PET ADOPTION SYSTEM

A MINI-PROJECT BY :

NISHANTH P-230701213

NITHISH.N - 230701219

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 2024

BONAFIDE CERTIFICATE

Certified that this project **"PET ADOPTION SYSTEM"** is the bonafide work of **"NISHANTH POSA , NITHISH.N"** who carried out the project work under my supervision.

Submitted for the practical examination held on _____

SIGNATURE

Ms. ASWANA LAL
Assistant Professor,
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

This project presents a **Pet Adoption System** that streamlines the process of adopting pets by integrating Java for the application's backend and user interface with MySQL as the database for managing data. The system provides a platform for pet shelters and potential adopters to interact efficiently.

The system enables shelters to manage their pet inventory, including details such as breed, age, health records, and adoption status. Adopters can browse through available pets, filter based on preferences, and apply for adoption. The platform incorporates features such as user authentication, real-time updates on pet availability, and a seamless application process.

The backend is implemented in Java, ensuring robust and scalable operations, while MySQL provides a reliable relational database to store and retrieve structured data efficiently. By leveraging Java's object-oriented capabilities and MySQL's query optimization, the system ensures secure, fast, and user-friendly interactions.

This solution aims to bridge the gap between shelters and adopters, promoting transparency and ease in pet adoption processes while reducing administrative burdens for shelter staff. It also incorporates features to track adoption histories and provide analytics for shelters to improve their operations.

TABLE OF CONTENTS

INTRODUCTION

1.1 INTRODUCTION

The project helps students by giving them an idea of potential research topics, where they will choose the interested topic and will be able to submit the finished dissertation on the website, where tutors will evaluate the project and results will be posted on the website.

1.1 IMPLEMENTATION

The **PET ADOPTION SYSTEM** project discussedhere is implemented using the concepts of **JAVA SWINGS** and **MYSQL**.

1.2 SCOPE OF THE PROJECT

The website is designed in a way where students will have to register on the website by creating an account for themselves in order for the students to get all their data in one place, where everything is organized for them.Thus saving them time and giving a sense of professionalism.

1.3 WEBSITE FEATURES

- Registering and login page.
- Custom profile for each student.
- Dashboard showing possible actions.
- Guide to choosing research topics.
- Option to submit project on website.

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS:

PROCESSOR : Intel i5

MEMORY SIZE : 4GB(Minimum)

HARD DISK : 500 GB of free space

2.2 SOFTWARE SPECIFICATIONS:

PROGRAMMING LANGUAGE : Java, MySQL

FRONT-END : Java

BACK-END : MySQL

OPERATING SYSTEM : Windows 10 **SAMPLE CODE**

SAMPLE CODE

3.1 HOME PAGE DESIGN

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class PetAdoptionUI {

    private static final String URL = "jdbc:mysql://localhost:3306/pet_adoption_db"; // Database
URL
    private static final String USER = "root"; // Your MySQL username
    private static final String PASSWORD = "1234"; // Your MySQL password

    private static Connection connection;

    // Swing components
    private JFrame frame;
    private JPanel panel;
    private JTextField petNameField, petBreedField, petAgeField, petTypeField,
adopterNameField;
    private JButton addPetButton, adoptPetButton;
    private JTable petTable;
    private DefaultTableModel tableModel;

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            try {
                PetAdoptionUI window = new PetAdoptionUI();
                window.frame.setVisible(true);
```

```
        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}
```

3.2 UI DESIGN

```
public PetAdoptionUI() {
    // Set up the database connection
    try {
        connection = DriverManager.getConnection(URL, USER, PASSWORD);
        System.out.println("Connected to the database.");
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(frame, "Error connecting to database: " +
e.getMessage());
    }

    // Set up the frame
    frame = new JFrame("Pet Adoption System");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(800, 600);

    panel = new JPanel();
    panel.setLayout(new BorderLayout()); // Main panel with BorderLayout

    JPanel inputPanel = new JPanel();
    inputPanel.setLayout(new GridLayout(7, 2, 10, 10)); // 2 columns with space between
rows and columns

    // Add components for adding a pet
    inputPanel.add(new JLabel("Pet Name:"));
    petNameField = new JTextField();
    inputPanel.add(petNameField);
```

