

ITE 1942 – ICT PROJECT

PROJECT REPORT

Level 01

**Management System for Pet Adoption
and Rehoming for Street Cats and Dogs
in Sri Lanka**

Submitted by:

Obeysekara KDPU

E2245294

Bachelor of Information Technology (External Degree)

Faculty of Information Technology

University of Moratuwa

Abstract

In Sri Lanka, as in many parts of the world, the issue of homeless street animals, particularly cats and dogs, is a pressing concern. These animals roam the streets without proper care or shelter, facing numerous challenges to their health and well-being. Despite the magnitude of this problem, there is a noticeable lack of organized efforts to address it effectively. In response to this urgent need, I propose the development of a comprehensive online platform aimed at facilitating the adoption and rehoming of these vulnerable animals.

The vision behind this initiative is simple yet powerful: to create a centralized platform where concerned individuals can report sightings of homeless cats and dogs and provide essential information about them. This platform will serve as a digital hub, allowing users to update details such as the animal's breed, traits, current location, and any known health issues. Through this system, users will be empowered to play an active role in connecting these animals with potential adopters, thereby giving them a chance at a better life.

The proposed website will feature user-friendly interfaces that facilitate seamless navigation and interaction. Upon registering, users will gain access to a range of features, including the ability to view detailed profiles of street animals in need of homes and connect with relevant stakeholders involved in animal welfare. By harnessing the power of technology and community engagement, this platform seeks to streamline the process of rehoming street animals and provide them with the care and attention they deserve.

Moreover, this project aligns with broader objectives of promoting compassion and responsibility toward animals within society. By raising awareness and fostering a culture of empathy, we aim to create a more humane and caring environment for both animals and humans alike. Through collaborative efforts and innovative solutions, we can make a meaningful impact on the lives of countless street animals in Sri Lanka, paving the way for a brighter future for all.

Keywords: street animals, pet adoption, rehoming, animal welfare, online platform, Sri Lanka, community engagement

Table of Contents

1 INTRODUCTION	04
1.1 BACKGROUND AND MOTIVATION	04
1.2 PROBLEM IN BRIEF	04
1.3 AIM AND OBJECTIVES.....	05
1.4 SIGNIFICANCE OF THE PROJECT.....	05
1.5 SUMMARY.....	06
2 RELATED WORKS	07
2.1 INTRODUCTION	07
2.2 SIMILAR SYSTEMS	07
2.3 DRAWBACKS AND WEAKNESSES OF THE EXISTING SYSTEM	10
2.4 SUMMARY	11
3 SYSTEM ANALYSIS	12
3.1 INTRODUCTION	12
3.2 SYSTEM REQUIREMENTS	12
3.3 FUNCTIONAL REQUIREMENTS	12
3.4 NON-FUNCTIONAL REQUIREMENTS	14
3.5 SUMMARY	15
4 SYSTEM DESIGN	16
4.1 FLOWCHARTS	16
4.2 PSUEDOCODES	23
5 SYSTEM IMPLEMENTATION	33
5.1 INTERFACES	33
5.2 IMPORTANT CODES	37
6 REFERENCES	43

1. INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Sri Lanka faces a significant issue with homeless cats and dogs, lacking proper care and facing health risks. The current approach is fragmented, highlighting the need for a more organized solution.

This project outlines the development of a centralized online platform for animal welfare. This platform would allow individuals to report sightings and details about homeless animals, creating a central hub for adoption. Users could access animal profiles and connect with relevant organizations, ultimately aiming to streamline the rehoming process and promote animal welfare in Sri Lanka.

1.2 PROBLEM IN BRIEF

Homeless cats and dogs in Sri Lanka suffer due to lack of care and face health risks. The current efforts to help them are scattered and ineffective.

- Uncared-for Animals: Large numbers of homeless cats and dogs roam Sri Lanka's streets, lacking necessities like food and shelter.
- Health Risks: These animals are exposed to dangers like disease, injuries from traffic accidents, and fights with other strays.
- Fragmented Support: Existing efforts to help these animals are disorganized and lack a central focus, making it difficult to address the problem effectively.
- Public Health Concerns: Uncontrolled breeding of strays can lead to overpopulation, increasing the risk of rabies transmission to humans and other animals.
- Negative Perceptions: Large numbers of strays can be seen as a nuisance by some, leading to fear, dislike, and even cruelty towards the animals.
- Limited Resources: Animal welfare organizations in Sri Lanka often operate with limited resources, making it difficult to provide adequate care, sterilization, and adoptions for all homeless animals.
- Cruelty and Neglect: Sadly, some homeless animals face deliberate cruelty, neglect, and abuse from humans.

1.3 AIM AND OBJECTIVES

This project aims to utilize technology and community engagement to address the widespread issue of homeless cats and dogs in Sri Lanka. Through a centralized online platform, the project seeks to streamline the rehoming process and improve the well-being of these vulnerable animals.

The project will establish a user-friendly online platform that connects concerned citizens with homeless animals and potential adopters. This platform will empower users to report animal sightings, provide details about the animals, and connect with animal welfare organizations. By fostering collaboration and raising awareness, the project aims to create a more compassionate and responsible society towards animals in Sri Lanka.

Specific Aims and Objectives:

- Establish a centralized platform: Create a user-friendly website where individuals can report sightings and access information about homeless cats and dogs.
- Facilitate seamless adoption: Allow potential adopters to view detailed profiles of animals and connect with relevant stakeholders for the adoption process.
- Empower user participation: Enable users to update animal details, fostering a collaborative environment for animal welfare.
- Promote animal welfare: Raise awareness about the plight of homeless animals and encourage responsible pet ownership within Sri Lanka.

These combined aims and objectives work together to create a comprehensive solution for rehoming street animals and improving their overall well-being.

1.4 SIGNIFICANCE OF THE PROJECT

The implementation of the proposed pet adoption system is expected to yield significant benefits including,

- Improved Animal Welfare: The platform directly addresses the critical issue of homeless animals in Sri Lanka. By facilitating adoptions and rehoming, it will significantly improve the lives of countless cats and dogs by providing them with proper care, shelter, and a loving home.
- Reduced Strain on Resources: Stray animals often rely on scavenging for food, leading to public health concerns. The project will lessen the number of strays, reducing the strain on sanitation efforts and animal control services.

- Community Engagement: The platform fosters a culture of compassion and responsibility towards animals. Users actively participate in reporting and sharing information, fostering a collaborative spirit for animal welfare.
- Increased Adoption Rates: By centralizing information and simplifying the adoption process, the platform is likely to increase successful adoptions, reducing the number of animals in shelters and on the streets.
- Positive Social Impact: Creating a more humane environment for animals can have a positive ripple effect on society. Responsible pet ownership and animal welfare practices contribute to a more caring and compassionate community.

This project offers a significant solution to the problem of homeless animals in Sri Lanka. It improves animal welfare, fosters community engagement, and contributes to a more positive social impact for both animals and humans.

1.5 SUMMARY

This project tackles the critical issue of homeless cats and dogs in Sri Lanka. These animals suffer due to neglect, facing health risks and limited resources. Scattered efforts to help them highlight the need for a more organized solution.

The project proposes a centralized online platform to improve animal welfare. This platform will connect concerned citizens with homeless animals and potential adopters. Users can report animal sightings, share details, and connect with relevant organizations, streamlining the rehoming process. This initiative is expected to improve animal lives, reduce the burden on animal control services, foster a culture of animal welfare, and create a more compassionate society in Sri Lanka.

2. RELATED WORK

2.1 INTRODUCTION

This chapter provides an overview of management platforms for pet adoption or rehoming for street cats and dogs that have been developed and implemented. The following section describes features of such systems and their details.

2.2 SIMILAR SYSTEMS

2.2.1 Embark Pet Adoption Platform



The screenshot shows the homepage of the Embark Pet Adoption Platform. At the top, there is a banner with a photo of a woman holding a dog and the text "Dogs in Sri Lanka". Below the banner are social media sharing icons for Facebook, Twitter, YouTube, and Instagram. The main content area features a large image of a woman holding a dog. Below this, there is a navigation bar with links: Home, About Us, Our Work ▾, The Brand ▾, Events, Donate Now ▾, Join Us ▾, Publications ▾, and News and Media. A sub-section below the navigation bar states: "We held Community Clinics for street dogs in Mirihana and Piliyandala to support the people in the community to treat their pets. Many are finding it difficult to treat them due to financial issues and rising costs." There are three smaller images below this text: one showing people at a clinic, one showing a dog, and one showing a close-up of a dog's face.

Embark Pet Adoption Platform is an innovative online platform designed to streamline the process of adopting rescued street dogs in Sri Lanka. The platform provides a user-friendly interface where potential adopters can browse through profiles of available dogs, complete with photos, descriptions, and information about their personalities and backgrounds. Adoption can easily search for dogs based on criteria such as age, size, and temperament, making it convenient to find the perfect match for their lifestyle and preferences.

The platform also facilitates the adoption process by allowing users to submit adoption applications online, schedule meet and greet sessions with dogs they are interested in and communicate with adoption coordinators. Additionally, the platform includes features such as online payment for adoption fees, virtual home checks, and access to resources on pet care and training. Through this platform, embark aims to make the adoption process more,

accessible, transparent, and efficient ultimately contributing to reducing the number of street dogs in Sri Lanka and improving their welfare.



Embark Volunteer Registration

e2245294@bit.uom.lk [Switch account](#) 

 Not shared

* Indicates required question

Embark Volunteer Registration

Full Name: *

Your answer

Date of Birth: *

Date

mm/dd/yyyy 

Volunteers below the age of 16 are required to receive parental permission to register as a Volunteer. *

Age: 16 and above

Age: Below 16 and have parental permission

National ID or Passport Number: *

Your answer

Province you Reside *

Choose 

2.2.2 Animal Welfare and Protection Association (AWPA)

AWPA platforms provide information about their mission, activities, and initiatives related to animal welfare in Sri Lanka. While specific details may vary, the website typically includes information about animals available for adoption, adoption procedures, and contact details for individuals interested in adopting or supporting AWPA's efforts. The website may also feature resources on responsible pet ownership, updates on ongoing projects and campaigns, opportunities for volunteering and donating, and ways to get involved in advocating for animal rights and welfare in Sri Lanka.

