Project Phase 2

Contents

Task Decomposition(TD) and Abstract Code(AC) for Tasks

Login

Abstract Code

- User enters *email* ('$Email'), *password* ('$Password') input fields.

- If data validation is successful for both username and password input fields, then:

  o When **Enter** button is clicked:

  o If User record is not found:

    > SELECT email FROM `User` WHERE email = '$Email' AND `password` = '$Password';

    - Go back to **Login** form, with error message.

  o Else

    - Check user type:

      > SELECT email FROM Volunteer WHERE email = '$Email';
      > SELECT email FROM ExecutiveDirector WHERE email = '$Email';
      > SELECT email FROM AdminUser WHERE email = '$Email';

      - If user type is Volunteer

        o If in SystemConfig table has SystemConfig.ConfigName = "loggedUser" and ConfigValue != '$Email' && ConfigValue != NULL

        > SELECT config_value FROM SystemConfig WHERE configName = 'loggedUser';

          - Go back to **Login** form, show error message (Only one volunteer user can log in at the same time).

        o Else

          - Store login information as session variable *Email* ('$Email').

          - Set the value of SystemConfig.ConfigValue = '$Email' for SystemConfig.ConfigName = "loggedUser"

          > UPDATE SystemConfig SET config_value = '$Email' WHERE configName = 'loggedUser';

- Go to **View Dashboard** form.

- If user type is ED or Admin

  o Go to **View Dashboard** form.

- Else email and password input fields are invalid, display **Login** form, with error message.

Logout

Abstract Code

- User clicks *Log Out* button on **Dog Dashboard**.

- If user type is Volunteer:

  ```
  SELECT email FROM Volunteer WHERE email = '$Email';
  ```
  o Set the value of SystemConfig.ConfigValue = NULL for SystemConfig.ConfigName = "loggedUser"

  ```
  UPDATE SystemConfig SET config_value = NULL WHERE configName = 'loggedUser';
  ```

- Clean login session information.

- Go to **Login** form.

View Dashboard

Task Decomp

Lock Types: Read only on Dog table and SystemConfig table.

Number of Locks: Two

Enabling Conditions: Trigger by successful login.

Frequency: Daily.

Consistency (ACID): not critical, order is not critical.

Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code

- Retrieve all dogs currently in the shelter from the Dog table.
- Display dog details in columns: Dog.DogID, Dog.Name, Dog.Breed, Dog.Sex, Dog.AlterationStatus, Dog.AgeWhenSurrender, Dog.AdoptionState, ordered by Dog.SurrenderDate (oldest to newest).
- Implement filter mechanism to filter dogs by Dog.AdoptionState.
- Create clickable list where clicking on a dog's name opens the **Dog Detail** form.
- If User clicks on a dog in the list:
  - Jump to the **View Dog Details** task.
- Calculate and display the number of currently available spaces in the shelter using the Dog and SystemConfig tables.
- Show **"Add Adoption Application"** tab.
- Upon clicking the **Add Adoption Application** button:
  - Jump to the **Add Adoption** task.
- If the shelter is not full:
  - Show **"Add Dog"** tab.
  - Upon clicking the **Add Dog** button:
    - Jump to the **Add Dog** task.
- Display **Log Out** button
- If User clicks **Log Out** button
  - Jump to the **Logout** task

View Dog Details

Abstract Code

- User selects a dog from the **Dog Dashboard**, system records *DogID* ('$DogID').

- Find dog where Dog.DogID = '$DogID', record Dog.SurrenderID as *SurrenderID('$SurrenderID')*.

```
SELECT name, sex, surrender_date, age_when_surrender, alteration_status,
description, surrenderID FROM Dog WHERE dogID = '$DogID';
```

- Display Dog.Name, Dog.Breed, Dog.Sex, Dog.AgeWhenSurrender, Dog.AlterationStatus, Dog.Description, and Dog.SurrenderDate.

```
SELECT breedname FROM DogBreed WHERE dogID = '$DogID';
```

- If the dog has a microchip (Dog.MicrochipID exists):

  - Display Dog.MicrochipID and MicrochipVendor.ManufacturedName.

    > SELECT Microchip.microchipID, MicrochipVendor.manufacture_name FROM Microchip JOIN MicrochipVendor ON Microchip.manufacture_name = MicrochipVendor.manufacture_name WHERE Microchip.dogID = '$DogID';

- Find expenses from Expense table, grouped by Expense.Category, display Expense.Date, Expense.VendorName, Expense.Amount.

  > SELECT
  >   Expense.expense_date AS Date,
  >   Expense.vendor_name AS VendorName,
  >   Expense.amount AS Amount,
  >   Expense.category AS Category
  > FROM Expense WHERE Expense.dogID = '$DogID'
  > ORDER BY Expense.category, Expense.expense_date;

- Calculate total expense per category.

