

**PET ADOPTION SYSTEM****A MINI-PROJECT REPORT**

*Submitted by*

**SAISANJAY S S                    240701458**

**BALAJI S                        240701069**

**LAKSHAN VEERA N            240701282**

*in partial fulfillment of the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**

**An Autonomous Institute**

**CHENNAI**

**NOVEMBER 2025**

## **BONAFIDE CERTIFICATE**

Certified that this project “**PET ADOPTION SYSTEM**” is the bona fide work of “**SAISANJAY S S, BALAJI S, LAKSHAN VEERA N**” who carried out the project work under my supervision.

### **SIGNATURE**

**Mrs. B . DEEPA**

**ASSISTANT PROFESSOR**

Dept. of Computer Science and Engg,  
Rajalakshmi Engineering College  
Chennai

This mini project report is submitted for the viva voce examination to be held on

---

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ABSTRACT

Pet adoption systems play an essential role in connecting homeless pets with caring adopters. The existing manual system of recording and managing pet data often leads to inefficiencies and a lack of proper record tracking.

The **Pet Adoption System** is a database-driven application developed using **Java Swing as the front end** and **MySQL as the back end**. It allows administrators to add, view, and manage pet information and adopter records. Users can view available pets and complete the adoption process efficiently.

The primary objective is to provide a **user-friendly interface** for managing pets and adopters, automating tasks such as adoption record updates, and ensuring transparency and accessibility of pet information.

This system helps animal shelters or pet agencies to maintain their database efficiently and improves the process of matching pets to potential adopters.

## **ACKNOWLEDGEMENT**

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Mrs. B. DEEPA**, for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

**1.SAISANJAY SS**

**2. BALAJI S**

**3.LAKSHAN VEERA N**

**TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO</b>
	<b>ABSTRACT</b>	<b>iv</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>5</b>
1.1	INTRODUCTION	7
1.2	SCOPE OF THE WORK	7
1.3	PROBLEM STATEMENT	7
1.4	AIM AND OBJECTIVES OF THE PROJECT	8
<b>2</b>	<b>SYSTEM SPECIFICATIONS</b>	<b>8</b>
2.1	HARDWARE SPECIFICATIONS	8
2.2	SOFTWARE SPECIFICATIONS	8
<b>3</b>	<b>MODULE DESCRIPTION</b>	<b>9</b>
<b>4</b>	<b>CODING</b>	<b>10</b>
<b>5</b>	<b>SCREENSHOTS</b>	<b>14</b>
<b>6</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>18</b>
<b>7</b>	<b>REFERENCES</b>	<b>19</b>

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>5.1</b>	<b>HOME PAGE</b>	<b>14</b>
<b>5.2</b>	<b>VIEW PETS</b>	<b>14</b>
<b>5.3</b>	<b>ADD PET</b>	<b>15</b>
<b>5.4</b>	<b>ADOPT A PET</b>	<b>15</b>
<b>5.5</b>	<b>VIEW ADOPTERS</b>	<b>15</b>
<b>5.6</b>	<b>EXITING THE PROJECT</b>	<b>16</b>
<b>5.7</b>	<b>DATABASE</b>	<b>17</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 INTRODUCTION

The Pet Adoption System helps organizations and individuals manage the process of adopting pets efficiently. This application connects people looking to adopt with the animals available for adoption.

The system enables easy addition, viewing, and updating of pet and adopter details. It eliminates the need for manual record keeping, making the process transparent and faster.

### 1.2 SCOPE OF THE WORK

The system benefits both pet adoption agencies and users by:

- 1.Allowing admins to manage pet data.
- 2.Helping users browse and adopt pets easily.
- 3.Maintaining records of adopters and adopted pets.
- 4.Reducing manual paperwork through automation.

### 1.3 PROBLEM STATEMENT

Animal shelters often maintain pet records manually, leading to **errors, inefficiency, and data loss**. Tracking pet availability and adoption history becomes difficult. The proposed system aims to automate these tasks, providing accurate and accessible information.

## 1.4 AIM AND OBJECTIVES OF THE PROJECT

- To create a database system for managing pet and adopter records.
- To allow users to view available pets.
- To record adopter details automatically during adoption.
- To provide an easy-to-use interface for users and administrators.

