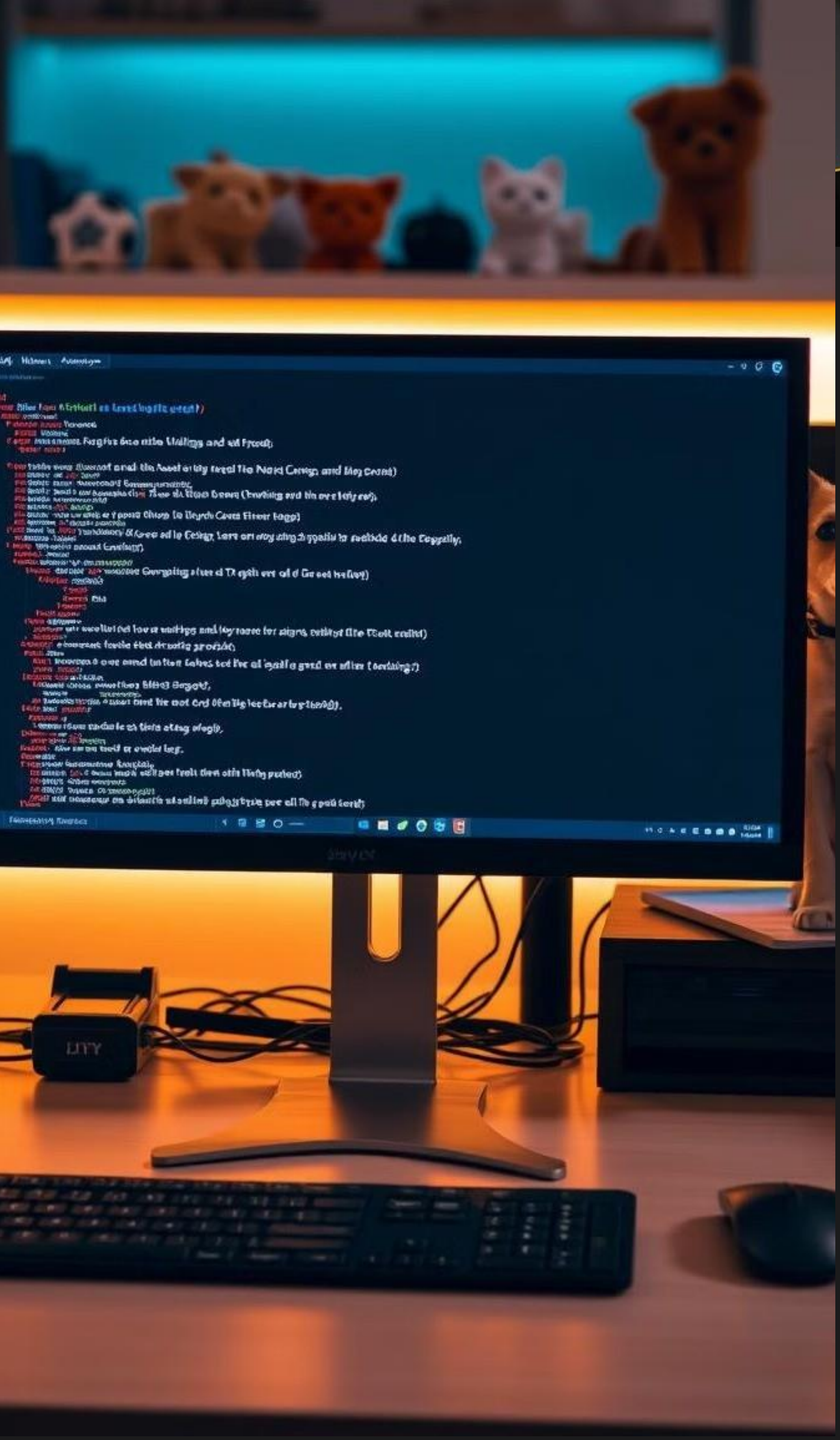


Pet Adoption Platform: A Java-based Application for Pet Management

Welcome to our innovative Pet Adoption Platform, a robust Java-based application designed to streamline pet management and facilitate adoptions. This project, developed by Ankit Kumar, combines the power of Java programming with MySQL database management to create a user-friendly console interface for managing pet information efficiently.

Our platform aims to bridge the gap between pets in need of homes and potential adopters, making the process smoother and more accessible. By leveraging technology, we're not just creating an application; we're fostering connections between humans and their future furry companions.