AWPA platform acts as a central hub for promoting their cause, connecting with supporters and potential adopters, and raising awareness about the importance of animal protection and welfare in Sri Lanka.

**THE ANIMAL WELFARE & PROTECTION ASSOCIATION SRI LANKA
MEMBERSHIP APPLICATION FORM**

Title *

First Name *
Last Name

Address
Address Line 1 * **Address Line 2**
City * **Country**

Phone *

Email *

Are you Employed?
 Yes

I'm interested in helping in *
 Volunteering in the shelters
 Fund-raising
 Visiting re-homed animals
 Assisting at sterilizations
 Providing data

Choose Membership
 Life membership (Rs.2,500)
 Student life membership (Rs.1,000) (Under 25 years of age)

Proposed by an existing member

I support the aim of the society to promote kindness and to prevent or suppress cruelty to animals and work for the welfare of animals

Submit Form

Beyond AWPA's informative website, some Animal Welfare and Protection Associations (AWPAs) in Sri Lanka might extend their services through foster care programs. These programs place rescued animals with volunteer caregivers who provide temporary care and socialization until a permanent home is found. Fostering allows AWPAs to help more animals by creating space within their shelters and enables potential adopters to experience the joys of pet ownership before making a lifelong commitment. This fosters a more informed and responsible adoption process, ultimately benefiting the well-being of animals in Sri Lanka.



Bart

This beautiful boy is as cheeky as his namesake, Bart Simpson! He's full of fun and games, and cuddles and snuggles are his forte!

Online Inquiry

Name

Email

Phone

Message

* You will automatically receive a copy of your request via email

Submit

[← Previous Post](#) [Next Post →](#)

2.3 DRAWBACKS AND WEAKNESSES OF THE EXISTING SYSTEM

Sri Lanka's current system for adopting homeless animals faces several challenges that hinder its effectiveness:

- Fragmented Approach: Reliance on individual AWPA websites and limited coordination between organizations leads to a disjointed effort. Potential adopters have difficulty finding animals in need due to the lack of a centralized platform.
- Lack of Centralized Information: The absence of a single platform makes it difficult for potential adopters to get a comprehensive view of animals up for adoption across

different AWPs and shelters. This scatters information and reduces the efficiency of the adoption process.

- Limited Visibility: Individual AWPA websites may have lower visibility compared to a dedicated adoption platform. This limits outreach and potentially reduces the number of people who are aware of animals available for adoption.
- Resource Constraints: AWPs might struggle with resource limitations, hindering their ability to effectively promote adoptions and care for a large number of animals. This can lead to overcrowding in shelters and limit their capacity to take in new animals.

These drawbacks significantly impact the efficiency of animal adoption in Sri Lanka. A centralized online platform can address these weaknesses by consolidating information, improving outreach, and streamlining the adoption process.

2.4 SUMMARY

This chapter explores existing systems for pet adoption and rehoming in Sri Lanka. There are two main approaches: dedicated adoption platforms and Animal Welfare Organizations (AWPs). Platforms like Embark offer user-friendly interfaces for browsing adoptable animals, with photos, descriptions, and even online applications. Some platforms provide pet care resources. AWPs maintain websites showcasing their work and adoption information, connecting potential adopters with animals. Some AWPs even run foster care programs for temporary pet placement.

Despite these efforts, the current system faces challenges. The fragmented approach, with information scattered across individual AWPA websites, makes it difficult for adopters to find animals. The lack of a central platform hinders a comprehensive view of adoptable pets and reduces outreach. Limited website visibility for some AWPs further restricts potential adopters. Finally, resource constraints can limit AWPA's ability to promote adoptions and manage animal capacity. These drawbacks highlight the need for a centralized online platform to streamline animal adoption in Sri Lanka.

Embark Pet Adoption Platform is a Sri Lankan website designed to streamline dog adoption. It offers a user-friendly interface for browsing dog profiles with photos, descriptions, and personality traits. Users can search by age, size, and temperament to find their perfect match. The platform streamlines the adoption process by allowing online applications, meet-and-greets, communication with coordinators, and even online adoption fee payment. It also provides virtual home checks and pet care resources.

3. SYSTEM ANALYSIS

3.1 INTRODUCTION

Sri Lanka faces a significant problem with homeless cats and dogs. These animals lack proper care and suffer health risks. Existing efforts to help them are scattered and ineffective. This project proposes a solution: a centralized online platform for animal welfare. This platform would connect concerned citizens with homeless animals and potential adopters. Users can report animal sightings, share details, and connect with relevant organizations. This aims to improve animal lives, reduce the burden on animal control, and foster a more compassionate society.

3.2 SYSTEM REQUIREMENTS

Software

- **Implementation:** Microsoft Visual Studio 2022
- **Database Connection:** Microsoft SQL Server Express 2022
- **Image Editor:** Ideogram.AI
- **Operating System:** Windows 11

Hardware

- **Processor:** Intel Core i5
- **Random Access Memory:** 4.00GB
- **System type:** 64-bit Operating System, x64-based processor
- **Computer name:** LAPTOP - QUEEHTJU

3.3 FUNCTIONAL REQUIREMENTS

Functional Requirements for the Sri Lankan Street Animal Rehoming Platform:

1. User Registration and Management:

Users can register as individuals or representatives of animal welfare organizations.
Users can create and manage their profiles.

2. Animal Sightings and Reporting:

Users can report sightings of homeless cats and dogs.

Reports should include details like location, animal type, breed (if identifiable), and any visible injuries.

Users can upload photos of the animals.

3. Animal Profiles and Management:

The platform should create profiles for each reported animal.

Profiles should display details like location, breed, description, temperament (if known), and any health concerns reported.

Authorized users from animal shelters or rescue organizations can claim responsibility for specific animals.

Authorized users can update animal profiles with additional information and photos.

4. Search and Filtering:

Users can search for adoptable animals based on various criteria like location, breed, size, and temperament.

5. Adoption Process Facilitation:

Users can express interest in adopting specific animals.

The platform should enable communication between potential adopters and animal shelters/rescue organizations responsible for the animal's care.

The platform can optionally include features to manage the adoption application process (if handled by the platform).

6. Content Management:

Authorized administrators can manage website content such as informative articles and resources about animal care and responsible pet ownership.

7. Reporting and Analytics:

The platform should generate reports on animal sightings, adoptions, and user activity.

These reports can be used to track progress, identify trends, and optimize platform functionality.

8. Security and User Management:

The platform should implement secure user authentication and authorization mechanisms.

Different user roles (e.g., regular user, shelter admin) should have access controls for specific functionalities.

3.4 NON-FUNCTIONAL REQUIREMENTS

In addition to the functional requirements that outline the specific actions and features of the platform, non-functional requirements define its overall characteristics and how it should operate. Here are some key non-functional requirements for the Sri Lankan Street Animal Rehoming Platform:

1. Performance:

The platform should be responsive and deliver fast loading times for users with varying internet speeds. This is crucial to ensure a smooth user experience, especially for those in remote areas with limited bandwidth.

2. Scalability:

The platform should be designed to accommodate a growing user base and increasing numbers of animal listings. The system should be able to handle surges in traffic and data without compromising performance.

3. Availability:

The platform should be highly available with minimal downtime. Regular maintenance and backups are essential to ensure continuous operation and accessibility for users seeking to report animals, browse listings, or adopt pets.

4. Usability:

The platform's interface should be user-friendly and intuitive for people with varying levels of technical expertise. A clear layout, simple navigation, and well-defined functionalities are essential for encouraging user adoption and engagement.

5. Accessibility:

The platform should be accessible to users with disabilities. This includes features like screen reader compatibility, keyboard navigation, and clear visual design to ensure everyone can access information and participate in the adoption process.

6. Security:

The platform must prioritize user and animal data security. This includes robust security measures to protect user information, login credentials, and animal details. Secure data storage and encryption are essential to prevent unauthorized access and ensure user trust.

7. Reliability:

The platform should function reliably and consistently. This means minimal bugs, errors, or crashes that could disrupt user experience or hinder animal adoption efforts. Regular testing and quality assurance processes are crucial to maintain platform reliability.

8. Maintainability:

The platform's codebase should be well-documented, modular, and easy to maintain. This allows for future updates, bug fixes, and feature enhancements to be implemented efficiently.

9. Interoperability:

The platform should ideally be interoperable with other animal welfare organizations' databases or applications (if applicable). This could allow for data sharing and streamlined adoption processes across different organizations.

10. Localization (Optional):

The platform interface and content can be localized to support multiple languages spoken in Sri Lanka. This can broaden user reach and cater to a wider audience, fostering a more inclusive environment for animal adoption.

3.5 SUMMARY

This chapter explains a system created to help Sri Lankan dogs and cats who are homeless. Users will be able to report adoptable street cats and dogs and report animal sightings on the system. Searches for animals can be done using parameters like breed and location. Enhancing animal welfare in Sri Lanka and matching new owners with animals in need are the two main objectives.

The technical specifications for the system are also included in the document. These include of reporting tools, search capabilities, and user account administration. The website should also be dependable, safe, and simple to use.

4. SYSTEM DESIGN

4.1 FLOW CHARTS

Flow Charts of some main sub systems are as follows.