  > SELECT
  >   Expense.category AS Category,
  >   SUM(Expense.amount) AS TotalExpense
  > FROM Expense
  > WHERE Expense.dogID = '$DogID' GROUP BY Expense.category;

- if exist Surrender.PhoneNumber where Surrender.SurrenderID = '$SurrenderID'

  - Display Surrender.PhoneNumber

    > SELECT phonenumber FROM Individual WHERE surrenderID = '$SurrenderID';
    >
    > SELECT phonenumber FROM LocalAnimalControlDepartment WHERE surrenderID = '$SurrenderID';

- Display Surrender.SurrenderType

- If the dog is adoptable and the user is the Executive Director, display the "***Add Adoption***" button.

  > SELECT email FROM ExecutiveDirector WHERE email = '$UserEmail';
  >
  > SELECT adoption_state FROM Dog WHERE dogID = '$DogID';

Update Dog Details

Abstract Code

- User clicks on the ***View Dog Details*** button on **Dog Dashboard**

- Form with prefilled dog info pops up (Use **View Dog Details** task).

- User can edit– *sex* ('$Sex') (if "unknown"), add *microchipID* ('$MicrochipID') (if not present) with selection of *Manufacture Name*('$ManufactureName'), add one or more *breed* ('$Breeds') (only if "Unknown" or "Mixed" are the current breed), and   update *alteration status* ('$AlterationStatus') (only if unaltered)

- Once *submit* button is clicked, the following queries to be executed to **update dog details.**

    o   If there is an update on *sex* or *alteration status* of the Dog

    ```
    UPDATE Dog
    SET Dog.sex = '$Sex'
    WHERE Dog.DogID = '$DogID';
    ```
    ```
    UPDATE Dog
    SET Dog.alteration_status = '$AlterationStatus'
    WHERE Dog.DogID = '$DogID';
    ```

    o   If there is an update on dog's *microchipID*

    ```
    INSERT INTO Microchip (microchipID, dogID, manufacture_name)
    VALUES('$MicrochipID', '$DogID', '$ManufactureName');
    ```

    o   If there is an update on dog's *breeds* and for each '$Breed' in '$Breeds'

    ```
    INSERT INTO DogBreed(dogID, breedname)
    VALUES('$DogID', '$Breed');
    ```

        ▪   After dog's *breeds* are updated, delete the 'Unknown' or 'Mixed' breeds for the dog

        ```
        DELETE FROM DogBreed
        WHERE (DogBreed.breedname = 'Unknown' OR
        DogBreed.breedname = 'Mixed') AND DogBreed.DogID = '$DogID';
        ```

Add Dog

Abstract Code

- User is logged in and entered **Dashboard.**

- If space is available (current number of dogs in shelter is less than shelter's maximum capacity)

    o   Display ***Add Dog*** button.

    o   Click ***Add Dog*** button.

- Jump to **Add Dog** Form page.

- User inputs for the following attributes for Dog:

    - *name* ('$name'), *sex* ('$sex'), *alterrationStatus* ('$alterrationStatus'), *ageWhenSurrendered* ('$ageWhenSurrendered') in months and breeds ('$breeds') from dropdown list(could be more than one breeds).

        - The content of dropdown list for '$breeds' is from the following query:

        ```
        SELECT breedname
        FROM Breed
        ```

        - Note that above attributes cannot be empty.

    - *description* ('$description'), *MicrochipID* ('$MicrochipID') and *MicrochipVendor* ('$MicrochipVendor') from dropdown list.

        - The content of dropdown list for '$MicrochipVendor' is from the following query:

        ```
        SELECT manufacture_name
        FROM MicrochipVendor
        ```

        - Note that above attribute could be empty.

- User inputs for the following attributes for Surrender:

    - Choose *surrenderType*('$surrenderType') from dropdown list. Its either *Individual* or *Local Animal Control Department*

    - *phoneNumber* ('$phoneNumber')

        - Phone number cannot be empty if surrenderType selected as Local Animal Control Department.

- When **Add** button is clicked after inputs.

- Check the validity of all input attributes.

    - If '$MicrochipID' is not empty, perform following query

    ```
    SELECT COUNT(dogID) as result
    FROM Microchip
    WHERE Microchip.microchipID = '$MicrochipID';
    ```

    If result > 0, mark input attribute '$MicrochipID' as invalid with error message of MicrochipID is not unique.

- If all input attributes are valid.

    - Run subtask task **Add Surrender** and get '$surrenderID'

    - Insert new Dog into database using inputs attributes listed above and Dog.SurrenderDate = $Today's date', Dog.AdoptionState = 'False'(*default*).

    ```
    INSERT INTO Dog(name, sex, surrender_date,
    age_when_surrender, alteration_status, description,
    adoption_state, surrenderID)
    VALUES('$name', '$sex', '$Today's date',
    '$ageWhenSurrendered', '$alterrationStatus', '$description',
    'False', '$surrenderID');
    ```
    '$DogID' is auto generated by database after insertion.

    - For each '$Breed' in '$Breeds'.