# CHAPTER 2

## SYSTEM SPECIFICATIONS

### 2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
HDD	:	500 GB or Above

### 2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10/11
Front - End	:	Java Swing (JFrame GUI)
Back - End	:	MySql
Database Connectivity	:	JDBC
IDE	:	Eclipse

## CHAPTER 3

### MODULE DESCRIPTION

The Pet Adoption System consists of the following main modules:

#### 1. Admin Module

- Add new pets to the system.
- View all pets and their adoption status.
- Update the adoption status when a pet is adopted.
- View and manage adopter details.

#### 2. User Module

- View available pets based on type (Dog, Cat, etc.).
- Adopt a pet by entering personal details.
- Receive confirmation upon successful adoption.

#### 3. Database Module

- Stores all pet and adopter details in tables.
- Ensures referential integrity between **Pets** and **Adopter** tables.

## CHAPTER 4

### SAMPLE CODING

**Language Used: Java (Swing GUI)**

**Database: MySQL**

**Connection: JDBC**

#### **Main Functions:**

1. **viewPets(type)** – Displays available pets of a particular type.
2. **addPets()** – Inserts new pet data into the database.
3. **adoptPet(id)** – Updates adoption status to “Adopted.”
4. **addAdopter()** – Stores adopter details in the database.
5. **viewAdopter()** – Displays list of all adopters with adopted pets.

This project uses event-driven programming through **ActionListener** in Swing, allowing dynamic response to button clicks (e.g., “View Pets”, “Add Pet”, “Adopt Pet”).

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import java.sql.*;

public class Project extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;

    static final String URL = "jdbc:mysql://localhost:3306/Project";
    static final String USERNAME = "root";
    static final String PASSWORD = "18062006";

    public static void main(String[] args) {
        EventQueue.invokeLater(() -> {
            try {
                Class.forName("com.mysql.cj.jdbc.Driver");
                Project frame = new Project();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        });
    }

    public Project() {
        setTitle("Pet Adoption System 🐾");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setExtendedState(JFrame.MAXIMIZED_BOTH);
        setLocationRelativeTo(null);
    }
}
```

```

public Project() {
    setTitle("Pet Adoption System");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setExtendedState(JFrame.MAXIMIZED_BOTH);
    setLocationRelativeTo(null);

    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(10, 10, 10, 10));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JMenuBar menuBar = new JMenuBar();
    JMenu fileMenu = new JMenu("File");
    JMenuItem exitItem = new JMenuItem("Exit");
    exitItem.addActionListener(e -> {
        int confirm = JOptionPane.showConfirmDialog(this, "Exit Application?", "Confirm", JOptionPane.YES_NO_OPTION);
        if (confirm == JOptionPane.YES_OPTION) System.exit(0);
    });
    fileMenu.add(exitItem);
    menuBar.add(fileMenu);
    setJMenuBar(menuBar);

    JLabel lblTitle = new JLabel("Connecting Hearts and Homes");
    lblTitle.setFont(new Font("Segoe UI", Font.BOLD, 22));
    lblTitle.setHorizontalAlignment(SwingConstants.CENTER);
    lblTitle.setBounds(0, 20, getWidth(), 40);
    contentPane.add(lblTitle);

    JButton btnViewPets = new JButton("View Pets");
    btnViewPets.setBounds(250, 100, 150, 35);
    contentPane.add(btnViewPets);
}

```

```

JButton btnAddPet = new JButton("Add Pet");
btnAddPet.setBounds(250, 150, 150, 35);
contentPane.add(btnAddPet);

JButton btnViewAdopters = new JButton("View Adopters");
btnViewAdopters.setBounds(250, 200, 150, 35);
contentPane.add(btnViewAdopters);

JButton btnAdoptPet = new JButton("Adopt a Pet");
btnAdoptPet.setBounds(250, 250, 150, 35);
contentPane.add(btnAdoptPet);

JButton btnExit = new JButton("Exit");
btnExit.setBounds(250, 300, 150, 35);
contentPane.add(btnExit);

