicationID');
```

- Show **Save** button on the form.
 - Click on **Save** button to save the updated [Approved](#) and [Rejected](#).
- Refresh the **Adoption Application Review** Form.

```
SELECT Application.applicationID, Application.date, Application.co_applicant_first_name,  
Application.co_applicant_last_Name, Applicant.email, Applicant.first_name,  
Applicant.last_name, Applicant.street, Applicant.city, Applicant.state, Applicant.zip_code,  
Applicant.phone_number  
FROM Application, Applicant  
WHERE Application.applicant_email = Applicant.email  
AND Application.applicationID NOT IN (SELECT applicationID FROM Approved)  
AND Application.applicationID NOT IN (SELECT applicationID FROM Rejected);
```

Reports (Admin Only)

Abstract Code

- User clicks on **Reports (Admin Only)** button from **Dog Dashboard**:
- Show the following tabs: **View Animal Control Report**, **View Monthly Adoption Report**, **Expense Analysis**, and **volunteer Lookup**.
 - When click on **View Animal Control Report**, run the following queries:
 - Filter [Surrender](#).animal_control with value equals to true
 - Filter [Surrender](#).date to include current month (inclusive report date) and the previous six months
 - Groupby number of dogs surrendered by each month
 - For each month, report will separate into 2 categories:
 - Surrendered (sort by Dog ID ascending):
 - [Dog](#).dogID

- [DogBreed.breed_name](#) (If a dog has multiple breeds, the breeds should be concatenated into a single value delimited with a forward slash in alphabetical order.)
- [Dog.sex](#)
- [Dog.alteration_status](#)
- [Microchip.microchipID](#)
- [Surrender.date](#)
- Adopted (listed by number of days in rescue descending, followed by dog ID descending)
 - [Dog.dogID](#)
 - [DogBreed.breed_name](#) (If a dog has multiple breeds, the breeds should be concatenated into a single value delimited with a forward slash in alphabetical order.)
 - [Dog.sex](#)
 - [Dog.alteration_status](#)
 - [Microchip.microchipID](#)
 - [Surrender.date](#)
 - If dog's [Surrender.date](#) equals or greater than 60 from [Adoption.date](#), display the difference in Dates
 - Total [Expense.amount](#) for the [Expense.date](#) with adoption month

```
-- count of dogs turned over by animal control (animal control surrenders)
SELECT DATE_FORMAT(DATE_FORMAT(Surrender.date , '%Y-%m-01'), "%b %Y") AS
month, Count(Surrender.dogID) AS num_dogs_from_animal_control
FROM Surrender
WHERE animal_control = true
AND Surrender.date between DATE_SUB(DATE_FORMAT(NOW() , '%Y-%m-01'),
INTERVAL 6 MONTH) AND NOW()
GROUP BY DATE_FORMAT(DATE_FORMAT(Surrender.date , '%Y-%m-01'), "%b
%Y"), DATE_FORMAT(Surrender.date , '%Y-%m-01')
ORDER BY DATE_FORMAT(Surrender.date , '%Y-%m-01');

-- dogs adopted during that month total expense and count who had spent in the
rescue 60 days or more
SELECT DATE_FORMAT(M, "%b %Y") AS month,
      SUM(Is60DAYS * NumAdoptedDogs) AS
num_adopted_dog_from_animal_control_60days,
      SUM(TotalExpense) AS total_expense_from_animal_control_dogs
```



```

FROM (
SELECT DATE_FORMAT(Surrender.date, '%Y-%m-01') AS M,
      CASE WHEN DATEDIFF(Adoption.date, Surrender.date) >= 60 THEN 1 ELSE 0
END AS Is60DAYS,
      Count(DISTINCT Surrender.dogID) AS NumAdoptedDogs,
      SUM(Expense.amount) AS TotalExpense
FROM Surrender INNER JOIN Adoption
ON Surrender.dogID = Adoption.dogID
LEFT OUTER JOIN Expense
ON Surrender.dogID = Expense.dogID
WHERE animal_control = true
AND DATEDIFF(Adoption.date, Surrender.date) >= 60
AND Surrender.date BETWEEN DATE_SUB(DATE_FORMAT(NOW(), '%Y-%m-01'),
INTERVAL 6 MONTH) AND NOW()
GROUP BY DATE_FORMAT(Surrender.date, '%Y-%m-01'),
      CASE WHEN DATEDIFF(Adoption.date, Surrender.date) >= 60 THEN 1 ELSE 0
END) AS t
GROUP BY M
ORDER BY M;

```