    ```
    INSERT INTO DogBreed(dogID, breedname)
    VALUES('$DogID', '$Breed');
    ```

    - If '$MicrochipID' is not empty.

    ```
    INSERT INTO Microchip (microchipID, dogID,
    manufacture_name)
    VALUES('$MicrochipID', '$DogID', '$MicrochipVendor');
    ```

    - Go to **View Dog Details.**

- Else

    - Go back to **Add Dog** form and display error messages for those invalid attributes.

Add Surrender

Abstract Code

- Run mother task **Add Dog** with all valid input attributes from user.

- With input attributes '$surrenderType' and '$phoneNumber', insert new Surrender into Surrender database. '$surrenderID' is auto generated after insertion.

```
INSERT INTO Surrender ()
VALUES ();
```

- If '$surrenderType' is 'Individual'

```
INSERT INTO Individual (surrenderID, phonenumber)
VALUES ('$surrenderID', '$phoneNumber');
```

- If '$surrenderType' is 'LocalAnimalControlDepartment'

```
INSERT INTO LocalAnimalControlDepartment (surrenderID, phonenumber)
VALUES ('$surrenderID', '$phoneNumber');
```

- Go back to mother task **Add Dog** with generated '$surrenderID'.

Add Adoper

Abstract Code

- User logs in as Executive Director.

- Click on "*Add Adopter*" button.

- Display a form to enter adopter details: '$firstname', '$lastname', '$phonenumber', '$household_size', '$email' and adopter's address consisting of '$street', '$city', '$state' and '$zipcode'.

- Validate the fields.

  o Check Attributes of '$firstname', '$lastname', '$phonenumber', '$household_size', '$email', '$street', '$city', '$state' and '$zipcode' which cannot be empty values.

  o Check the uniqueness of '$email'.

  ```
  SELECT COUNT(email) as result
  FROM Adopter
  WHERE Adopter.email = '$email';
  ```
  If result > 0, the input '$email' is not unique and mark validation as failed with error message.

- If validation succeeds

  o Insert a new record into the Adopter table.

  ```
  INSERT INTO Adopter (email, phone_number, household_size, firstname, lastname, street, city, state, zipcode)
  VALUES ('$email', '$phonenumber', '$household_size', '$firstname', '$lastname', '$street', '$city', '$state', '$zipcode');
  ```
  o Display a confirmation message: "Adopter added successfully.

- If validation failed

o   Display the error message.

View Adoption Application

Abstract Code:

- User is logged in and role is Executive Director.

- *View Adoption Application* button is clicked.

- Find all **Adoption Applications** who are *PendingApproval*.

```
SELECT A.email, A.application_date
FROM AdoptionApplication A
WHERE NOT EXISTS (
        SELECT 1 FROM RejectedApplication R
        WHERE A.email = R.email AND A.application_date = R.application_date
)
AND NOT EXISTS (
        SELECT 1 FROM ApprovedApplication P
        WHERE A.email = P.email AND A.application_date = P.application_date
)
ORDER BY A.application_date ASC;
```

- Display a list of Adoption Applications with applicationDate('$application_date') and applicant's email('$email') under section **PendingApproval.**

- When user clicked a particular *Adoption Application in the list*

  o   Find the applicant who applied for the Adoption Application from **ADOPER** database.

  ```
  SELECT phone_number, household_size, firstname, lastname, street, city,
  state, zipcode
  FROM Adopter
  WHERE Adopter.email = '$email';
  ```

  o   Display **Adoption Application Details** page with display applicant's details using Adoper.name, Adoper.address, Adoper.householdSize, Adoper.Email and Adoper.PhoneNumber and 2 buttons are displayed, *Approve* and *Reject*.

  o   Executive Director could click either *Approve* button or *Reject* button to **approve/reject application.**

Add Adoption Application

Abstract Code:

- User is logged in.

- User click **add adoption application**

- User select an Adopter from Adopter database.

  > SELECT Adoper.Email FROM Adopter WHERE email = '$AdopterEmail;

- If query return null:

  - Run subtask **Add Adopter**

- User hits the submit button

- New AdoptionApplication are written to AdoptionApplication table with AdoptionApplication.ApplicationDate = $(Today's Date).

  > INSERT INTO AdoptionApplication (email, application_date)   VALUES
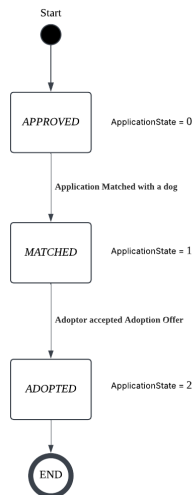 > ('$AdopterEmail', '$(Today's Date)');

Accept/Reject Adoption Application

Abstract Code

- Run mother task **View Adoption Application.**

- A particular adoption application whose state is equal to **PENDINGAPPROVAL** is clicked by Executive Director.Jump into **Adoption Application Details** and display adoption application detail with $(Email) and $(applicationDate) and 2 buttons, **APPROVE** and **REJECT**.