JTextArea outputArea = new JTextArea();
outputArea.setEditable(false);
outputArea.setFont(new Font("Monospaced", Font.PLAIN, 14));
JScrollPane scrollPane = new JScrollPane(outputArea);
scrollPane.setBounds(450, 100, 600, 400);
contentPane.add(scrollPane);

```

```

btnViewAdopters.addActionListener(e -> {
    try {
        String result = viewAdopter();
        outputArea.setText(result);
    } catch (Exception ex) {
        showError(ex);
    }
});

btnAdoptPet.addActionListener(e -> {
    try {
        String idStr = JOptionPane.showInputDialog("Enter Pet ID to adopt:");
        if (idStr == null || idStr.isEmpty()) return;
        int petId = Integer.parseInt(idStr);

        String name = JOptionPane.showInputDialog("Your Name:");
        String email = JOptionPane.showInputDialog("Your Email:");
        String phone = JOptionPane.showInputDialog("Your Phone:");

        if (name == null || email == null || phone == null ||
            name.isEmpty() || email.isEmpty() || phone.isEmpty()) {
            JOptionPane.showMessageDialog(this, "All fields are required!");
            return;
        }

        adoptPet(petId);
        addAdopter(name, email, phone, petId);
        JOptionPane.showMessageDialog(this, "✅ Adoption successful! Welcome to the family.");
    } catch (Exception ex) {
        showError(ex);
    }
});

```

```

btnViewPets.addActionListener(e -> {
    try {
        String type = JOptionPane.showInputDialog("Enter type (Dog, Cat, etc.):");
        if (type != null && !type.isEmpty()) {
            String result = viewPets(type);
            outputArea.setText(result);
        }
    } catch (Exception ex) {
        showError(ex);
    }
});

btnAddPet.addActionListener(e -> {
    try {
        String type = JOptionPane.showInputDialog("Type:");
        String name = JOptionPane.showInputDialog("Name:");
        String breed = JOptionPane.showInputDialog("Breed:");
        String age = JOptionPane.showInputDialog("Age:");
        String gender = JOptionPane.showInputDialog("Gender:");
        String price = JOptionPane.showInputDialog("Price:");

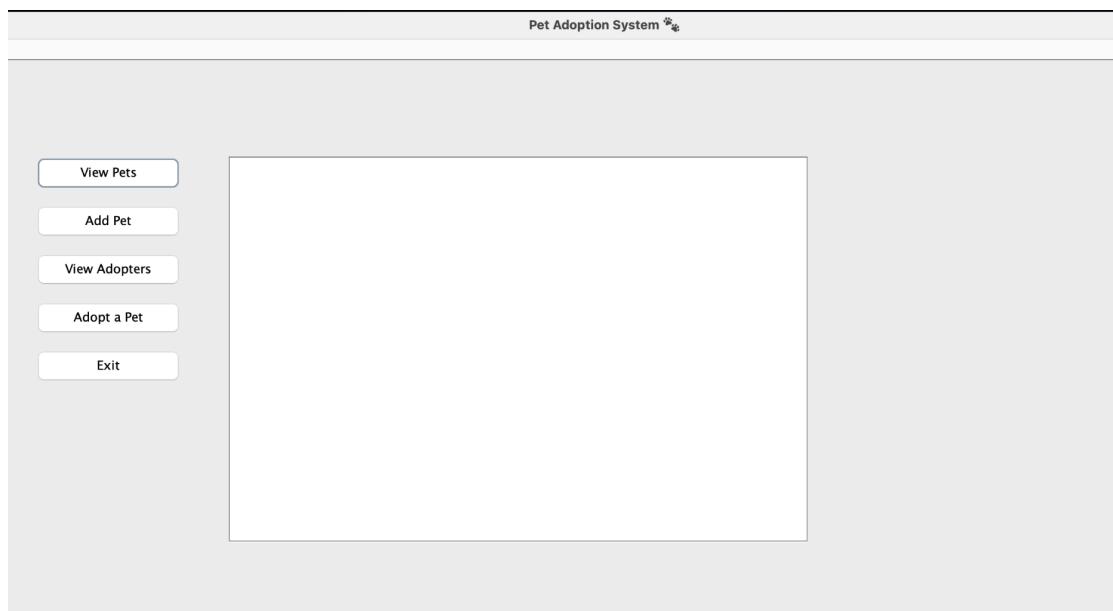
        if (type == null || name == null || breed == null ||
            age == null || gender == null || price == null ||
            type.isEmpty() || name.isEmpty() || breed.isEmpty() ||
            age.isEmpty() || gender.isEmpty() || price.isEmpty()) {
            JOptionPane.showMessageDialog(this, "All fields are required!");
            return;
        }