```

-- dogs turned over by animal control (animal control surrenders)
SELECT Dog.dogID, ConBreed.ConcatBreeds AS breed_name, Dog.sex,
Dog.alteration_status, Microchip.microchipID, Surrender.date
FROM Dog INNER JOIN
(SELECT dogID, GROUP_CONCAT(DISTINCT breed_name ORDER BY breed_name
SEPARATOR '/') AS ConcatBreeds
FROM DogBreed GROUP BY dogID) AS ConBreed
ON Dog.dogID = ConBreed.dogID
INNER JOIN Surrender
ON Dog.dogID = Surrender.dogID
LEFT JOIN Microchip
ON Dog.dogID = Microchip.dogID
WHERE animal_control = true
AND DATE_FORMAT(Surrender.date, '%Y-%m-01') = '$LookupMonth'
ORDER BY Dog.dogID;

```

```

-- dogs turned over by animal control were adopted
SELECT Dog.dogID, ConBreed.ConcatBreeds AS breed_name, Dog.sex,
Dog.alteration_status, Microchip.microchipID, Surrender.date,

```

```

CASE WHEN DATEDIFF(Adoption.date, Surrender.date) >= 60 THEN
DATEDIFF(Adoption.date, Surrender.date) ELSE NULL END AS days_in_shelter
FROM Dog INNER JOIN
(SELECT dogID, GROUP_CONCAT(DISTINCT breed_name ORDER BY breed_name
SEPARATOR '/') AS ConcatBreeds
FROM DogBreed GROUP BY dogID) AS ConBreed
ON Dog.dogID = ConBreed.dogID
INNER JOIN Surrender
ON Dog.dogID = Surrender.dogID
INNER JOIN Adoption
ON Dog.dogID = Adoption.dogID
LEFT JOIN Microchip
ON Dog.dogID = Microchip.dogID
WHERE animal_control = true
AND DATE_FORMAT(Surrender.date, '%Y-%m-01') = '$LookupMonth'
ORDER BY DATEDIFF(Adoption.date, Surrender.date) DESC, Dog.dogID;

```

- When click on **View Monthly Adoption Report:**
 - Display Monthly Adoption Report
 - Filter DogBreed.breed_name by Surrender.date and Adoption.date equals last 12 months. This will generate a list of breeds that needs to be included for each month's report (sorted alphabetically).
 - Create a table with the following columns:
 - Month: derived from Surrender.date and Adoption.date.
 - Dog Breed: derived from filtered dog breed list.
 - Number of surrenders: derived from counting the number of surrenders of each filtered dog breed per month.
 - Number of adoptions: derived from count number of adoptions of each filtered dog breed per month.
 - Expenses: derived from sum of expenses of each filtered dog breed per month.
 - Adoption fees: derived from sum of adoption fees of each filtered dog breed per month.

- o Net expenses: difference of expenses and adoption fees of each filtered dog breed per month.
- Each row would be an unique representation of Month and one breed from the filtered list. (starting with January)
- In case a breed from the previous filtered result has no record for adoption or surrederation, should be default to 0 in the cell.
- Visual Representation of the Table for 3 filtered dog breeds:

Month	Dog Breed	NumOfSurre	NumOfAdop	Expense	AdoptionFee	NetExpense
Jan	Dog Breed #1	2	2	1000	500	500
Jan	Dog Breed #2	3	2	1000	600	400
Jan	Dog Breed #3	4	3	1000	700	300
Feb	Dog Breed #1	6	0	1000	0	1000
Feb	Dog Breed #2	0	0	0	0	0
Feb	Dog Breed #3	1	1	500	100	400
....
....