- Executive Director could click either **Approve** button or **Reject** button to approve or reject application.

  - If user clicked **Approve** button

    > INSERT INTO ApprovedApplication (email, application_date,
    > application_state, date_approved)
    > VALUES ('$email', '$application_date', 'APPROVED', '$(Today's Date)');

○ Note that for an approved application, its internal state transitions will be as follows:

○ If user clicked **Reject** button

```
INSERT INTO RejectedApplication (email, application_date, date_rejected)
VALUES ('$email', '$application_date', '$(Today's Date)');
```

○ Jump back to **View Adoption Application** with confirmation message.

Search for Adopers

Abstract Code

- Perform mother task **Match Adoption Application**.

- User input ('$name') for searching the adopter.

- Find all Adopters whose Adopter.lastname contains ('$name')

- Display all adopters in the list including Adoper.name, Adoper.address, Adoper.householdSize, Adoper.Email and Adoper.PhoneNumber.

```
SELECT email, phone_number, household_size, firstname, lastname, street, city,
state, zipcode
FROM Adopter
WHERE Adoper.Lastname like (%'$name'%);
```

Add Adoption Details

Abstract Code

- User click **Add Adoption** button on __View Dog Details__ page

- Record current *dog ID* ('$DogID'), *dog name* ('$DogName')

- User enters *adopter name*('$AdopterName') input field

- Search for adopter by last name (Use task: __Search for Adopter__, pass parameter: '$AdopterName')

- If at least one adopter matched:

  o User selects the adopter from the search results, record Adoper Email('$AdopterEmail').

  o Display the latest adoption application where AdoptionApplication.Date= MAX(AdoptionApplication.Date), record *application date*('$ApplicationDate')

  > SELECT application_date FROM ApprovedApplication WHERE email = 'adopter2@example.com' ORDE BY application_date DESC LIMIT 1;

  o Compute the adoption fee based on recorded expenses (Use task: __Calculate Adoption Fee__, pass parameter: '$DogID'), record *adoption fee* ('$AdoptionFee')

  o User is then prompted to enter the *adoption date* ('$AdoptionDate')

  o Display a confirmation screen listing the dog's name '$DogName', the adopter's contact information '$AdopterEmail', the adoption fee '$AdoptionFee'(if the dog belongs to any Terrier breed, and is named "Sideways", append   "(waived)" after fee), and adoption date '$AdoptionDate'.

  o If user click "confirm":

    ▪ Update related information:

      AdoptionDetails.dogID = '$DogID',

      AdoptionDetails.email = '$AdopterEmail',

      AdoptionDetails.application_date= '$ApplicationDate',

      AdoptionDetails.adoption_date = '$AdoptionDate',

AdoptionDetails.adoption_fee = '$AdoptionFee',

Dog.adoption_state = 1,

ApprovedApplication.application_state = 'ADOPTED'

INSERT INTO AdoptionDetails (dogID, email, application_date, adoption_date, adoption_fee) VALUES ('$DogID', '$AdopterEmail', '$ApplicationDate', '$AdoptionDate', '$AdoptionFee');
UPDATE Dog SET adoption_state = 1 WHERE dogID = '$DogID';
UPDATE ApprovedApplication SET application_state = 'ADOPTED' WHERE email = '$AdopterEmail' AND application_date = '$ApplicationDate';

- o Else:

    - Back to **Adopter Select** page

- Else:

    - o Show error message (No adopter matched)

Match Adoption Application

Abstract Code

- Display for available dogs (AdoptionState = 0, AlterationStatus = 1).

```
SELECT  name, sex, surrender_date, age_when_surrender, alteration_status,
description, surrenderID
FROM Dog
WHERE
AdoptionState = 0
AND AlterationStatus = 1;
SELECT Breedname FROM DogBreed WHERE dogID = '$DogID';
```

- User selects a dog and clicks "**Start *Match***" button.

- Display adoption application from ApprovedApplication whose ApprovedApplication.application_state = APPROVED.

```
SELECT ApprovedApplication.email,Adopter.firstname, Adopter.lastname,
ApprovedApplication.application_date
FROM ApprovedApplication
JOIN Adopter ON ApprovedApplication.Email = Adopter.Email WHERE
ApprovedApplication.application_state = 'APPROVED';
```

- Executive Director selects an approved adoption application and confirm selection:

    o Record ApprovedApplication.email = '$Email',

      ApprovedApplication.application_date = '$ApplicationDate'

    o Update ApprovedApplication.ApplicationState = 'Matched'.

    o Link ApprovedApplication to Dog.DogID.

    o Display confirmation: "Application matched to DogID $DogID."

```
UPDATE ApprovedApplication
SET ApprovedApplication.application_state = 'MATCHED',
ApprovedApplication.dogID = '$DogID'
WHERE ApprovedApplication.email = '$Email'
AND ApprovedApplication.application_date = '$ ApplicationDate';
```