```
inputPanel.add(new JLabel("Pet Breed:"));
petBreedField = new JTextField();
inputPanel.add(petBreedField);

inputPanel.add(new JLabel("Pet Age:"));
petAgeField = new JTextField();
inputPanel.add(petAgeField);

inputPanel.add(new JLabel("Pet Type:"));
petTypeField = new JTextField();
inputPanel.add(petTypeField);

addPetButton = new JButton("Add Pet");
addPetButton.addActionListener(e -> addNewPet());
inputPanel.add(addPetButton);

// Add components for adopting a pet
inputPanel.add(new JLabel("Adopter Name:"));
adopterNameField = new JTextField();
inputPanel.add(adopterNameField);

adoptPetButton = new JButton("Adopt Pet");
adoptPetButton.addActionListener(e -> adoptPet());
inputPanel.add(adoptPetButton);

panel.add(inputPanel, BorderLayout.NORTH);

// Set up the table to display the pets
String[] columnNames = { "Pet ID", "Pet Name", "Breed", "Age", "Type", "Status" };
tableModel = new DefaultTableModel(columnNames, 0);
petTable = new JTable(tableModel);
JScrollPane scrollPane = new JScrollPane(petTable);
panel.add(scrollPane, BorderLayout.CENTER);

frame.add(panel);
```

```
        // Load available pets into the table
        loadAvailablePets();
    }
```

3.3 ADOPTING A PET

```
/ Method to adopt a pet
    private void adoptPet() {
        int selectedRow = petTable.getSelectedRow();
        if (selectedRow != -1) {
            int petId = (int) petTable.getValueAt(selectedRow, 0);
            String petName = (String) petTable.getValueAt(selectedRow, 1);

            String adopterName = adopterNameField.getText();

            // Check if the pet is available
            String checkPetQuery = "SELECT * FROM pets WHERE pet_id = ? AND status =
'Available'";
            try (PreparedStatement checkStmt = connection.prepareStatement(checkPetQuery)) {
                checkStmt.setInt(1, petId);
                ResultSet rs = checkStmt.executeQuery();
                if (rs.next()) {
                    // Pet is available, proceed with adoption
                    String sql = "INSERT INTO adoptions (pet_id, adopter_name, adoption_date)
VALUES (?, ?, CURDATE())";
                    try (PreparedStatement pstmt = connection.prepareStatement(sql)) {
                        pstmt.setInt(1, petId);
                        pstmt.setString(2, adopterName);
                        pstmt.executeUpdate();

                        // Update pet status to adopted
                        String updateStatusSql = "UPDATE pets SET status = 'Adopted' WHERE
pet_id = ?";
                        try (PreparedStatement updateStmt =
connection.prepareStatement(updateStatusSql)) {
                            updateStmt.setInt(1, petId);
                            updateStmt.executeUpdate();
```

```
                }
                JOptionPane.showMessageDialog(frame, "Congrats! You have adopted " +
petName);
                loadAvailablePets(); // Reload the pet table after adoption
            }
        } else {
            JOptionPane.showMessageDialog(frame, "Sorry, this pet is not available for
adoption.");
        }
      } catch (SQLException e) {
        JOptionPane.showMessageDialog(frame, "Error adopting pet: " + e.getMessage());
        e.printStackTrace();  // Print the stack trace for debugging
      }
    } else {
      JOptionPane.showMessageDialog(frame, "Please select a pet to adopt.");
    }
  }
}
```

3.4 LOADING A PET

```
// Method to load available pets into the table
  private void loadAvailablePets() {
    // Clear existing rows
    tableModel.setRowCount(0);

    String sql = "SELECT pet_id, name, breed, age, type, status FROM pets WHERE status
= 'Available'";
    try (Statement stmt = connection.createStatement();
         ResultSet rs = stmt.executeQuery(sql)) {
        while (rs.next()) {
            int petId = rs.getInt("pet_id");
            String petName = rs.getString("name");
            String petBreed = rs.getString("breed");
            int petAge = rs.getInt("age");
            String petType = rs.getString("type");
```

```
            String petStatus = rs.getString("status");
            // Add pet data to the table
            tableModel.addRow(new Object[]{ petId, petName, petBreed, petAge, petType,
petStatus });
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(frame, "Error loading pets: " + e.getMessage());
        e.printStackTrace();  // Print the stack trace for debugging
    }
}
```

3.5 STYLING THE APP

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    background-color: #f4f4f4;
    color: #333;
}

header {
    background-color: #ff6b6b;
    color: white;
    padding: 20px 0;
    text-align: center;
}

header h1 {
    margin: 0;
}

header nav ul {
    list-style: none;
```

```
    padding: 0;
}

header nav ul li {
    display: inline-block;
    margin: 0 10px;
}

header nav ul li a {
    color: white;
    text-decoration: none;
    font-weight: bold;
}

#hero {
    background: url('hero-bg.jpg') no-repeat center center/cover;
    color: white;
    text-align: center;
    padding: 100px 20px;
}

#hero h2 {
    font-size: 2.5rem;
}

#hero .button {
    background-color: #ff6b6b;
    color: white;
    padding: 10px 20px;
    text-decoration: none;
    border-radius: 5px;
    font-size: 1.2rem;
}

#hero .button:hover {
    background-color: #ff4c4c;
```

```
}

#adopt {
    padding: 40px 20px;
    text-align: center;
    background-color: white;
}

#adopt h2 {
    font-size: 2rem;
    margin-bottom: 20px;
}

.pet-list {
    display: flex;
    justify-content: center;
    gap: 20px;
}

.pet {
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    padding: 20px;
    text-align: center;
    width: 250px;
}

.pet img {
    width: 100%;
    border-radius: 8px;
}

.pet h3 {
    margin: 10px 0;
}
```

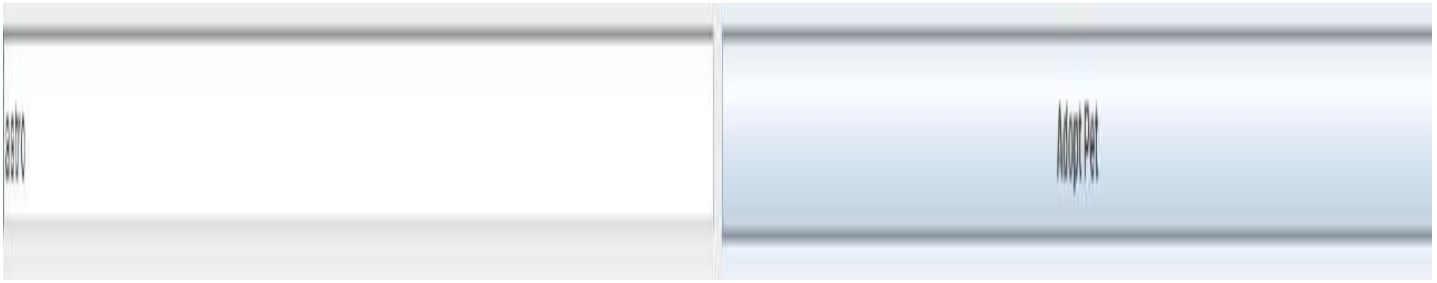
```
.pet p {
    margin: 5px 0;
}

.pet button {
    background-color: #ff6b6b;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

.pet button:hover {
    background-color: #ff4c4c;
}

#about, #contact {
    padding: 40px 20px;
    background-color: #fff;
    text-align: center;
}

#contact form {
    display: flex;
    flex-direction: column;
    align-items: center;
}

#contact label, #contact input, #contact textarea {
    width: 100%;
    max-width: 400px;
    margin: 10px 0;
}
```

```
#contact input[type="submit"] {
    background-color: #ff6b6b;
    color: white;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}

#contact input[type="submit"]:hover {
    background-color: #ff4c4c;
}

footer {
    background-color: #333;
    color: white;
    text-align: center;
    padding: 10px;
}

footer p {
    margin: 0;
}
```

SNAPSHOTS

LOGIN PAGE

OPERATING PAGE



CONCLUSION

The **Pet Adoption System** developed using Java and MySQL offers a comprehensive and user-friendly solution for connecting pet shelters with potential adopters. By digitizing and automating critical processes, the system reduces the administrative workload for shelters, enhances user experience, and improves the visibility of pets in need of homes.

With features such as pet filtering, real-time updates, and user authentication, the platform ensures a smooth and secure adoption journey. Additionally, the system's scalability and data analytics capabilities empower shelters to optimize their operations and track adoption success rates over time.

In conclusion, this project not only fosters efficiency and convenience but also contributes to the broader goal of increasing pet adoptions, ensuring that more animals find loving homes while simplifying the process for all stakeholders involved.

REFERENCES

1. **https://www.javatpoint.com/java-tutorial**
2. https://www.wikipedia.org/
3. https://www.w3schools.com/sql/
4. SQL | Codecademy