        addPet(type, name, breed, age, gender, price);
        JOptionPane.showMessageDialog(this, "✅ Pet added successfully!");
    } catch (Exception ex) {
        showError(ex);
    }
});

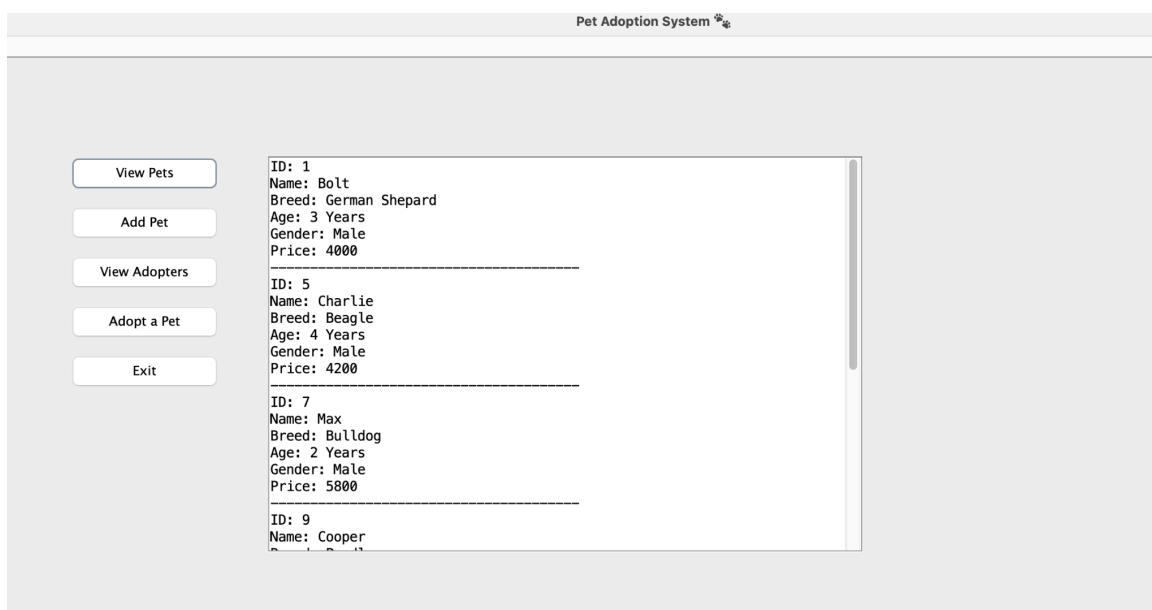
```

## CHAPTER 5

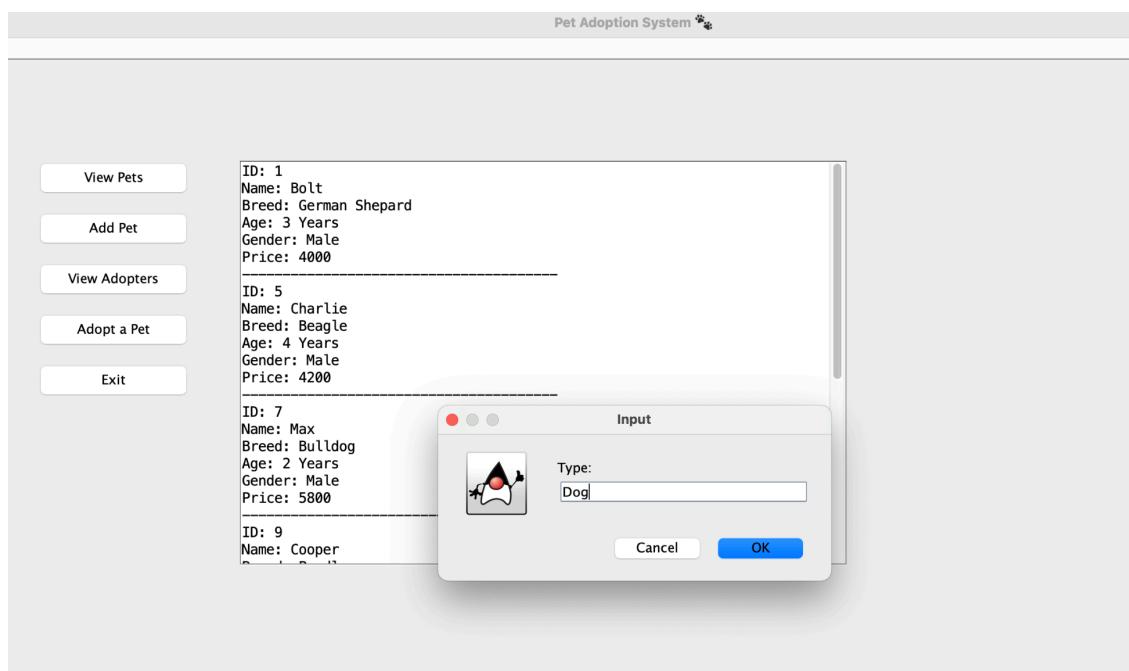
### SCREEN SHOTS



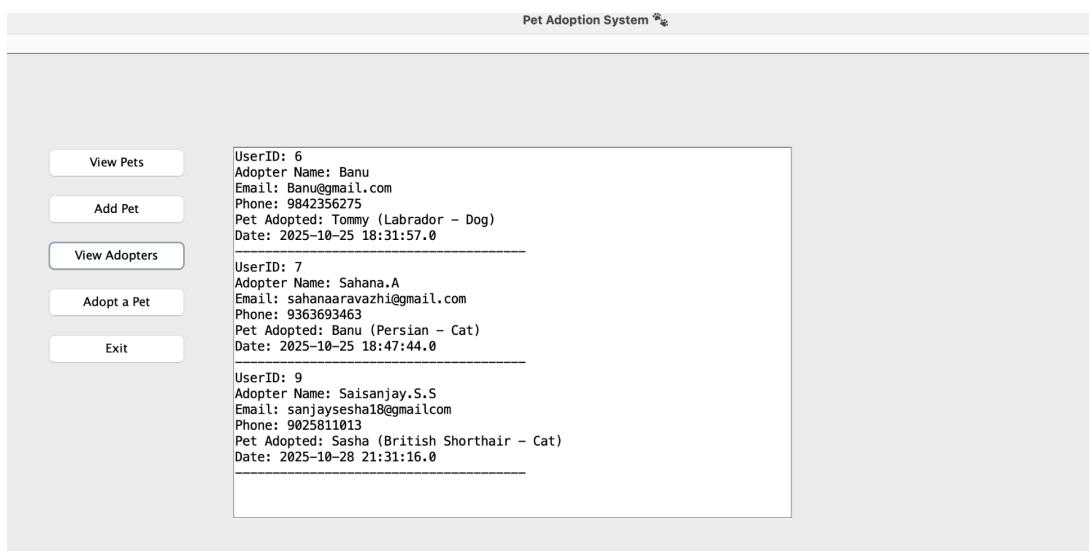
**Fig 5.1 Home page**



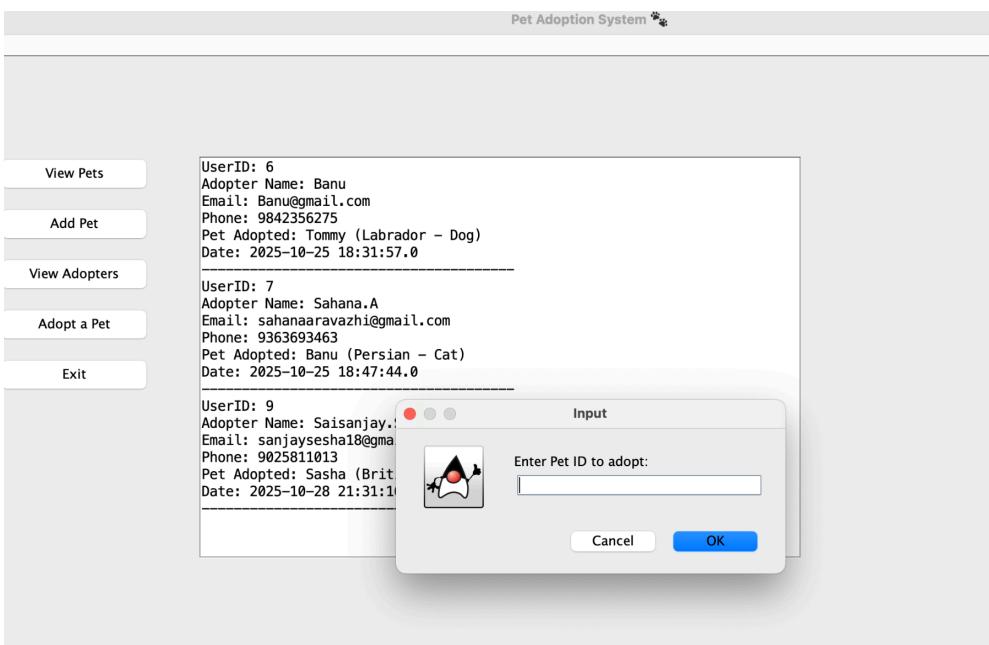
**Fig 5.2 View Pets**



**Fig 5.3 Add Pets**



**Fig 5.4 View Adopters**



**Fig 5.5 Adopt a Pet**



**Fig 5.6 Exiting the project**

PetID	Name	Type	Breed	Age	Gender	Adoption_Status	Price
1	Bolt	Dog	German Shepard	3 Years	Male	Available	4000
2	Tommy	Dog	Labrador	1 Years	Male	Adopted	3750
4	Meow	Cat	Persian	5 months	Female	Available	3550
5	Charlie	Dog	Beagle	4 Years	Male	Available	4200