Add Expense

Abstract Code

- User selects "***Add Expense***" button on **Dog Detail** screen.
- Record current *dog ID* ('$DogID')
- If user type is Volunteer:
    o Query User.Age by '$UserID' stored in HTTP Session, validate if User.Age ≥
      18.
```
SELECT age FROM Volunteer WHERE email = '$Email'
```
- User enters *vendor name*('$VendorName'), *amount*('$Amount'),
  category('$Category'), *date*('$Date')
- Check for duplicate expense:
    o If an expense already exists for the same Expense.VendorName =
      '$VendorName', and Expense.Date = '$Date', display an error message.
```
SELECT vendor_name, expense_date FROM Expense WHERE
vendor_name = '$VendorName' AND expense_date = '$Date'
```
- Insert new expense record into Expense table with '$DogID', '$VendorName',
  '$Amount', '$Category', and '$Date'.
```
INSERT INTO Expense (dogID, vendor_name, amount, category, expense_date)
```

```
VALUES ('$DogID', '$ VendorName', '$ Amount', '$ Category', '$ Date');
```

- Display confirmation message and update total expenses on **Dog Detail** screen.

Calculate Adoption Fee

Abstract Code

- When Adopter Sign back the Adoption contract, User will need to add Adoption details associated with a dog with '$DogID'.
- Adoption Fee field will be calculated for the dog.
- Find all Expenses from Expense Table associated with Dog.DogID = ('$DogID')
- Sum up all expenses by its value and notes as ('$Adoption Fee')

```
SELECT SUM(amount) FROM Expense WHERE DogID = '$DogID'
```

- If the dog is brought in my animal control, Update '$Adoption Fee' = ('$Adoption Fee') x 1.10.

```
SELECT surrenderID FROM Individual WHERE surrenderID = (SELECT
surrenderID FROM Dog WHERE DogID = '$DogID')
```

```
SELECT surrenderID FROM LocalAnimalControlDepartment WHERE surrenderID
= (SELECT surrenderID FROM Dog WHERE DogID = '$DogID')
```

- If the dog is a terrier, and named sideways, the adoption fee is waived.
- Else, Update ('$Adoption Fee') = ('$Adoption Fee') x 1.25.

```
SELECT AdoptionDetails.adoptionDetailsID
FROM AdoptionDetails
NATURAL JOIN ApprovedApplication
NATURAL JOIN Dog
WHERE Dog.DogID = $'DogID;
UPDATE AdoptionDetails
SET adoption_fee = '$Adoption Fee'
WHERE adoptionDetailsID = '$adoptionDetailsID';
```

View Expense Analysis

Abstract Code

- Executive Director select **View Expense Analysis** button
- Find all Expenses from Expense group by Expense.VendorName.

- Calculate the sum of the amounts spent from the Expense table, group the results by Vendor Name and name it as Total Spent. Order the results in descending order based on the total amount spent.
- Display results in a chart format with columns: Expense.VendorName, Total Spent.

```
SELECT vendor_name, SUM(amount) AS ' total_spent ' FROM Expense GROUP BY
vendor_name ORDER BY ' total spent ' DESC;
```

View Animal Control Report

Abstract Code

- Login Verification-Ensure the user is logged in as the Executive Director before proceeding.

- Executive Director selects "***View Animal Control Report****" Button.*

- Fetch Animal Control Data:

  -Query the Surrender table to get the count of dogs surrendered by animal control in the last 6 months, including the current month.

  -Query the Adoption table to get data on dogs that were adopted after being in the shelter for 60+ days during the same period.[i]