SUMIT KUMAR - 23SCSE1012617
KAVYA KAIRA - 23SCSE1011695
ANKIT KUMAR - 23SCSE1012390
VISHNU SHANKAR PANDEY – 23SCSE1011324

Introduction to the Pet Adoption Platform

Overview

The Pet Adoption Platform is a comprehensive Java application designed to revolutionize the way we manage pet adoptions. It provides a centralized system for storing, retrieving, and managing pet information, making it easier for shelters and adoption agencies to keep track of their animal residents.

Purpose

Our primary goal is to facilitate the adoption process by providing a user-friendly interface for managing pet information. By streamlining data management, we aim to increase the efficiency of pet adoption agencies and improve the chances of pets finding their forever homes.

Impact

By digitalizing the pet adoption process, we're not only saving time and resources but also potentially saving more animal lives. This platform serves as a bridge between pets in need and loving homes, making the adoption process more accessible and efficient for all parties involved.

Key Features of the Pet Adoption Platform

1

Add New Pets

Our platform allows users to easily add new pets to the database. This feature captures essential information such as the pet's name, breed, age, and availability status. By providing a structured input process, we ensure that all necessary details are recorded accurately, creating a comprehensive profile for each animal.

2

View Available Pets

Users can effortlessly browse through all available pets for adoption. This feature provides a quick overview of the current animal residents, helping potential adopters and shelter staff to easily access and review pet information. The view function displays details in a clear, organized manner, facilitating informed decision-making.

3

User-Friendly Console Interface

We've designed an intuitive console interface that makes navigation and operation simple, even for those with limited technical experience. The menu-driven system guides users through various options, ensuring that all functions are easily accessible and user-friendly.

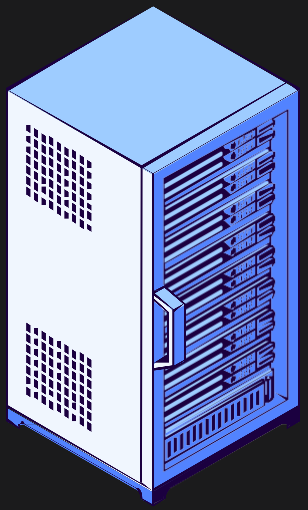
Technology Stack

Java

As our primary programming language, Java offers robust, object-oriented design capabilities. Its platform independence ensures our application can run on various systems without modification.

MySQL

We utilize MySQL as our relational database management system. It provides a secure and efficient way to store and retrieve pet data, supporting complex queries and data integrity.