Figure 1 The flowchart guides potential adopters through the process of finding their perfect pet. It begins by welcoming users who can then browse available animals based on their preferences. Once they choose a furry friend, the system checks if the user meets the adoption criteria. If eligible, an application is submitted for review, followed by a background check from the shelter. Finally, the application is approved or denied. If approved, the user pays the adoption fee and finalizes the process, taking their new pet home!

Figure 2 The pet adoption system flowchart starts by welcoming potential adopters. Users browse pets based on their preferences and choose one. An eligibility check ensures the user meets the requirements (like having a fenced yard). If eligible, an application is submitted, followed by a background check by the shelter. The application is then approved or denied. If approved, the user pays the adoption fee and finalizes the process, taking their furry friend home!

Figure 3 The pet adoption system starts by welcoming potential adopters. Users can then browse pets based on their preferences, such as species or age. Once they've chosen a furry companion, the system checks if the user meets the adoption criteria. This might involve factors like having a fenced yard or prior pet ownership. If eligible, users submit an adoption application which is then reviewed by the shelter, often accompanied by a background check. Finally, the application is approved or denied. If successful, the user pays the adoption fee and finalizes the process, welcoming their new pet into their loving home!

Figure 4

The pet adoption system welcomes potential adopters and allows them to browse a selection of furry companions based on their preferences. Once a captivating critter catches their eye, the system verifies if the user meets the adoption criteria. This might involve factors like having a fenced yard or experience caring for pets. If deemed suitable, users submit an adoption application for review. The shelter then assesses the application, often accompanied by a background check to ensure a safe and loving environment. Finally, the application is approved or denied. If successful, the user pays the adoption fee and finalizes the process, welcoming their new furry friend home!

Figure 5:

The pet adoption system welcomes potential adopters with open arms! Users can browse a delightful menagerie of furry (or perhaps feathery or scaled) companions, filtering by their preferences. Once a captivating critter catches their eye, the system verifies if the user meets the adoption criteria. This might involve factors like having a fenced yard or experience caring for pets. If deemed a suitable companion, users submit an adoption application for review. The shelter then carefully assesses the application, often accompanied by a background check to ensure a safe and loving environment for the animal. Finally, the application is approved or denied. If the user is chosen as a perfect match, they pay the adoption fee and finalize the process, welcoming their new furry friend home to a lifetime of love and cuddles!

Figure 1:

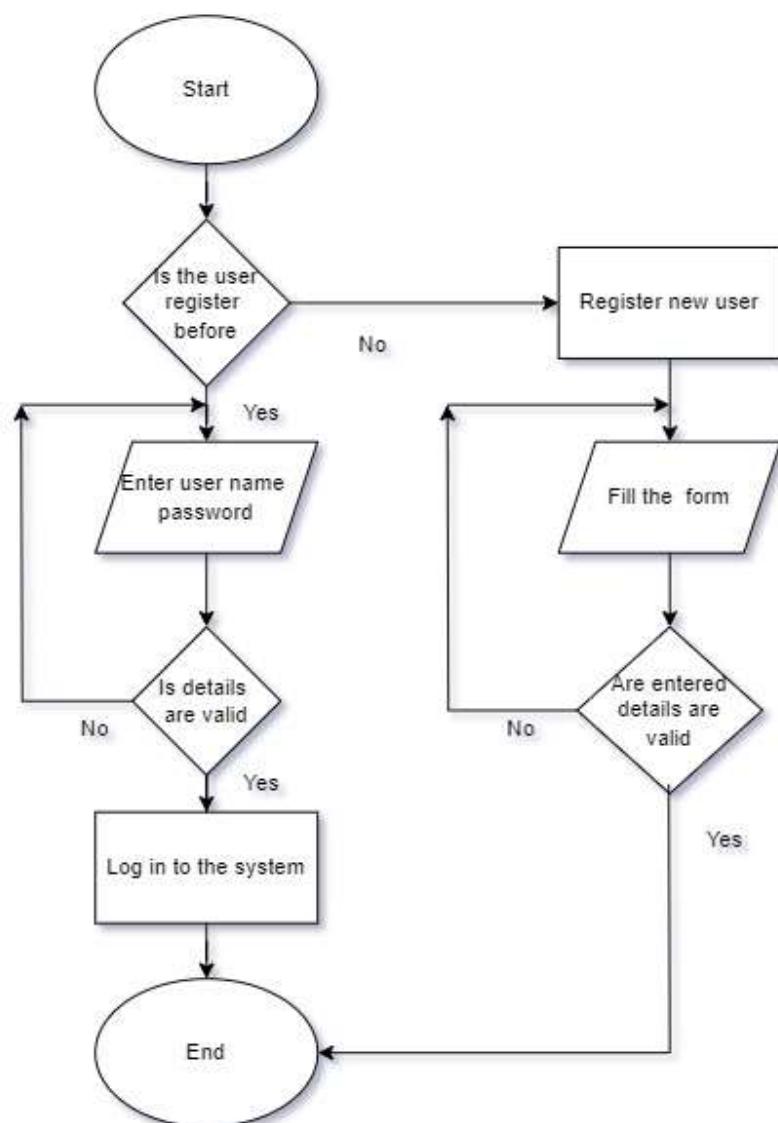


Figure 2:

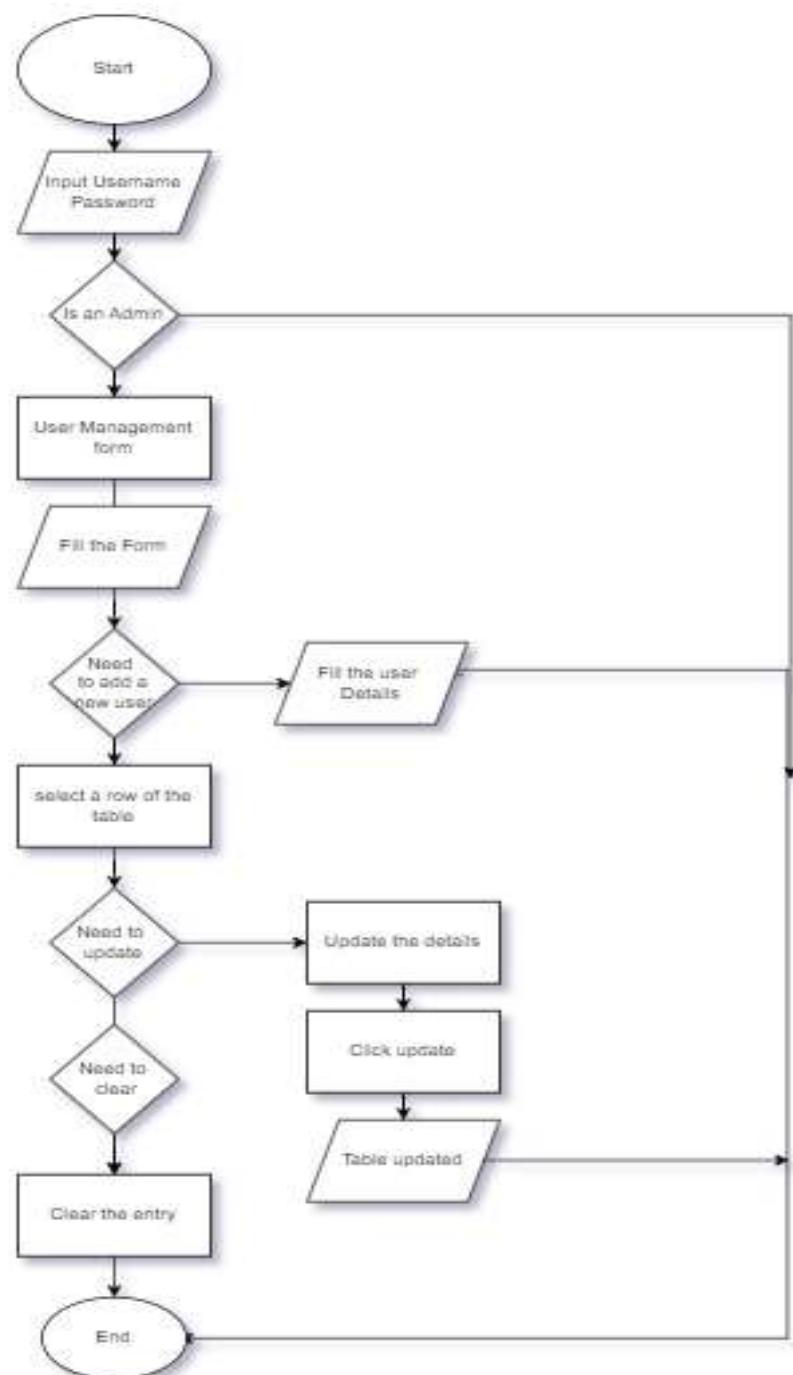


Figure 3:

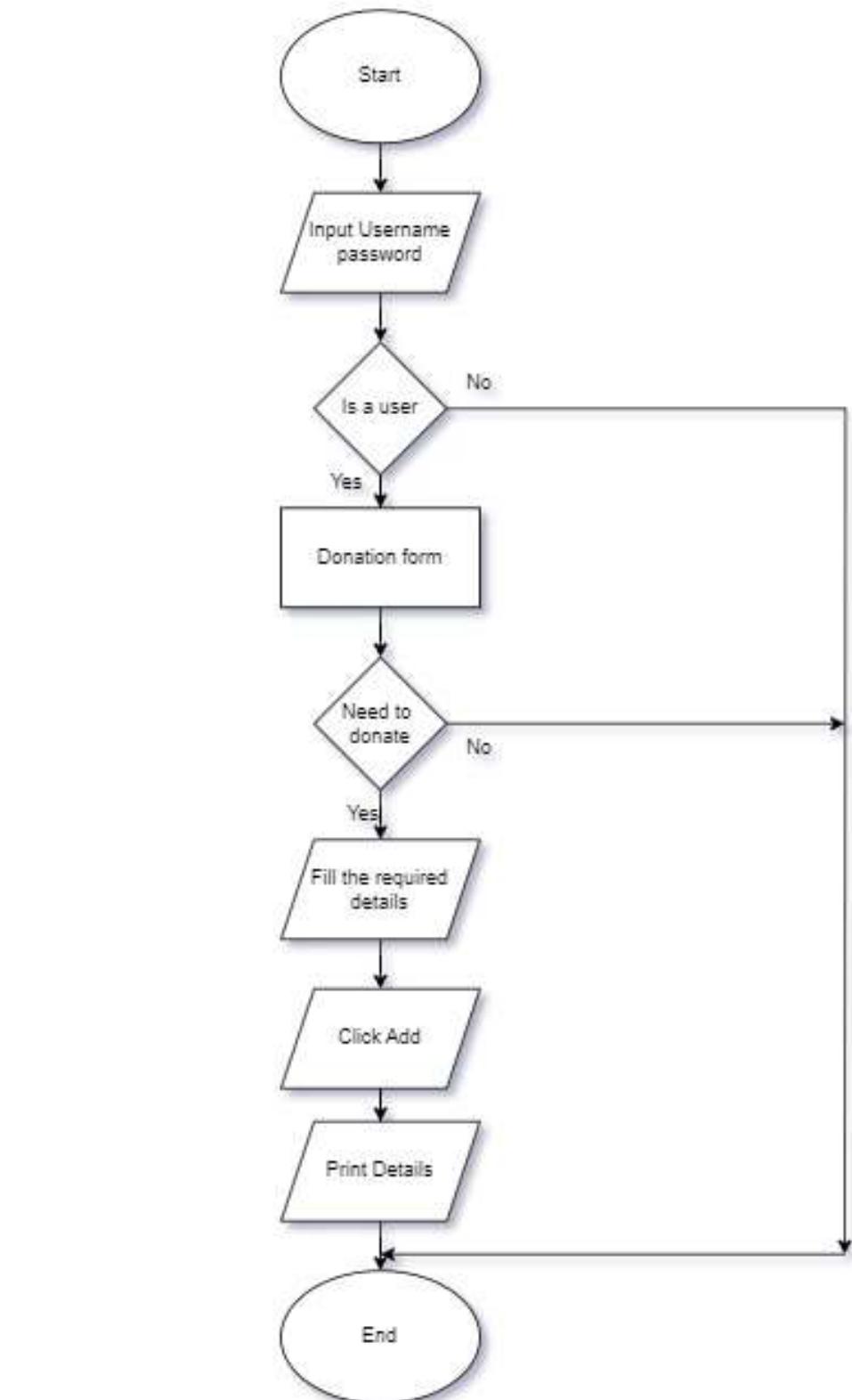


Figure 4:

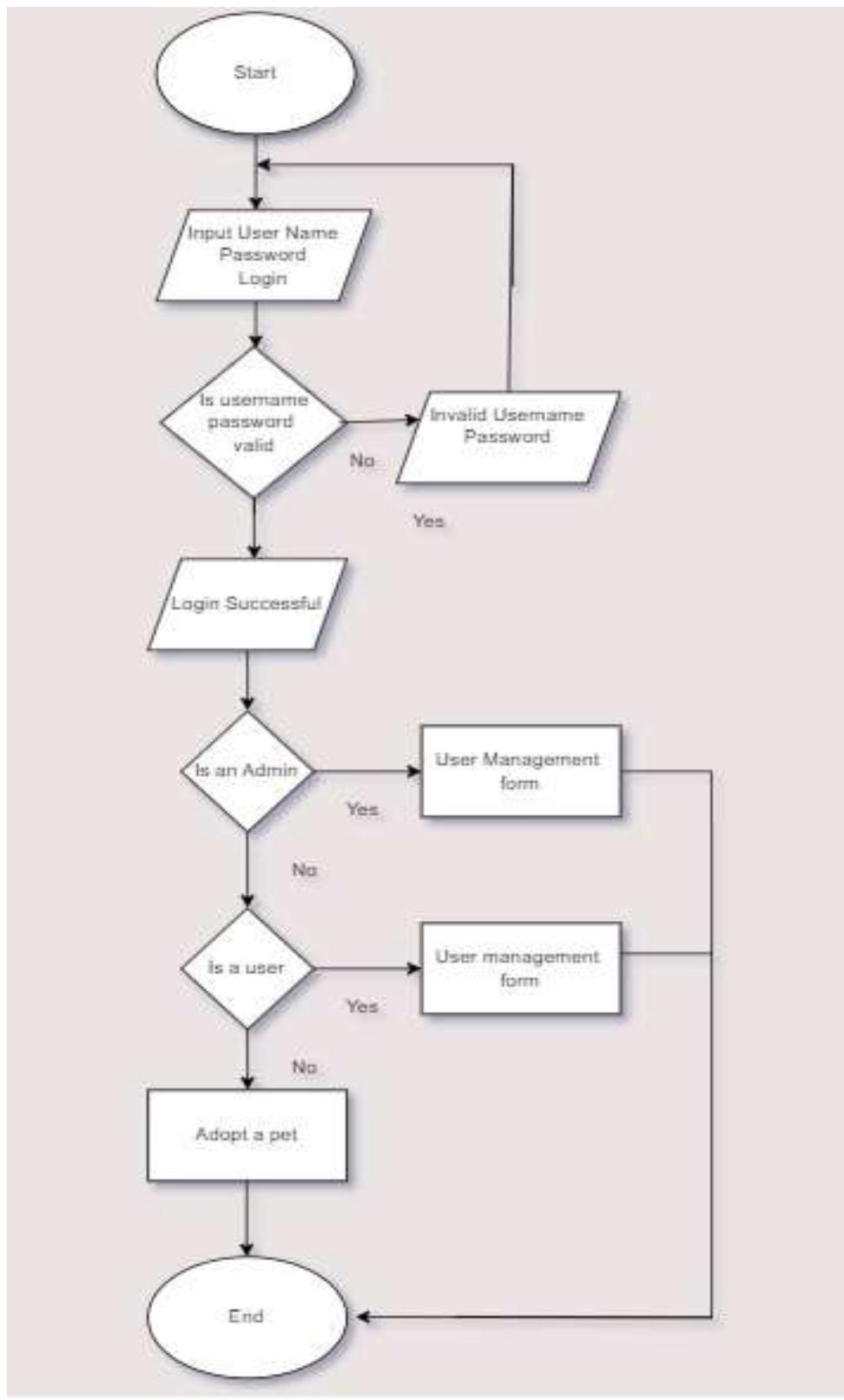
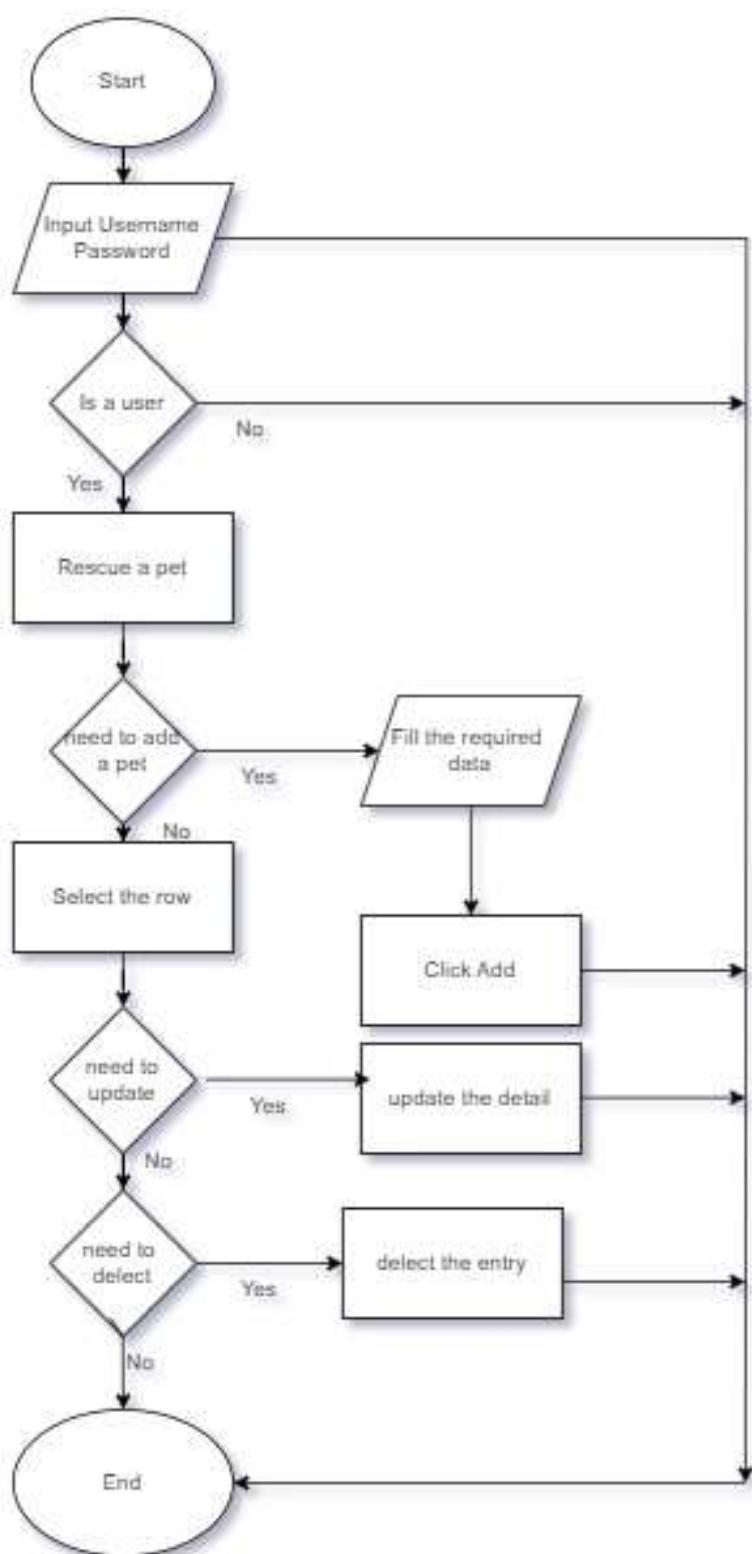


Figure 5:



4.2 PSEUDOCODES

Figure1:

START

DISPLAY "Welcome to the Pet Adoption System!"