 - Query the Expense table to calculate the total expenses for the adopted dogs.

```
SELECT m.month,
  IFNULL(surrendered_count, 0) AS surrendered_by_animal_control,
  IFNULL(expense_total, 0) AS total_expenses,
  IFNULL(adopted_count, 0) AS adopted_after_60_days
FROM (SELECT DATE_FORMAT(CURRENT_DATE - INTERVAL 5 MONTH,
'%Y-%m') AS month UNION ALL
  SELECT DATE_FORMAT(CURRENT_DATE - INTERVAL 4 MONTH, '%Y-%m')[iiiii]
UNION ALL
  SELECT DATE_FORMAT(CURRENT_DATE - INTERVAL 3 MONTH, '%Y-%m')
UNION ALL
  SELECT DATE_FORMAT(CURRENT_DATE - INTERVAL 2 MONTH, '%Y-%m')
UNION ALL
  SELECT DATE_FORMAT(CURRENT_DATE - INTERVAL 1 MONTH, '%Y-%m')
UNION ALL
```

```
  SELECT DATE_FORMAT(CURRENT_DATE, '%Y-%m')) AS m
  LEFT JOIN (SELECT DATE_FORMAT(d.surrender_date, '%Y-%m') AS month,
  COUNT(DISTINCT CASE WHEN lacd.surrenderID IS NOT NULL THEN d.dogID
END) AS surrendered_count
  FROM Dog d
  LEFT JOIN Surrender s
  ON d.surrenderID = s.surrenderID
  LEFT JOIN LocalAnimalControlDepartment lacd ON s.surrenderID =
lacd.surrenderID
  WHERE d.surrender_date >= DATE_FORMAT(CURRENT_DATE - INTERVAL 6
MONTH, '%Y-%m-01')
  GROUP BY DATE_FORMAT(d.surrender_date, '%Y-%m')) AS s ON m.month =
s.month
  LEFT JOIN (SELECT DATE_FORMAT(e.expense_date, '%Y-%m') AS month,
  SUM(e.amount) AS expense_total
  FROM Expense e
  WHERE e.expense_date >= DATE_FORMAT(CURRENT_DATE - INTERVAL 6
MONTH, '%Y-%m-01')
  GROUP BY DATE_FORMAT(e.expense_date, '%Y-%m')) AS e ON m.month =
e.month
  LEFT JOIN (SELECT DATE_FORMAT(ad.adoption_date, '%Y-%m') AS month,
  COUNT(DISTINCT CASE WHEN DATEDIFF(ad.adoption_date,
d.surrender_date) >= 60 THEN d.dogID END) AS adopted_count
  FROM Dog d
  LEFT JOIN approvedapplication aa
  ON d.dogID = aa.dogID
  LEFT JOIN AdoptionDetails ad ON aa.email = ad.email AND aa.application_date
= ad.application_date
  WHERE ad.adoption_date >= DATE_FORMAT(CURRENT_DATE - INTERVAL 6
MONTH, '%Y-%m-01')
  GROUP BY DATE_FORMAT(ad.adoption_date, '%Y-%m')) AS a ON m.month =
a.month
ORDER BY m.month DESC;
```

- Display Aggregated Report:

  -Show the number of dogs surrendered by animal control for each month.

  -Display the count of dogs adopted after 60+ days in the shelter.

  -Show the total expenses for the dogs adopted in each month.

- Drill-Down Functionality:

-Allow drill-downs for each category

-Animal Control Surrenders Drill Down: record selected *month* '$SelectedMonth',
*s*how Dog.DogID, DogBreed.Breed where dogID = '$dogID', Dog.Sex,
Dog.AlterationStatus, Microchip.MicrochipID, Dog.SurrenderDate.

```
SELECT
  d.dogID,
  GROUP_CONCAT(DISTINCT db.breedname ORDER BY db.breedname
SEPARATOR ', ') AS breed,
  d.sex,
  d.alteration_status,
  MAX(m.microchipID) AS microchipID,
  d.surrender_date
FROM Dog d
  JOIN Surrender s ON d.surrenderID = s.surrenderID
  JOIN LocalAnimalControlDepartment lacd ON s.surrenderID = lacd.surrenderID
  LEFT JOIN DogBreed db ON d.dogID = db.dogID
  LEFT JOIN Microchip m ON d.dogID = m.dogID
WHERE DATE_FORMAT(d.surrender_date, '%Y-%m') = '$SelectedMonth'
GROUP BY d.dogID, d.sex, d.alteration_status, d.surrender_date
ORDER BY d.dogID ASC;
```

```
-- Drill Down Report Animal control surrenders
SELECT
    DOG.dogID,
    GROUP_CONCAT(dogbreed.breedname  ORDER  BY  dogbreed.breedname
SEPARATOR ', ') AS breed_names,
    DOG.Sex,
    DOG.alteration_status,
    ANY_VALUE(microchip.microchipID) as microchipID,
    DOG.surrender_date
FROM DOG
NATURAL JOIN localanimalcontroldepartment
NATURAL JOIN approvedapplication
NATURAL JOIN adoptiondetails
NATURAL JOIN dogbreed
LEFT JOIN microchip ON DOG.dogID = microchip.dogID
WHERE     DATE_FORMAT(adoptiondetails.adoption_date,     '%Y-%m')     =
'$SelectedMonth'
GROUP BY DOG.dogID
ORDER BY DOG.dogID ASC;
```

-Long Stay Adoptions Drill Down:

- record selected *month* '$SelectedMonth', show Dog. Dog ID, DogBreed.Breed where dogID = '$dogID', Dog. Sex, Dog. Microchip ID, Dog. Surrender Date in Rescue.

```
SELECT
  d.dogID,
  GROUP_CONCAT(DISTINCT db.breedname ORDER BY db.breedname
SEPARATOR ', ') AS breed,
  d.sex,
  MAX(m.microchipID) AS microchipID,
  d.surrender_date,
  DATEDIFF(ad.adoption_date, d.surrender_date) AS days_in_rescue
FROM Dog d
  LEFT JOIN ApprovedApplication aa ON d.dogID = aa.dogID
  LEFT JOIN AdoptionDetails ad ON aa.email = ad.email AND aa.application_date
= ad.application_date
  LEFT JOIN DogBreed db ON d.dogID = db.dogID
  LEFT JOIN Microchip m ON d.dogID = m.dogID
WHERE DATEDIFF(ad.adoption_date, d.surrender_date) >= 60
  AND DATE_FORMAT(ad.adoption_date, '%Y-%m') = '$SelectedMonth'
GROUP BY d.dogID, d.sex, d.surrender_date, days_in_rescue
ORDER BY days_in_rescue DESC, d.dogID DESC;
```

```
-- Drill Down Report Dogs adopted in the rescue 60 days or more
SELECT
    DOG.dogID,
    GROUP_CONCAT(dogbreed.breedname ORDER BY dogbreed.breedname
SEPARATOR ', ') AS breed_names,
    DOG.Sex,
    DOG.alteration_status as AlterationStatus,
    ANY_VALUE(microchip.microchipID) as microchipID,
    DOG.surrender_date as surrender_date,
    ANY_VALUE(adoptiondetails.adoption_date) as adoption_date,
    ANY_VALUE(DATEDIFF(adoptiondetails.adoption_date,
DOG.surrender_date) + 1) AS NumberofDaysInRescue
FROM DOG
LEFT JOIN localanimalcontroldepartment ON DOG.surrenderID =
localanimalcontroldepartment.surrenderID
LEFT JOIN individual ON DOG.surrenderID = individual.surrenderID
NATURAL JOIN adoptiondetails
NATURAL JOIN dogbreed
LEFT JOIN microchip ON DOG.dogID = microchip.dogID
WHERE DATE_FORMAT(adoptiondetails.adoption_date, '%Y-%m') =
'$SelectedMonth'
```

```
GROUP BY DOG.dogID HAVING NumberofDaysInRescue >= 60
ORDER BY NumberofDaysInRescue DESC, DOG.dogID DESC;
```

- Adoption Expenses Drill Down:

  - Record selected *month* '$SelectedMonth', show Dog.DogID, DogBreed.Breed
  where dogID = '$dogID', Dog.Sex, Microchip.MicrochipID where dogID = '$dogID',
  AdoptionDetails. AdoptionDate, Dog.SurrenderDate, Expense.Amount

  Animal Control Indicator, Total Expenses.

```
SELECT
    DOG.dogID,
    GROUP_CONCAT(dogbreed.breedname ORDER BY dogbreed.breedname
    SEPARATOR ', ') AS breed_names,
    DOG.Sex,
    ANY_VALUE(microchip.microchipID) as microchipID,
    IF(individual.surrenderID IS NULL, 'YES', 'FALSE') AS
    AnimalControlSurrenderIndicator,
    DOG.surrender_date,
    ANY_VALUE(adoptiondetails.adoption_fee) as TotalExpenses
FROM DOG
    LEFT JOIN localanimalcontroldepartment ON DOG.surrenderID =
    localanimalcontroldepartment.surrenderID
    LEFT JOIN individual ON DOG.surrenderID = individual.surrenderID
    NATURAL JOIN approvedapplication
    NATURAL JOIN adoptiondetails
    LEFT JOIN dogbreed ON dogbreed.dogID = dog.dogID
    LEFT JOIN microchip ON DOG.dogID = microchip.dogID
WHERE DATE_FORMAT(adoptiondetails.adoption_date, '%Y-%m') = '2025-03'
GROUP BY DOG.dogID;
```

- Display Results:

  Show the aggregated data and drill-downs for the last 6 months+, reflecting real-time
  data for the current month.

View Monthly Adoption Report

Abstract Code

- Login Verification-Ensure the user is logged in as the Executive Director before

proceeding.