JDBC

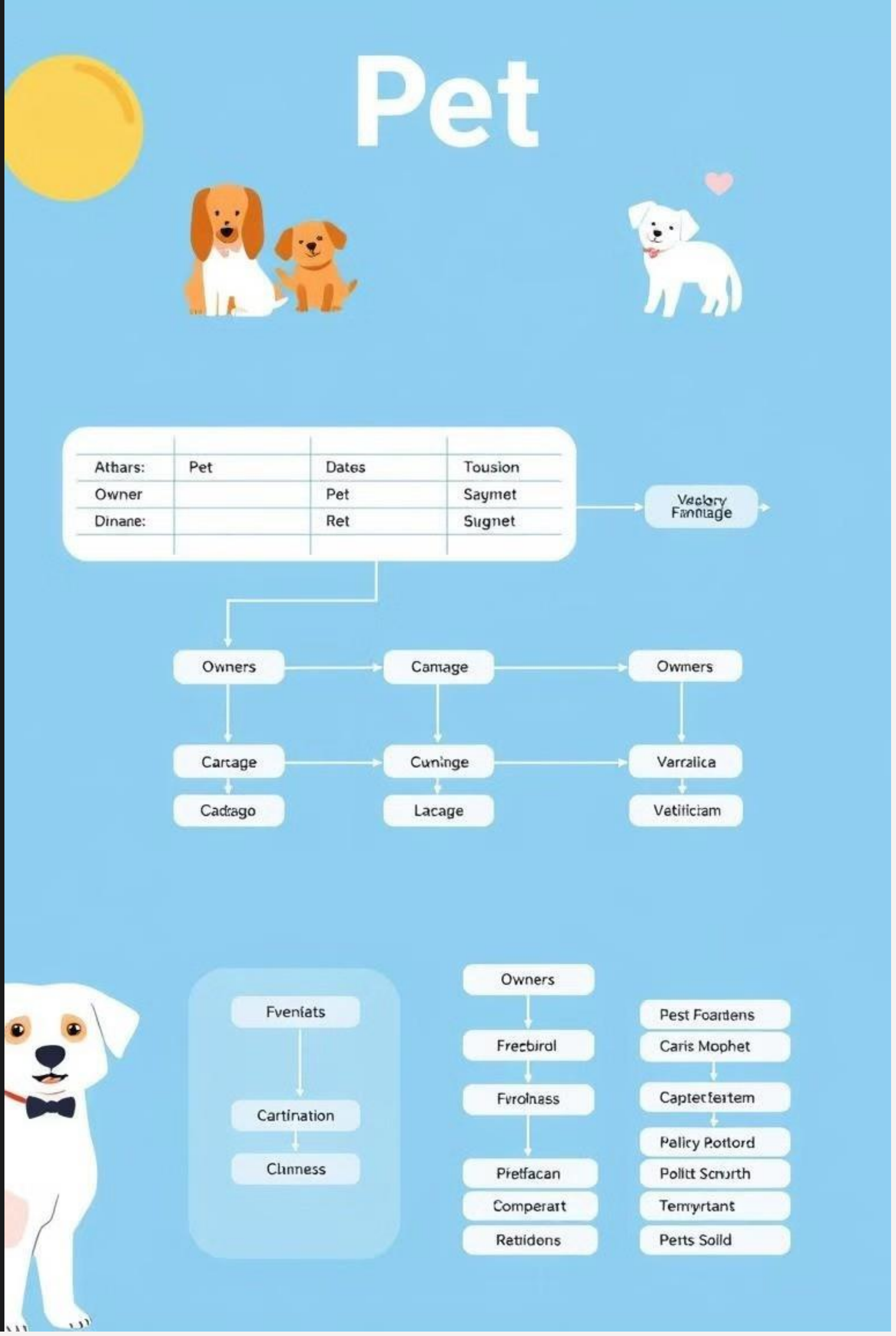
Java Database Connectivity (JDBC) serves as our bridge between Java and MySQL, enabling seamless database operations and ensuring efficient data management.