LOOP UNTIL user decides to adopt

ASK "Would you like to adopt a pet (yes/no)?"

READ userDecision

IF userDecision == "yes" THEN

SET userCanAdopt to TRUE

ENDIF

END LOOP

IF userCanAdopt THEN

DISPLAY Available Pets (by species, age, etc.)

LOOP UNTIL user selects a pet

ASK "Enter the ID of the pet you'd like to adopt:"

READ petID

IF ValidPetID(petID) THEN

SET selectedPet to GetPetDetails(petID)

ELSE

DISPLAY "Invalid pet ID. Please try again."

ENDIF

END LOOP

IF EligibleToAdopt(selectedPet) THEN

```
DISPLAY "Congratulations! You appear eligible to adopt this pet."
IF SubmitAdoptionApplication(selectedPet) THEN
    shelterRunsBackgroundCheck(user)
    LOOP UNTIL Application Status Known
    SET applicationStatus to GetApplicationStatus()
END LOOP

IF applicationStatus == "approved" THEN
    DISPLAY "Your application to adopt ${selectedPet.name} has been approved!"
    PayAdoptionFee(selectedPet.adoptionFee)
    FinalizeAdoption(selectedPet, user)
    DISPLAY "Congratulations! ${selectedPet.name} is now your pet!"
ELSE
    DISPLAY "We regret to inform you that your application to adopt ${selectedPet.name} has
been denied."
ENDIF

ELSE
    DISPLAY "An error occurred while submitting your application. Please try again later."
ENDIF

ELSE
    DISPLAY "We're sorry, you do not meet the criteria to adopt this pet."
ENDIF

ENDIF

DISPLAY "Thank you for using the Pet Adoption System!"
```

END

Figure2:

```
FUNCTION AdoptPet()
BEGIN
    DISPLAY "Welcome to the Pet Adoption System!"
    DO WHILE (user Doesn't Decide to Adopt)
        DISPLAY "Would you like to adopt a pet (yes/no)?"
        READ userDecision
        IF userDecision == "yes" THEN
            userCanAdopt := TRUE
        ENDIF
    END WHILE
    IF userCanAdopt THEN
        DISPLAY Available Pets (by species, age, etc.)

        DO WHILE (user Hasn't Selected Pet)
            DISPLAY "Enter the ID of the pet you'd like to adopt:"
            READ petID
            IF ValidPetID(petID) THEN
                selectedPet := GetPetDetails(petID)
            ELSE
                DISPLAY "Invalid pet ID. Please try again."
           ENDIF
        END WHILE
        IF EligibleToAdopt(selectedPet) THEN
            DISPLAY "Congratulations! You appear eligible to adopt this pet."
            applicationSubmitted := SubmitAdoptionApplication(selectedPet)
            IF applicationSubmitted THEN
                shelterRunsBackgroundCheck(user)
                DO WHILE (Application Status Unknown)
                    applicationStatus := GetApplicationStatus()
                END WHILE
            END IF
        END IF
    END IF
END FUNCTION
```

```

END WHILE

IF applicationStatus == "approved" THEN

    DISPLAY "Your application to adopt ${selectedPet.name} has been approved!"
    PayAdoptionFee(selectedPet.adoptionFee)
    FinalizeAdoption(selectedPet, user)
    DISPLAY "Congratulations! ${selectedPet.name} is now your pet!"

ELSE

    DISPLAY "We regret to inform you that your application to adopt ${selectedPet.name} has been denied."

ENDIF

ELSE

    DISPLAY "An error occurred while submitting your application. Please try again later."

ENDIF

ELSE

    DISPLAY "We're sorry, you do not meet the criteria to adopt this pet."

ENDIF

ENDIF

DISPLAY "Thank you for using the Pet Adoption System!"

END

```

Figure 3:

```

BEGIN

DISPLAY "Welcome to the Pet Adoption System!"

WHILE user hasn't decided to adopt

    ASK "Would you like to adopt a pet (yes/no)?"

    READ userDecision

    IF userDecision is "yes"

        SET userCanAdopt to TRUE

```

```

ENDIF

END WHILE

IF userCanAdopt

DISPLAY available pets (by species, age, etc.)

WHILE user hasn't selected a pet

ASK "Enter the ID of the pet you'd like to adopt:"

READ petID

IF valid pet ID (petID)

SET selectedPet to get pet details (petID)

ELSE

DISPLAY "Invalid pet ID. Please try again."

ENDIF

END WHILE

IF eligible to adopt (selectedPet)

DISPLAY "Congratulations! You appear eligible to adopt this pet."

IF submit adoption application (selectedPet)

shelter runs background check (user)

WHILE application status unknown

SET applicationStatus to get application status()

ENDIF

IF applicationStatus is "approved"

DISPLAY "Your application to adopt ${selectedPet.name} has been approved!"

pay adoption fee (selectedPet.adoptionFee)

finalize adoption (selectedPet, user)

DISPLAY "Congratulations! ${selectedPet.name} is now your pet!"

ELSE

```

```

    DISPLAY "We regret to inform you that your application to adopt ${selectedPet.name} has
been denied."
ENDIF

ELSE
DISPLAY "An error occurred while submitting your application. Please try again later."
ENDIF

ELSE
DISPLAY "We're sorry, you do not meet the criteria to adopt this pet."
ENDIF

ENDIF

DISPLAY "Thank you for using the Pet Adoption System!"

END

```

Figure 4:

```

BEGIN

DISPLAY "Welcome to the Pet Adoption System!"

WHILE user hasn't decided to adopt

    ASK "Would you like to adopt a pet (yes/no)?"

    READ userDecision

    IF userDecision is "yes"

        SET userCanAdopt to TRUE

    ENDIF

END WHILE

IF userCanAdopt

    DISPLAY available pets (by species, age, etc.)

```

```

WHILE user hasn't selected a pet

ASK "Enter the ID of the pet you'd like to adopt:"

READ petID

IF valid pet ID (petID)

    SET selectedPet to get pet details (petID)

ELSE

    DISPLAY "Invalid pet ID. Please try again."

ENDIF

END WHILE

IF eligible to adopt (selectedPet)

    DISPLAY "Congratulations! You appear eligible to adopt this pet."

    IF submit adoption application (selectedPet)

        shelter runs background check (user)

        WHILE application status unknown

            SET applicationStatus to get application status()

        END WHILE

        IF applicationStatus is "approved"

            DISPLAY "Your application to adopt ${selectedPet.name} has been approved!"

            pay adoption fee (selectedPet.adoptionFee)

            finalize adoption (selectedPet, user)

            DISPLAY "Congratulations! ${selectedPet.name} is now your pet!"

        ELSE

            DISPLAY "We regret to inform you that your application to adopt ${selectedPet.name} has been denied."

        ENDIF

    ELSE


```

```
DISPLAY "An error occurred while submitting your application. Please try again later."  
ENDIF  
  
ELSE  
  
DISPLAY "We're sorry, you do not meet the criteria to adopt this pet."  
ENDIF  
  
ENDIF  
  
DISPLAY "Thank you for using the Pet Adoption System!"  
END
```

Figure 5:

```
BEGIN  
  
DISPLAY "Welcome to the Pet Adoption System!"  
  
WHILE user hasn't decided to adopt  
    ASK "Would you like to adopt a pet (yes/no)?"  
    READ userDecision  
    IF userDecision is "yes"  
        SET userCanAdopt to TRUE  
    ENDIF  
END WHILE  
  
IF userCanAdopt  
    DISPLAY available pets (by species, age, etc.)  
  
WHILE user hasn't selected a pet
```

ASK "Enter the ID of the pet you'd like to adopt:"

READ petID

IF valid pet ID (petID)

SET selectedPet to get pet details (petID)

ELSE

DISPLAY "Invalid pet ID. Please try again."

ENDIF

END WHILE

IF eligibleToAdopt(selectedPet)

DISPLAY "Congratulations! You appear eligible to adopt this pet."

IF submitAdoptionApplication(selectedPet)

shelterRunsBackgroundCheck(user)

WHILE applicationStatus unknown

SET applicationStatus = getAdoptionApplicationStatus()

END WHILE

IF applicationStatus is "approved"

DISPLAY "Your application to adopt \${selectedPet.name} has been approved!"

payAdoptionFee(selectedPet.adoptionFee)

finalizeAdoption(selectedPet, user)

DISPLAY "Congratulations! \${selectedPet.name} is now your pet!"

ELSE

DISPLAY "We regret to inform you that your application to adopt \${selectedPet.name} has been denied."

ENDIF

ELSE

DISPLAY "An error occurred while submitting your application. Please try again later."

ENDIF

ELSE

DISPLAY "We're sorry, you do not meet the criteria to adopt this pet."

ENDIF

ENDIF

DISPLAY "Thank you for using the Pet Adoption System!"

END

5. SYSTEM IMPLEMENTATION

5.1 INTERFACES

Selected number of user interfaces of the system are as follows.