```

WITH
S AS (
    SELECT DATE_FORMAT(Surrender.date, '%Y-%m-01') AS
    Month, ConBreed.ConcatBreeds AS breed_name,
    Count(Surrender.dogID) AS NumDogsSurrendered
    FROM Surrender INNER JOIN
    (SELECT dogID, GROUP_CONCAT(DISTINCT breed_name
    ORDER BY breed_name SEPARATOR '/') AS ConcatBreeds
    FROM DogBreed
    GROUP BY dogID) AS ConBreed
    ON Surrender.dogID = ConBreed.dogID
    WHERE Surrender.date between
    DATE_SUB(DATE_FORMAT(NOW(), '%Y-%m-01'), INTERVAL 13
    MONTH) AND LAST_DAY(NOW()) - INTERVAL 1 MONTH)
    GROUP BY DATE_FORMAT(Surrender.date, '%Y-%m-01'),
    ConBreed.ConcatBreeds),
A AS (SELECT DATE_FORMAT(Adoption.date, '%Y-%m-01') AS Month,
    ConBreed.ConcatBreeds AS breed_name, Count(DISTINCT
    Adoption.dogID) AS NumDogsAdopted,
    SUM(Adoption.fee) AS TotalAdoptionFee,
    SUM(Expense.amount) AS TotalExpense, SUM(Adoption.fee)-
    SUM(Expense.amount) AS Profit

```

```

FROM Adoption INNER JOIN
(SELECT dogID, GROUP_CONCAT(DISTINCT breed_name
ORDER BY breed_name SEPARATOR '/') AS ConcatBreeds
FROM DogBreed
GROUP BY dogID) AS ConBreed
ON Adoption.dogID = ConBreed.dogID
LEFT OUTER JOIN Expense
ON Adoption.dogID = Expense.dogID
WHERE Adoption.date between
DATE_SUB(DATE_FORMAT(NOW(), '%Y-%m-01'), INTERVAL 13
MONTH) AND LAST_DAY(NOW()) - INTERVAL 1 MONTH)
GROUP BY DATE_FORMAT(Adoption.date, '%Y-%m-01'),
ConBreed.ConcatBreeds)
SELECT DATE_FORMAT(Month, "%b %Y") as month, breed_name as
breed_name, IFNULL(NumDogsSurrendered,0) AS
num_dogs_surrendered, IFNULL(NumDogsAdopted,0) AS
num_dogs_adopted, IFNULL(TotalAdoptionFee,0) AS
total_adoption_fee, IFNULL(TotalExpense,0) AS total_expense,
IFNULL(Profit,0) AS profit
FROM (
SELECT COALESCE(S.Month, A.Month) AS Month,
COALESCE(S.breed_name, A.breed_name) AS breed_name,
NumDogsSurrendered, NumDogsAdopted, TotalAdoptionFee,
TotalExpense, Profit FROM S
LEFT OUTER JOIN A ON S.Month = A.Month AND S.breed_name =
A.breed_name
UNION
SELECT COALESCE(S.Month, A.Month) AS Month,
COALESCE(S.breed_name, A.breed_name) AS breed_name,
NumDogsSurrendered, NumDogsAdopted, TotalAdoptionFee,
TotalExpense, Profit FROM S
RIGHT OUTER JOIN A ON S.Month = A.Month AND S.breed_name =
A.breed_name
) AS t
ORDER BY DATE(Month), breed_name;

```

- When click on **Expense Analysis**:

- List all the vendors in the **Vendor** table. For each vendor, calculate the total spent regardless of date and display the total.
- Sort the report by total in descending order.

```
SELECT Vendor.name, IFNULL(SUM(Expense.amount),0) AS total_expense
FROM Vendor LEFT JOIN Expense
ON Vendor.name = Expense.vendor_name
GROUP BY Vendor.name
ORDER BY SUM(Expense.amount) DESC;
```

○ When click on **Volunteer Lookup**:

- Search volunteers by **Volunteer.first_name** and/or **Volunteer.last_name** (case insensitive and allow partial matching)
- Display the following information for the matched volunteers:
 - **Volunteer.first_name**
 - **Volunteer.last_name**
 - **Volunteer.email**
 - **Volunteer.phone**
- Sort the results by **Volunteer.last_name** in ascending order and **Volunteer.first_name** in ascending order.

```
SELECT first_name, last_name, User.email, phone_number, start_date
FROM User
WHERE User.email not in (SELECT email FROM Admin)
AND (LOWER(User.first_name) LIKE '%$lookup_name%' OR
LOWER(User.last_name) LIKE '%$lookup_name%')
ORDER BY last_name, first_name;
```