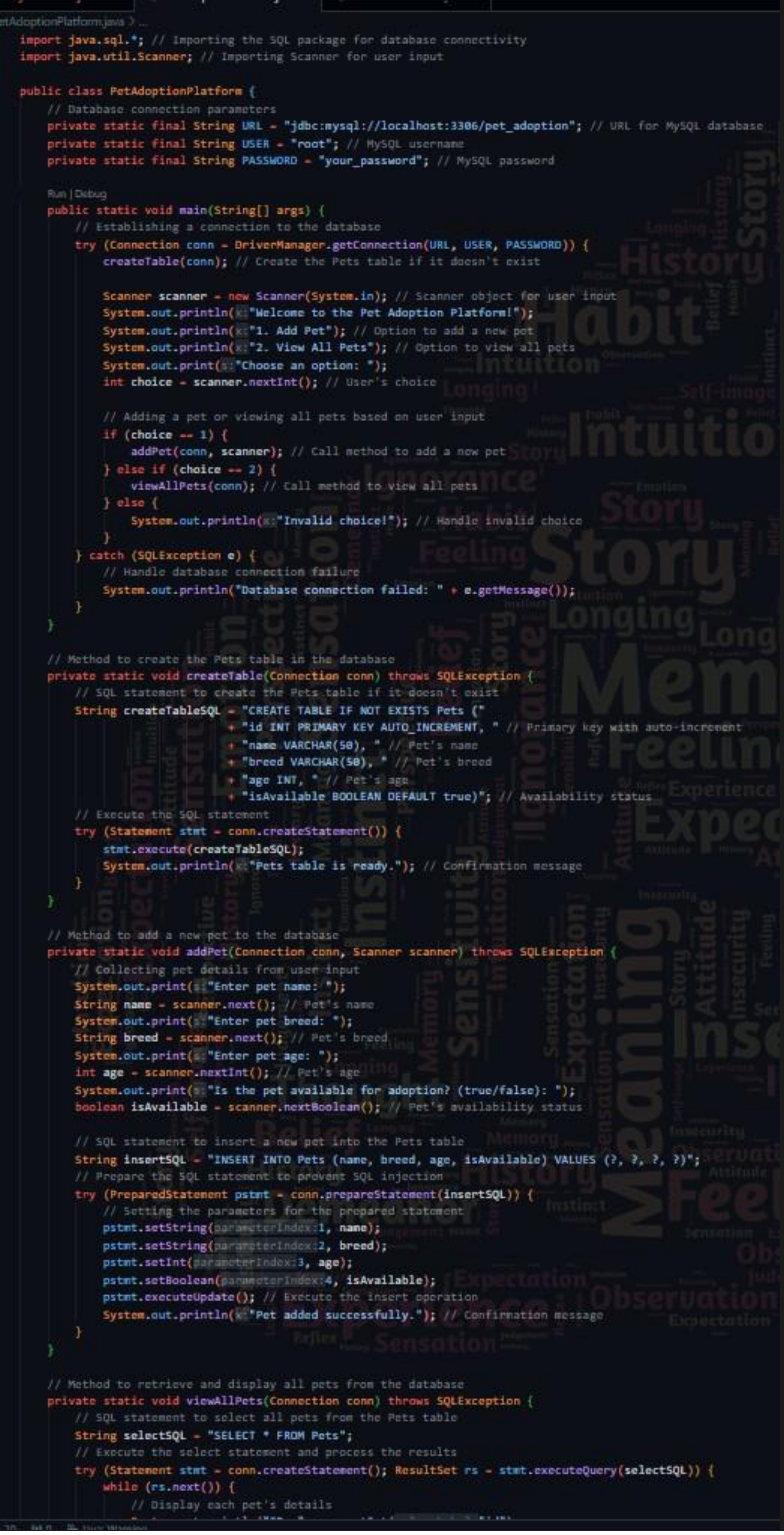
Scanner

Java's Scanner class is employed for handling user inputs, providing a straightforward way to capture and process user interactions within the console interface.

Database Design

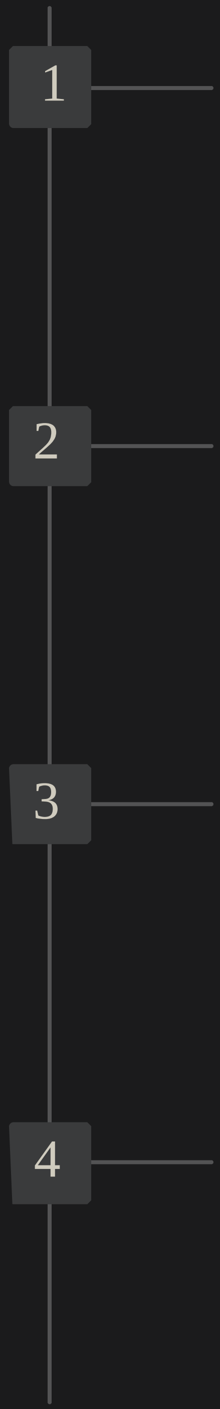
Column Name	Data Type	Constraints
id	INT	Primary Key, Auto Increment Not Null Not Null
name	VARCHAR(50)	Not Null Default:
breed	VARCHAR(50)	true
age	INT	
isAvailable	BOOLEAN	





```java
import java.sql.*; // Importing the SQL package for database connectivity
import java.util.Scanner; // Importing Scanner for user input

public class PetAdoptionPlatform {
    // Database connection parameters
    private static final String URL = "jdbc:mysql://localhost:3306/pet_adoption"; // URL for MySQL database
    private static final String USER = "root"; // MySQL username
    private static final String PASSWORD = "your_password"; // MySQL password

    Run | Debug
    public static void main(String[] args) {
        // Establishing a connection to the database
        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
            createTable(conn); // Create the Pets table if it doesn't exist

            Scanner scanner = new Scanner(System.in); // Scanner object for user input
            System.out.println(x:"Welcome to the Pet Adoption Platform!");
            System.out.println(x:"1. Add Pet"); // Option to add a new pet
            System.out.println(x:"2. View All Pets"); // Option to view all pets
            System.out.print(s:"Choose an option: ");
            int choice = scanner.nextInt(); // User's choice

            // Adding a pet or viewing all pets based on user input
            if (choice == 1) {
                addPet(conn, scanner); // Call method to add a new pet
            } else if (choice == 2) {
                viewAllPets(conn); // Call method to view all pets
            } else {
                System.out.println(x:"Invalid choice!"); // Handle invalid choice
            }
        } catch (SQLException e) {
            // Handle database connection failure
            System.out.println("Database connection failed: " + e.getMessage());
        }
    }

    // Method to create the Pets table in the database
    private static void createTable(Connection conn) throws SQLException {
        // SQL statement to create the Pets table if it doesn't exist
        String createTableSQL = "CREATE TABLE IF NOT EXISTS Pets ("