5.1.1 Loading Form



5.1.2 Home Form

A screenshot of the AdoptAPet home page. On the left is a vertical sidebar with icons and labels: "Home" (house), "Dashboard" (chart), "Rescue a pet" (hand holding heart), "Adopt a pet" (hand holding paw), "Donation" (piggy bank), and "Logout" (corgi). The main content area has a title "Helping Street Animals Find Loving Homes..." in red. Below it is a mission statement: "Our Mission: We connect homeless cats and dogs in Sri Lanka with caring adopters". It then describes "How it Works:" with two bullet points: "- Report sightings of homeless animals." and "- Connect with potential adopters.". At the bottom, it encourages users to "Together, let's create a brighter future for Sri Lanka's street animals". There is another small dog and cat icon. At the very bottom, there is a contact section with a phone icon, the text "Contact Us", and the phone numbers "011 222 2222 / 077 183 3956".

5.1.3 Login Form

The screenshot shows a mobile application interface. On the left, there is a "USER LOGIN" section with a paw print icon. It contains fields for "Username" and "Password", each with a red placeholder bar. Below these are "LOGIN" and "CLEAR" buttons. At the bottom, a blue link reads "Don't have an account? SIGN UP". On the right, there is a large image of two dogs, one brown and one black and white, standing together. Above the image, the text "Open your heart, open a home..." is visible, along with a small dog icon.

5.1.4 Signup Form

The screenshot shows a mobile application interface. On the left, there is a promotional message: "Find your perfect match. Register to adopt!" above a large image of a dog and a cat together. On the right, there is a "SIGNUP FORM" section with a "REGISTER" button at the top. It contains fields for "Name", "User Name", "Password", and "Confirm Password", each with a red placeholder bar. Below these is a field for "Job Role (Admin / User / Guest)" with a red placeholder bar. At the bottom, there are "SUBMIT" and "CLEAR" buttons, and a "Back" button in the bottom right corner.

5.1.5 User Management Form

USER MANAGEMENT

User ID:

Name:

User Name:

Password:

Job Role:

NEW **UPDATE**

DELETE **CLEAR**

[Back to Home](#)

5.1.6 Rescue a pet Form

 **Rescue a pet**

ADD **UPDATE** **DELETE** **CLEAR**

Pet_ID:

Pet type (cat/dog):

Suggest a name (If you like):

Age (To be like):

Gender (If you can recognize):

If any disability:

Location (With a landmark):

Any other characteristics: (colour/behaviour...)

User Details

User's name:

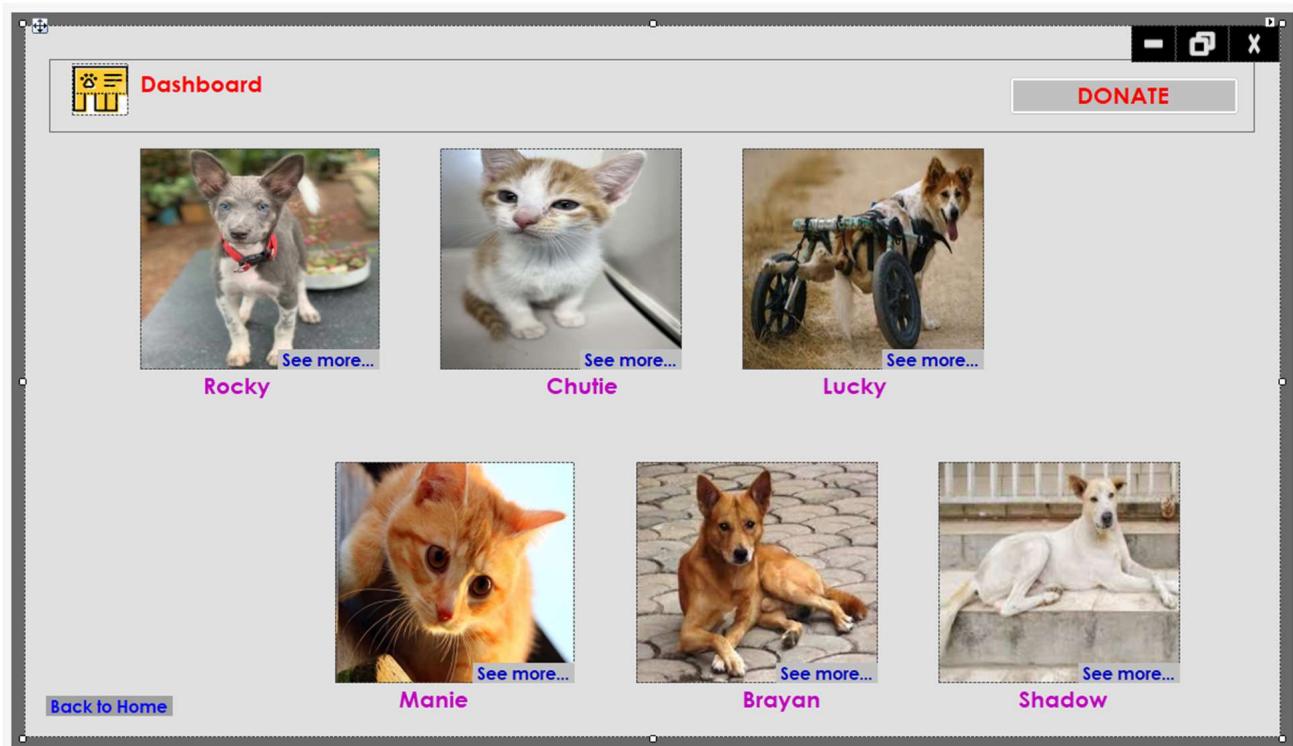
NIC Number:

Tel Number:

[Like to Adopt](#)

[Back to Home](#)

5.1.7 Dashboard Form



5.1.8 Donation and printing preview Form

This form allows users to enter donation details. It includes fields for Pet Name, Name of the Donator, Bank Account No., Bank, and Branch. There are also buttons for ADD, DELETE, and CLEAR, along with a Search function. A large central area is provided for previewing the donation document.

Pet Name *	Name of the Donator *	Bank Account No.
<input type="text"/>	<input type="text"/>	<input type="text"/>
Pet Owner	Amount like to Donate (Rs.) *	Bank
<input type="text"/>	<input type="text"/>	<input type="text"/>
		Branch
		<input type="text"/>

Print

Total:

Bill No.:

Back to Home

printDocument1 printPreviewDialog1

5.2 Important Codes

5.2.1 Generate a unique pet ID and resequencing pet ID when a entry is deleted

```
3 references
private int GeneratePetID()
{
    return GetMaxID() + 1;
}

1 reference
private int GetMaxID()
{
    int maxID = 0;
    string query = "SELECT MAX(Pet_ID) FROM Rescue_Pet_Table";

    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        using (SqlCommand cmd = new SqlCommand(query, conn))
        {
            try
            {
                conn.Open();
                object result = cmd.ExecuteScalar();
                if (result != DBNull.Value)
                {
                    maxID = Convert.ToInt32(result);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            finally
            {
                conn.Close();
            }
        }
    }
}

309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
887
```

5.2.2 Insert Pet details to the database table

```
75
76
77     string query = "INSERT INTO Rescue_Pet_Table (Pet_ID, Pet_Type, Name, Age, Gender, Disability, Location, Characteristics, User_Name, NIC, Telephone_Number) " +
78         "@VALUES (@Pet_ID, @Pet_Type, @Name, @Age, @Gender, @Disability, @Location, @Characteristics, @User_Name, @NIC, @Tel_Number)";
79
80     using (SqlConnection conn = new SqlConnection(connectionString))
81     {
82         using (SqlCommand cmd = new SqlCommand(query, conn))
83         {
84             cmd.Parameters.AddWithValue("@Pet_ID", pet_ID);
85             cmd.Parameters.AddWithValue("@Pet_Type", type);
86             cmd.Parameters.AddWithValue("@Name", name);
87             cmd.Parameters.AddWithValue("@Age", age);
88             cmd.Parameters.AddWithValue("@Gender", gender);
89             cmd.Parameters.AddWithValue("@Disability", disability);
90             cmd.Parameters.AddWithValue("@Location", location);
91             cmd.Parameters.AddWithValue("@Characteristics", characteristics);
92             cmd.Parameters.AddWithValue("@User_Name", username);
93             cmd.Parameters.AddWithValue("@NIC", nicnumber);
94             cmd.Parameters.AddWithValue("@Tel_Number", telnumber);
95
96             try
97             {
98                 conn.Open();
99                 cmd.ExecuteNonQuery();
100                LoadData();
101                ClearForm();
102            }
103            catch (Exception ex)
104            {
105                MessageBox.Show("Error: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
106            }
107            finally
108            {
109                conn.Close();
110            }
111        }
112    }
113
114
115     private int GeneratePetID()
116     {
```

```
2 references
55     private void btnAdd_Click(object sender, EventArgs e)
56     {
57         int pet_ID = GeneratePetID();
58         string type = txtType.Text.Trim();
59         string name = txtName.Text.Trim();
60         string age = txtAge.Text.Trim();
61         string gender = comboGender.Text.Trim();
62         string disability = txtDisability.Text.Trim();
63         string location = txtLocation.Text.Trim();
64         string characteristics = txtCharacteristics.Text.Trim();
65         string username = txtUserName.Text.Trim();
66         string nicnumber = txtNICNumber.Text.Trim();
67         string telnumber = txtTelNumber.Text.Trim();
68
69
70         if (string.IsNullOrWhiteSpace(type) || string.IsNullOrWhiteSpace(name) || string.IsNullOrWhiteSpace(age) || string.IsNullOrEmpty(gender) || string.IsNullOrWhiteSpace(disability))
71         {
72             MessageBox.Show("Please fill in all required fields.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
73             return;
74         }
```

