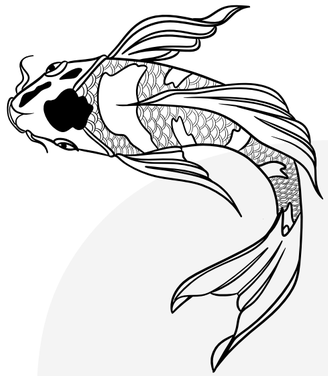
Koi Care System at Home

SWP391

Group 3

FINAL REPORT

1/11/2024



Trần Xuân Trường

Leader/FullStack Developer

Nguyễn Hoàng Phụng

Back-end Developer

Our Team Members

Table of Content

Overview

The Koi Care System at Home is a smart, automated system designed to help koi fish owners monitor and maintain optimal conditions for their fish ponds. Koi fish require a carefully managed environment to thrive, including consistent water quality, temperature, and feeding schedules. The project aims to integrate technology into koi pond care, allowing users to monitor and control various environmental factors from the comfort of their homes.

Problem and Context (Why)

Problem

Time and Expertise Requirement

Managing a koi pond requires time and expertise, which can be challenging for new or busy koi owners

Environmental Stability

Koi fish need a stable environment, with optimal water quality parameters like pH, temperature, and oxygen levels, to thrive.

Health Risks

Without proper management, koi fish are at risk of stress, disease, and even mortality due to inconsistent care.

Context

Digital Solution

The "Koi Care System at Home" project offers a digital platform that integrates automated monitoring, water quality management, and feeding schedules to address these challenges.

Use of Technology

The system utilizes technology such as sensors, notifications, and analytical tools to make pond management more accessible and convenient.

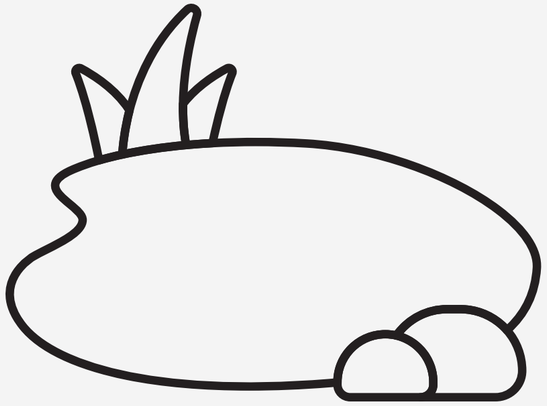
User Roles and Functionality

With user roles like Admin, Shop, and Member, users can manage pond conditions, track koi health, order supplies, and access updates through blogs and reports.

Benefit to Koi Health

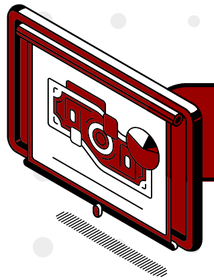
This system not only simplifies pond care but also enhances koi health and longevity through proactive management practices.

Feature (What)



Feature

User Roles



Pond Management

Water Parameter Management

Koi Fish Mangement

Feeding Calculations

Feature

Product Ordering

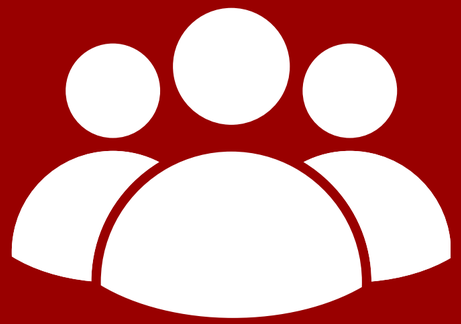
News and Blog Section

Dashboard and Reporting

Salt Calculation

Actors (Who)

Guest



- Description: Users who have not yet registered or logged in. They can view limited information about the system, including general information about koi care, product listings, and blog posts.
- Functions:
- Browse the homepage and shop.
- Read blog posts and news articles.

- Description: Registered users with an account who have access to additional features for managing their koi ponds and fish.
- Functions:
- Manage personal profiles and account settings.
- Access pond and koi fish management features.
- Monitor water parameters and receive recommendations.
- Calculate feeding and salt requirements.
- Order products for pond and fish care.
- View and manage their feedback and comments.

Admin

- Description: Users with the highest level of access to manage the overall system, user accounts, and critical functionalities.
- Functions:
- Perform CRUD operations on user accounts (create, read, update, delete).
- Manage roles and permissions of different users.
- Oversee the overall operation of the system and ensure data integrity.
- Generate reports for system analytics and performance monitoring.

Shop

- Description: Users responsible for managing the shop's inventory, blog content, and user interactions.
- Functions:
- Create and manage reports on sales and user activity.
- Add, edit, and delete products in the inventory.
- Create and manage blog posts and slider content.
- Manage user feedback and respond to inquiries.
- Oversee order management, including viewing, updating status, and canceling orders.

Technology

spring BOOT

React JS

PROS	CONS
Rapid Development: Spring Boot simplifies the setup and configuration of Spring applications, allowing for faster development and deployment.	**Learning Curve:** While Spring Boot simplifies many aspects, it still has a steep learning curve, especially for those unfamiliar with the Spring ecosystem.
Microservices Support: It is well-suited for microservices architecture, enabling you to build applications as a set of small, independent services.	**Complexity:** For small projects, Spring Boot might be overkill, introducing unnecessary complexity compared to simpler frameworks.
Embedded Servers: Spring Boot provides embedded servers (like Tomcat, Jetty) which means you can run applications easily without needing to deploy them to an external server.	**Configuration Overhead:** Although it aims to minimize configuration, some advanced configurations can still be complex and require a good understanding of the framework.
Production-Ready Features: Out-of-the-box support for metrics, health checks, and externalized configuration makes it easier to create production-ready applications.	**Memory Consumption:** Applications built with Spring Boot can be heavier in terms of memory consumption compared to lightweight frameworks.

React JS

PROS	CONS
Component-Based Architecture: React's component-based structure promotes reusability, making it easier to manage and scale your application.	**Learning Curve:** Although easier to learn than some frameworks, React still has a learning curve, especially when it comes to concepts like state management and hooks.
Virtual DOM: React's use of a virtual DOM improves performance by minimizing direct manipulation of the real DOM, leading to faster updates.	**JSX Syntax:** JSX can be unfamiliar and may take some time for new developers to get used to, as it blends HTML and JavaScript.
Strong Community and Ecosystem: React has a vibrant community and a wide range of libraries and tools, making it easier to find resources and support.	**State Management Complexity:** Managing state in large applications can become complex. While tools like Redux or Context API can help, they add additional complexity.
Declarative UI: React allows for a declarative approach to building UIs, which can make the code easier to read and debug.	**Frequent Updates:** The React ecosystem evolves rapidly, which can sometimes lead to challenges in keeping up with best practices and updates.

Demo

Next slide →

Thank You