                              + "id INT PRIMARY KEY AUTO_INCREMENT, " // Primary key with auto-increment
                              + "name VARCHAR(50), " // Pet's name
                              + "breed VARCHAR(50), " // Pet's breed
                              + "age INT, " // Pet's age
                              + "isAvailable BOOLEAN DEFAULT true)"; // Availability status
        // Execute the SQL statement
        try (Statement stmt = conn.createStatement()) {
            stmt.execute(createTableSQL);
            System.out.println(x:"Pets table is ready."); // Confirmation message
        }
    }

    // Method to add a new pet to the database
    private static void addPet(Connection conn, Scanner scanner) throws SQLException {
        // Collecting pet details from user input
        System.out.print(s:"Enter pet name: ");
        String name = scanner.next(); // Pet's name
        System.out.print(s:"Enter pet breed: ");
        String breed = scanner.next(); // Pet's breed
        System.out.print(s:"Enter pet age: ");
        int age = scanner.nextInt(); // Pet's age
        System.out.print(s:"Is the pet available for adoption? (true/false): ");
        boolean isAvailable = scanner.nextBoolean(); // Pet's availability status

        // SQL statement to insert a new pet into the Pets table
        String insertSQL = "INSERT INTO Pets (name, breed, age, isAvailable) VALUES (?, ?, ?, ?)";
        // Prepare the SQL statement to prevent SQL injection
        try (PreparedStatement pstmt = conn.prepareStatement(insertSQL)) {
            // Setting the parameters for the prepared statement
            pstmt.setString(parameterIndex:1, name);
            pstmt.setString(parameterIndex:2, breed);
            pstmt.setInt(parameterIndex:3, age);
            pstmt.setBoolean(parameterIndex:4, isAvailable);
            pstmt.executeUpdate(); // Execute the insert operation
            System.out.println(x:"Pet added successfully."); // Confirmation message
        }
    }

    // Method to retrieve and display all pets from the database
    private static void viewAllPets(Connection conn) throws SQLException {
        // SQL statement to select all pets from the Pets table
        String selectSQL = "SELECT * FROM Pets";
        // Execute the select statement and process the results
        try (Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(selectSQL)) {
            while (rs.next()) {
                // Display each pet's details
```

Code Overview: Main Class



Initialization

The PetAdoptionPlatform class serves as the entry point of our application. It initializes the database connection and sets up the user interface.

Main Loop

A while loop presents options to the user and processes their input, calling appropriate methods based on the selected option.

Error Handling

Robust error handling is implemented to manage potential exceptions, ensuring a smooth user experience even in case of unexpected inputs or database issues.

Resource Management

The class properly manages resources, closing database connections and scanners when they're no longer needed to prevent resource leaks.

Key Functions: Database Operations

1

Create Table

The createTable method ensures the database structure is ready for operations. It uses a SQL statement to create the Pets table if it doesn't exist, setting up the necessary columns with appropriate data types and constraints.

2

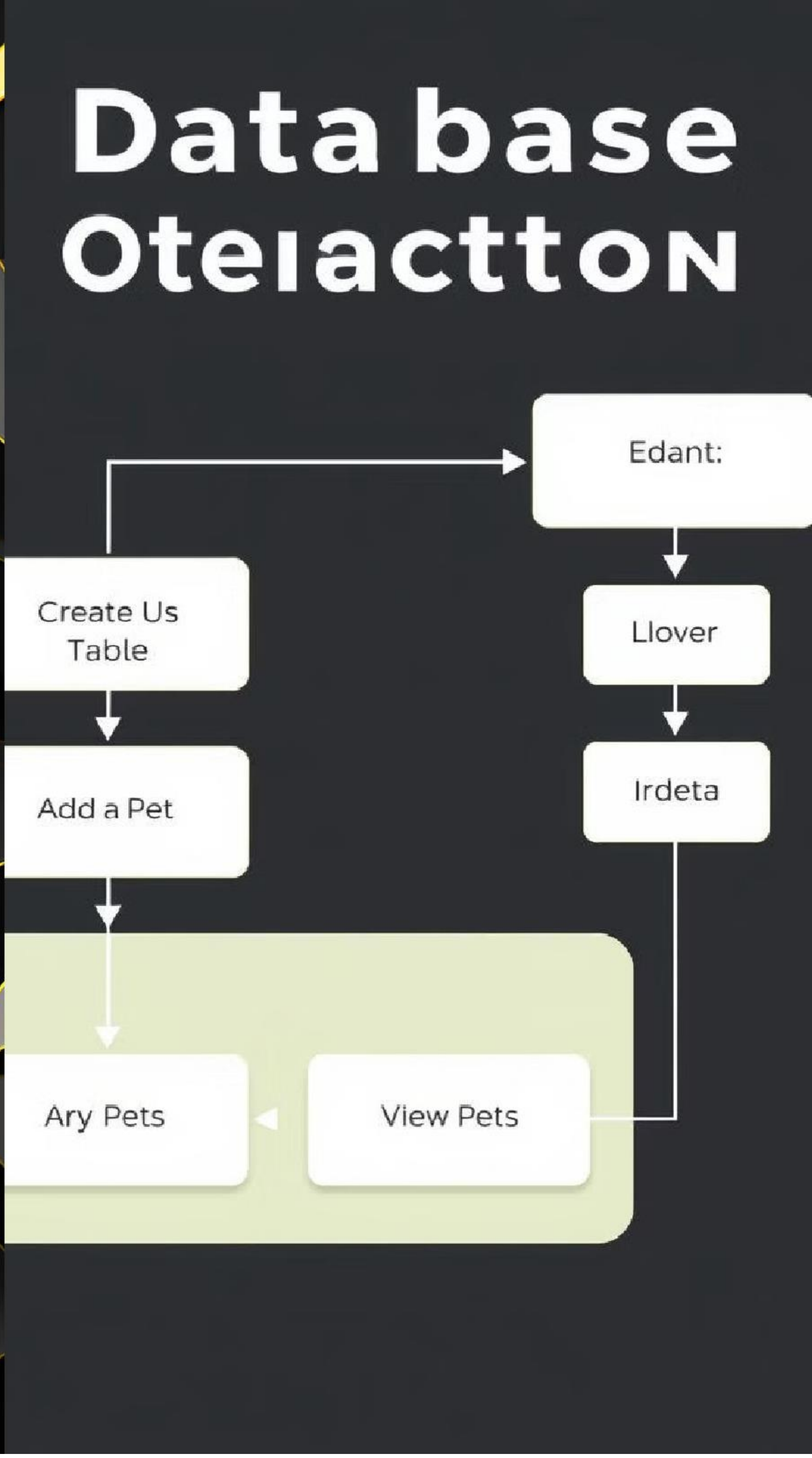
Add Pet

The addPet function collects pet details from user input and inserts new pet records into the Pets table. It utilizes PreparedStatement to prevent SQL injection, ensuring data security and integrity.

3

View Pets

The viewAllPets method retrieves and displays all pet records. It executes a SELECT query and uses ResultSet to iterate through the database results, presenting the information in a user-friendly format.



Sample Output and User Interaction

Main Menu

The application greets users with a welcome message and presents a menu of options: 1. Add Pet, 2. View All Pets. This intuitive interface allows users to easily navigate the system's functionality.

Adding a Pet

When adding a pet, the system prompts for specific details: name, breed, age, and availability. It then confirms successful addition, providing immediate feedback to the user.

Viewing Pets

The view function displays a list of all pets in the database, including their ID, name, breed, age, and availability status. This comprehensive overview helps users quickly assess the current pet inventory.

Error Handling

The system includes robust error handling, providing clear messages for invalid inputs or database errors, ensuring a smooth user experience even when unexpected issues arise.

Conclusion and Future Work

Current Achievement

Our Pet Adoption Platform represents a significant step forward in digitalizing pet management. It offers a robust, efficient system for tracking and managing adoptable pets, streamlining the work of shelters and adoption agencies.

Future Enhancements

Looking ahead, we envision expanding the platform with a graphical user interface (GUI) for improved usability. We also plan to implement features for updating and deleting pet records, and potentially introduce user accounts for adopters to interact directly with the system.

Broader Impact

By continually improving and expanding this platform, we aim to make a lasting impact on the pet adoption process, ultimately helping more animals find loving homes and supporting the vital work of animal welfare organizations worldwide.

THANK YOU !

TEAM EDGE