5.2.3 Update pet details

```
197
198     2 references
199     private void btnUpdate_Click(object sender, EventArgs e)
200     {
201         if (int.TryParse(txtPetID.Text, out int petID))
202         {
203             string type = txtType.Text.Trim();
204             string name = txtName.Text.Trim();
205             string age = txtAge.Text.Trim();
206             string gender = combGender.Text.Trim();
207             string disability = txtDisability.Text.Trim();
208             string location = txtLocation.Text.Trim();
209             string characteristics = txtCharacteristics.Text.Trim();
210             string username = txtUserName.Text.Trim();
211             string nicnumber = txtNICNumber.Text.Trim();
212             string telnumber = txtTelNumber.Text.Trim();
213
214             if (string.IsNullOrWhiteSpace(type) || string.IsNullOrWhiteSpace(name) || string.IsNullOrWhiteSpace(age) || string.IsNullOrEmpty(gender) || string.IsNullOrWhiteSpace(disability) || string.IsNullOrWhiteSpace(location) || string.IsNullOrWhiteSpace(characteristics) || string.IsNullOrWhiteSpace(username) || string.IsNullOrWhiteSpace(nicnumber) || string.IsNullOrWhiteSpace(telnumber))
215             {
216                 MessageBox.Show("Please fill in all required fields.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
217                 return;
218             }
219
220             // Update existing pet record in the database
221             string query = "UPDATE Rescue_Pet_Table SET Pet_Type = @Pet_Type, Name = @Name, Age = @Age, Gender = @Gender, Disability = @Disability, Location = @Location, Characteristics = @Characteristics, User_Name = @User_Name WHERE Pet_ID = @Pet_ID";
222
223             using (SqlConnection conn = new SqlConnection(connectionString))
224             {
225                 using (SqlCommand cmd = new SqlCommand(query, conn))
226                 {
227                     cmd.Parameters.AddWithValue("@Pet_ID", petID);
228                     cmd.Parameters.AddWithValue("@Pet_Type", type);
229                     cmd.Parameters.AddWithValue("@Name", name);
230                     cmd.Parameters.AddWithValue("@Age", age);
231                     cmd.Parameters.AddWithValue("@Gender", gender);
232                     cmd.Parameters.AddWithValue("@Disability", disability);
233                     cmd.Parameters.AddWithValue("@Location", location);
234                     cmd.Parameters.AddWithValue("@Characteristics", characteristics);
235                     cmd.Parameters.AddWithValue("@User_Name", username);
236                     cmd.Parameters.AddWithValue("@NIC", nicnumber);
237                     cmd.Parameters.AddWithValue("@Tel_Number", telnumber);
238
239                     try
240                     {
241                         conn.Open();
242                         cmd.ExecuteNonQuery();
243                         LoadData();
244                     }
245                     catch (Exception ex)
246                     {
247                         MessageBox.Show("Error: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
248                     }
249                 }
250             }
251         }
252     }
253
254 }
```

5.2.4 Search Pet names

```
241
242
243
244     1 reference
245     private void textBox7_TextChanged(object sender, EventArgs e)
246     {
247         dv.RowFilter = $"Pet_Name LIKE '{textBox7.Text}%' OR Name_of_the_Donor LIKE '{textBox7.Text}%' OR Convert(Amount, 'System.String')";
248
249     1 reference
250     private void label9_Click(object sender, EventArgs e)
251     {
252         Form3 form3 = new Form3();
253         form3.Show();
254         this.Hide();
255     }
256 }
```

5.2.5 Close, minimize and maximize the windows

```
255 // Minimize the window  
256 1 reference  
257 private void pictureBox11_Click(object sender, EventArgs e)  
258 {  
259     this.WindowState = FormWindowState.Minimized;  
260 }  
261 // Maximize the window  
262 1 reference  
263 private void pictureBox10_Click(object sender, EventArgs e)  
264 {  
265     this.WindowState = FormWindowState.Maximized;  
266 }  
267 // Close the window  
268 1 reference  
269 private void pictureBox12_Click(object sender, EventArgs e)  
270 {  
271     Application.Exit();  
272 }  
273 }  
274 }  
275 }
```

5.2.6 Print function

```
221 1 reference  
222 private void button5_Click(object sender, EventArgs e)  
223 {  
224     PrintInvoice();  
225 }  
226  
227 1 reference  
228 private void PrintInvoice()  
229 {  
230     try  
231     {  
232         PrintDialog pd = new PrintDialog();  
233         pd.Document = printDocument1;  
234         if (pd.ShowDialog() == DialogResult.OK)  
235         {  
236             printDocument1.Print();  
237         }  
238     catch (Exception ex)  
239     {  
240         MessageBox.Show($"An error occurred while printing: {ex.Message}", "Print Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  
241     }  
242 }  
243 }
```

```

149
150     2 references
151     private void GenerateBillNumber()
152     {
153         billNumber++;
154         label7.Text = $"Bill No: {billNumber}";
155     }
156
157     2 references
158     private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
159     {
160         Image image = Properties.Resources.shutterstock_1128546527;
161         e.Graphics.DrawImage(image, 320, 20, 200, 80);
162
163         e.Graphics.DrawString("Invoice", new Font("Arial", 20, FontStyle.Bold), Brushes.Black, new Point(25, 150));
164
165         e.Graphics.DrawString("Date: " + DateTime.Now.ToShortDateString(), new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(25, 180));
166         e.Graphics.DrawString("Customer Name: " + textBox5.Text, new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(25, 210));
167         e.Graphics.DrawString(label7.Text, new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(25, 240));
168
169         int y = 280;
170
171         e.Graphics.DrawString("No", new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(26, y));
172         e.Graphics.DrawString("Pet Name", new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(101, y));
173         e.Graphics.DrawString("Name of the Donator", new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(301, y));
174         e.Graphics.DrawString("Amount", new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(635, y));
175
176         y += 30;
177
178         e.Graphics.DrawString("-----");
179         y += 15;
180
181         int No = 1;
182         foreach (DataGridViewRow row in dataGridView1.Rows)
183         {
184             string petname = row.Cells["Pet_Name"].Value?.ToString();
185             string donator = row.Cells["Name_of_the_Donator"].Value?.ToString();
186             string amount = row.Cells["Amount"].Value?.ToString();
187
188             e.Graphics.DrawString(No.ToString(), new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(26, y));
189             e.Graphics.DrawString(petname, new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(101, y));
190             e.Graphics.DrawString(donator, new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(301, y));
191             e.Graphics.DrawString(amount, new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(635, y));
192
193             No++;
194             y += 25;
195         }
196
197         e.Graphics.DrawString("-----");
198         y += 25;
199
200         e.Graphics.DrawString(lblAmount.Text, new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(630, y));
201     }
202
203     1 reference
204     private void printDocument1_EndPrint(object sender, System.Drawing.Printing.PrintEventArgs e)
205     {
206         MessageBox.Show("Print successfully...", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
207     }
208
209     0 references
210     private void button4_Click(object sender, EventArgs e)
211     {
212         GenerateBillNumber();
213         try
214         {
215             printPreviewDialog1.ShowDialog();
216             MessageBox.Show("Print Successfully!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
217         }
218         catch (Exception ex)
219         {
220             MessageBox.Show($"An error occurred while printing: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
221         }
222     }
223
224     1 reference
225     private void button5_Click(object sender, EventArgs e)
226     {
227         PrintInvoice();
228     }

```

5.2.2 Generate Bill number for the invoice

```

208
209     0 references
210     private void button4_Click(object sender, EventArgs e)
211     {
212         GenerateBillNumber();
213         try
214         {
215             printPreviewDialog1.ShowDialog();
216             MessageBox.Show("Print Successfully!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
217         }
218         catch (Exception ex)
219         {
220             MessageBox.Show($"An error occurred while printing: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
221         }
222     }
223
224     1 reference
225     private void button5_Click(object sender, EventArgs e)
226     {
227         PrintInvoice();
228     }

```

```
150      2 references
151      private void GenerateBillNumber()
152      {
153          billNumber++;
154          label7.Text = $"Bill No: {billNumber}";
155      }
156      2 references
157      private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
158      {
159          Image image = Properties.Resources.shutterstock_1128546527;
160          e.Graphics.DrawImage(image, 320, 20, 200, 80);
161
162          e.Graphics.DrawString("Invoice", new Font("Arial", 20, FontStyle.Bold), Brushes.Black, new Point(25, 150));
163
164          e.Graphics.DrawString("Date: " + DateTime.Now.ToString("MM/dd/yyyy"), new Font("Arial", 15, FontStyle.Regular), Brushes.Black, new Point(25, 180));
165      }
166  
```

6. REFERENCES

1. [Embark Passion: About Us]"(<https://embarkpassion.com/category/news/>). Their primary focus is on rescue and rehoming initiatives, including weekly adoption days (<https://embarkpassion.com/colombo-adoption-day/>).
2. Ceylon Society for the Protection of Animals (CASPA): [CASPA](#)
3. Blue Paws: Primarily in Sinhala, but offers some information in English as well: [Blue Paws](#)
4. "Sri Lanka's Stray Dog Crisis: A Growing Problem" by The Diplomat ([The Diplomat, Sri Lanka's Stray Dog Crisis: A Growing Problem](#)). (2022, September 21)
5. "Plight of stray animals in Sri Lanka: A never-ending battle" by Ceylon Today ([Ceylon Today, Plight of stray animals in Sri Lanka: A never-ending battle](#)). (1999, September 23)
6. "Animal Welfare in Sri Lanka" by Lanka Business News ([Lanka Business News, Animal Welfare in Sri Lanka](#)). (2023, August 1)
7. Microsoft. (n.d.). *Visual Studio 2022 Documentation*.
<https://learn.microsoft.com/en-us/visualstudio/windows/?view=vs-2022>
8. Microsoft. (n.d.). *Microsoft SQL Server Express 2022 Documentation*.
<https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>
9. World Wide Web Consortium (W3C). (n.d.). *Web Accessibility Initiative (WAI)*.
<https://www.w3.org/WAI/>
10. National Institute of Standards and Technology (NIST). (2020, December). *Special Publication 800-63B: Digital Identity Authentication Guidance*. [National Institute of Standards and Technology (.gov), pages.nist.gov]

Appendix A

More Forms

1. Adopt a pet form

The screenshot shows a window titled "Adopt a pet". On the left is a large image of a brown and white puppy named Rocky. To the right of the image is a text area with a small icon of hands holding a heart. The text reads:

Homeless cats and dogs are a big problem in Sri Lanka.
There's no good way to find these animals new homes.
A website could be made to connect homeless pets with people who want to adopt them.
This would help the animals and make Sri Lanka a more caring place.