- Executive Director selects *"**Monthly Adoption Report**"* Button.

```sql
-- monthly adoption report
CREATE VIEW PAST_12MTHS AS
SELECT DATE_FORMAT(CURDATE() - INTERVAL m MONTH, '%Y-%m') AS month
FROM (
  SELECT 1 AS m UNION ALL
  SELECT 2 UNION ALL
  SELECT 3 UNION ALL
  SELECT 4 UNION ALL
  SELECT 5 UNION ALL
  SELECT 6 UNION ALL
  SELECT 7 UNION ALL
  SELECT 8 UNION ALL
  SELECT 9 UNION ALL
  SELECT 10 UNION ALL
  SELECT 11 UNION ALL
  SELECT 12
) AS months;

CREATE VIEW DOG_STATUS AS
SELECT
  dog.dogID,
  GROUP_CONCAT( DISTINCT dogbreed.breedname ORDER BY dogbreed.breedname
SEPARATOR ', ') AS breed_names,
  ANY_VALUE(dog.surrender_date) AS surrender_date,
  ANY_VALUE(adoptiondetails.adoption_date) AS adoption_date,
  CASE WHEN individual.surrenderID IS NULL THEN 0 ELSE 1 END AS
SurrenderedbyIndividual
FROM dog
  LEFT JOIN approvedapplication on approvedapplication.dogID = DOG.dogID
  LEFT JOIN adoptiondetails ON adoptiondetails.application_date =
approvedapplication.application_date AND adoptiondetails.email =
approvedapplication.email
  LEFT JOIN dogbreed ON dogbreed.dogID = DOG.dogID
  LEFT JOIN individual ON DOG.surrenderID = individual.surrenderID
GROUP BY DOG.dogID;

SELECT
  PAST_12MTHS.`month`,
  DOG_STATUS.breed_names,
  COUNT(PAST_12MTHS.`month` = DATE_FORMAT(DOG_STATUS.surrender_date,
'%Y-%m')) AS DogsSurrendered,
```

```
   COUNT(PAST_12MTHS.`month` = DATE_FORMAT(DOG_STATUS.adoption_date,
'%Y-%m')) AS DogsAdopted,
 IFNULL(SUM(expense.amount), 0) AS TotalExpenses,
 IFNULL(SUM(adoptiondetails.adoption_fee), 0) AS AdoptionFees,
 (IFNULL(SUM(adoptiondetails.adoption_fee), 0) - IFNULL(SUM(expense.amount), 0) *
ANY_VALUE(DOG_STATUS.SurrenderedbyIndividual)) AS NetProfit
FROM PAST_12MTHS
 LEFT JOIN DOG_STATUS ON
 (PAST_12MTHS.`month` = DATE_FORMAT(DOG_STATUS.adoption_date, '%Y-%m')
 OR
 PAST_12MTHS.`month` = DATE_FORMAT(DOG_STATUS.surrender_date, '%Y-%m'))
 LEFT JOIN approvedapplication ON approvedapplication.dogID = DOG_STATUS.dogID
 LEFT JOIN adoptiondetails ON adoptiondetails.application_date =
approvedapplication.application_date AND adoptiondetails.email =
approvedapplication.email AND PAST_12MTHS.`month` =
DATE_FORMAT(adoptiondetails.adoption_date, '%Y-%m')
 LEFT JOIN expense ON expense.dogID = DOG_STATUS.dogID AND
PAST_12MTHS.`month` = DATE_FORMAT(expense.expense_date, '%Y-%m')
GROUP BY PAST_12MTHS.`month`, breed_names
ORDER BY PAST_12MTHS.`month`, breed_names;
```

- o Display the number of dogs surrendered and adopted each month, the total expenses, total adoption fees, and net "profit" (animal controlsurrendered dogs' expenses should not be considered)

- o Perform the queries order by month ASC and by breed name alphabetically (A-Z)

View Birthday Celebration Report

Abstract Code

- Executive Director selects *"View Birthday Celebration Report"* Button.

- Perform query: Find volunteer details (firstname, lastname, email, birthdate) from CasualUser table, check if any volunteers have birthday in specific month and is it a milestone birthday. User enter the *selected month* '$SelectedMonth' (default is current month) and *selected year* '$SelectedYear' (default is current year, only the current and previous year).

```
SELECT
    cu.firstname,
    cu.lastname,
    cu.email,
    CASE
```

```
            WHEN ('$SelectedYear' - YEAR(cu.birthday))   % 10 = 0 THEN 'Yes'
            ELSE 'No'
        END AS is_milestone
FROM
        CasualUser cu
        INNER JOIN Volunteer v ON cu.email    = v.email
WHERE
        MONTH(cu.birthday)    = '$SelectedMonth'
        AND cu.birthday    IS NOT NULL;
```

- If no record found for the *selected month* and *selected year*:
    - Display message: "No volunteer birthdays this month!"
- Else:
    - Display report with columns from CasualUser: firstname, lastname, email, and append is_milestone Birthday.

Search for Volunteers

Abstract Code

- User enters *volunteer name* ('$Volunteername') input field.
- Show **Search** tab.
- If data validation is successful for volunteer name input field, then:
    - When **Search** button is clicked:
        - Perform query: Find CasualUser.FirstName, CasualUser.LastName, CasualUser.Email, CasualUser.PhoneNumber of users where the CasualUser.firstName or CasualUser.lastName (in lowercase) matches the lowercase version of the specified volunteer name. Then, order the results by last name in ascending order and, within the same last name, by first name in ascending order.

```
SELECT firstname, lastname, email, phone_number
FROM CasualUser
NATURAL JOIN volunteer
WHERE
    LOWER(firstname) LIKE LOWER(CONCAT('%',
'$Volunteername', '%')) OR
    LOWER(lastname) LIKE LOWER(CONCAT('%',
'$Volunteername', '%'))
ORDER BY lastname ASC, firstname ASC;
```

        - If results are found:
            - Display results in a table format with columns:
                CasualUser.FirstName, CasualUser.LastName,

CasualUser.Email Address, CasualUser.PhoneNumber.

- Else:
  - Display search form with error message: "No results found for the entered name."
- Else:

Display search form with error message: "Invalid input. Please enter a valid name."