6	Mia	Cat	Persian	1.5 Years	Female	Available	3800
7	Max	Dog	Bulldog	2 Years	Male	Available	5800
8	Nala	Cat	Ragdoll	3 Months	Female	Available	4900
9	Cooper	Dog	Poodle	5 Years	Male	Available	3500
10	Nemo	Cat	Bengal	7 Years	Male	Available	3200
11	Lucy	Dog	Shih Tzu	9 Months	Female	Adopted	4700
12	Oreo	Cat	Tuxedo (Mix)	6 Years	Male	Available	1800
13	Bailey	Dog	Pomeranian	1 Year	Female	Available	5100
14	Sasha	Cat	British Shorthair	2 Months	Female	Adopted	4300
15	Banu	Cat	Persian	2 Years	Female	Adopted	6000
16	Rocky	Dog	Dobermann	4	Male	Available	5500

UserID	Name	Email	Phone	AdoptionDate	PetID
6	Banu	Banu@gmail.com	9842356275	2025-10-25 18:31:57	2
7	Sahana.A	sahanaaravazhi@gmail.com	9363693463	2025-10-25 18:47:44	15
9	Saisanjay.S.S	saniaysesha18@gmail.com	9025811013	2025-10-28 21:31:16	14

**Fig 5.7 MySQL database structure with Pets and Adopter tables.**

## CHAPTER 6

### CONCLUSION AND FUTURE ENHANCEMENT

The Pet Adoption System efficiently manages the details of pets and adopters. It simplifies the adoption process, reduces paperwork, and provides a transparent, automated way for managing pet adoption records.

- 1.Add user login and authentication system.
- 2.Integrate online payment for adoption fees.
- 3.Include photo upload feature for pets.
- 4.Develop a web or mobile-based version for wider access.

## REFERENCES

1. <https://docs.oracle.com/javase/8/docs/api/javax/swing/package-summary.html>
2. <https://docs.oracle.com/javase/tutorial/jdbc/index.html>
3. <https://dev.mysql.com/doc/refman/8.0/en/>
4. <https://dev.mysql.com/doc/connector-j/8.0/en/>
5. <https://www.oracle.com/java/technologies/data-access-object.html>