Below this is a blue banner with the text "Together, we can make a difference for homeless animals!". Underneath is a note: "After you adopt a pet in the list, please make sure to update the table by DELETING the relevant entry". To the right is a decorative graphic of a cat and a dog sitting together with butterflies and paw prints. At the bottom left is a "Back to Home" button.

2. Dashboard form

The screenshot shows a window titled "Dashboard". At the top right is a "DONATE" button. Below it are four cards with animal photos and names: "Rocky" (a puppy), "Chutie" (a kitten), "Lucky" (a dog in a wheelchair), and "Manie" (an orange cat). Each card has a "See more..." link at the bottom right. Below these are two more cards: "Brayan" (a brown dog) and "Shadow" (a white dog). Each of these also has a "See more..." link. At the bottom left is a "Back to Home" button.

3. Pet profile details forms

Rocky

The puppy looks like a mixed breed dog.
It's gray and white with a docked tail.
Puppy is very playful and was found at a temple in Balangoda.

[Back to Dashboard](#)



Chutie

The kitten is with orange and white fur..
Found it's perfect home!



Lucky

Wheelchair was worn after adopting this baby from a railway station.
All smiles on a joyride!
This happy dog proves that anything is possible with a little help.

[Back to Dashboard](#)



Brayan

Very innocent babay dog.
Found it's purfect home!
Dogs' lives are short, too short!



Shadow

Puppy with bright eyes and a wagging tail.
Life is better with a pet by your side.



- | X
Back to Dashboard

Manie

A close-up photo of a tabby kitten with orange and brown fur.
Found it's purfect home!



More Codes

1. Login Details

```
1 reference
29  private void button1_Click(object sender, EventArgs e)
30  {
31      String username, password, role;
32      username = textBox1.Text;
33      password = textBox2.Text;
34
35      try
36      {
37          String query = "SELECT * FROM User_Table_PetAdoption WHERE username = @Username AND password = @Password";
38          SqlDataAdapter sda = new SqlDataAdapter(query, conn);
39          sda.SelectCommand.Parameters.AddWithValue("@Username", username);
40          sda.SelectCommand.Parameters.AddWithValue("@Password", password);
41
42          DataTable dataTable = new DataTable();
43          sda.Fill(dataTable);
44
45          if (dataTable.Rows.Count > 0 )
46          {
47              role = dataTable.Rows[0]["Role"].ToString();
48
49              switch (role.ToLower())
50              {
51                  case "admin":
52                      Form6 form6 = new Form6();
53                      form6.Show();
54                      break;
55
56                  case "guest":
57                      Form3 form3 = new Form3();
58                      form3.Show();
59                      break;
56
59
60                  case "user":
61                      Form5 form5 = new Form5();
62                      form5.Show();
63                      break;
64
65                  default:
66                      MessageBox.Show("Unknown user", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
67                      break;
68
69
70              }
71          }
72      }
73  }
```

2. Calculate Total price of the invoice

```
114  | references
115  | private void CalculateTotalPrice()
116  | {
117  |     double sum = 0;
118  |     foreach (DataGridViewRow row in dataGridView1.Rows)
119  |     {
120  |         if (row.Cells["Amount"].Value != null && double.TryParse(row.Cells["Amount"].Value.ToString(), out double amount))
121  |         {
122  |             sum += amount;
123  |         }
124  |     }
125  |
126  |     lblAmount.Text = $"Total: Rs.{sum:F2}";
127  |
128  |     1 reference
129  |     private void button2_Click(object sender, EventArgs e)
130  |     {
131  |         if (dataGridView1.SelectedRows.Count > 0)
132  |         {
133  |             foreach (DataGridViewRow row in dataGridView1.SelectedRows)
134  |             {
135  |                 dataGridView1.Rows.Remove(row);
136  |             }
137  |             CalculateTotalPrice();
138  |         }
139  |         else
140  |         {
141  |             MessageBox.Show("Please select a row to delete.");
142  |         }
143  |     }
144  | }
```

3. Load Pet owners

```
41   | 1 reference
42   | private void LoadCombo()
43   | {
44   |     try
45   |     {
46   |         conn.Open();
47   |         string query = "SELECT * FROM Table_Donation";
48   |         SqlCommand cmd = new SqlCommand(query, conn);
49   |         SqlDataAdapter sda = new SqlDataAdapter(cmd);
50   |
51   |         DataTable comboDt = new DataTable();
52   |         sda.Fill(comboDt);
53   |
54   |         comboBox1.DataSource = comboDt;
55   |         comboBox1.DisplayMember = "Pet_Name";
56   |         comboBox1.ValueMember = "Pet_ID";
57   |         comboBox1.SelectedIndex = -1;
58   |     }
59   |     catch (Exception ex)
60   |     {
61   |         MessageBox.Show("Error: " + ex.Message);
62   |     }
63   |     finally
64   |     {
65   |         conn.Close();
66   |     }
67   | }
```

```

68     | reference
69     | {
70     |     private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
71     |     {
72     |         if (comboBox1.SelectedIndex != -1 && ((DataTable)comboBox1.DataSource).Rows.Count > 0)
73     |         {
74     |             DataRow dr = ((DataRowView)comboBox1.SelectedItem).Row;
75     |             textBox2.Text = dr["Pet_Owner"].ToString();
76     |             textBox1.Text = dr["Bank_Account"].ToString();
77     |             textBox3.Text = dr["Bank"].ToString();
78     |             textBox4.Text = dr["Branch"].ToString();
79     |         }
80     |         else
81     |         {
82     |             textBox2.Text = string.Empty;
83     |             textBox1.Text = string.Empty;
84     |             textBox3.Text = string.Empty;
85     |             textBox4.Text = string.Empty;
86     |         }
87     |     }
88     |
89     | }

```

Created SQL Server management Database tables

1. User table

The screenshot shows the Object Explorer on the left with a tree view of database objects like Login, Database Diagrams, Tables, System Tables, etc. The User table is selected. To the right, a grid view displays the contents of the User table:

	User...	Name	Username	Password	Role
1	Pooja	pooja	1234	Admin	
2	Uma...	umangani	5678	Guest	
3	Neja	neja	1212	User	
4	Tharu	tharu	4545	User	
5	Senura	senura	4545	User	
6	Sherin	sherin	8989	Guest	
.	NULL	NULL	NULL	NULL	NULL

2. Pet owner details

The screenshot shows the Object Explorer on the left with a tree view of database objects like Login, Database Diagrams, Tables, System Tables, etc. The Pet owner table is selected. To the right, a grid view displays the contents of the Pet owner table:

	Pet_ID	Pet_Name	Pet_owner	Bank_Account	Bank	Branch
1	Rocky	A.S Peiris	8006495689	Amana	Gampaha	
2	Chutie	K.N Pannala	9015265656	Peoples	Balangoda	
3	Lucky	B.S Perera	95326263361	Peoples	Negombo	
4	Manie	J.A Amal	56565563666	BOC	Badulla	
5	Brayan	G.H Fernando	56553265632	Sampath	Gampola	
6	Shadow	I.G Mangala	6535165421...	BOC	Kandy	
.	NULL	NULL	NULL	NULL	NULL	NULL

3. Pet rescue table

Object Explorer LAPTOP-...t_Table LAPTOP...onation LAPTOP...doption

The screenshot shows the SSMS interface with the Object Explorer on the left and two tables open on the right.

Object Explorer:

- Connect...
- Login
- Database Diagrams
- Tables
- System Tables
- FileTables
- External Tables
- Graph Tables
- dbo.logintable
- dbo.Rescue_Pet_Tab

LAPTOP-...t_Table (Viewed)

Pet_ID	Pet_...	Name	Age	Gender	Disa...	Location	Characterisitics	User_Name	NIC	Telephone...
1	cat	Shelly	Abou...	Female	No	Gampa...	black and whi...	Pooja	2001...	0771833652
2	Dog	Shadow	Abou...	Female	No	Kurune...	black and whi...	Senura	2003...	0771833652
3	Dog	Shello	Abou...	Female	No	Kalutha...	black and whi...	Pooja	2001...	0718659324
4	cat	Shagi	Verty...	I can't ...	No	Maradh...	White	Senura	2005...	0752334089
5	Dog	lucky	2 yea...	Male	No	Near ga...	gray	Neja	2004...	0775628456
6	cat	Cutie	2 mo...	I can't ...	Yes. I...	Kelani t...	black and gray	Hiruka	1995...	0774615963
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

LAPTOP...doption

Pet_ID	Pet_...	Name	Age	Gender	Disa...	Location	Characterisitics	User_Name	NIC	Telephone...
1	cat	Shelly	Abou...	Female	No	Gampa...	black and whi...	Pooja	2001...	0771833652
2	Dog	Shadow	Abou...	Female	No	Kurune...	black and whi...	Senura	2003...	0771833652
3	Dog	Shello	Abou...	Female	No	Kalutha...	black and whi...	Pooja	2001...	0718659324
4	cat	Shagi	Verty...	I can't ...	No	Maradh...	White	Senura	2005...	0752334089
5	Dog	lucky	2 yea...	Male	No	Near ga...	gray	Neja	2004...	0775628456
6	cat	Cutie	2 mo...	I can't ...	Yes. I...	Kelani t...	black and gray	Hiruka	1995...	0774615963
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL