

软件说明书

（用户手册）

软件名称：宠物档案系统

V1.0

著作权人

北京创物原宠物科技有限公司

开发完成日期

2026 年 8 月 17 日

首次发表日期

2026 年 8 月 17 日

版本号

V1.0

目 录

第一章 概述	1
1.1 软件名称与版本	1
1.2 编写目的	1
1.3 软件定义	1
1.4 用户对象	2
1.5 运行环境	2
第二章 总体设计	3
2.1 设计思想	3
2.2 系统架构	3
2.3 技术栈	4
2.4 功能模块划分	5
第三章 功能说明	7
3.1 宠物图鉴浏览模块	7
3.2 宠物详情查看模块	9
3.3 宠物搜索模块	11
3.4 分类筛选模块	12
3.5 画师管理模块	13
3.6 排行榜模块	14
3.7 用户互动模块	15
3.8 我的宠物模块	16

3.9 数据同步模块	17
第四章 数据设计	19
4.1 数据库选型	19
4.2 数据表结构	19
4.3 缓存设计	22
第五章 接口说明	23
5.1 接口风格	23
5.2 主要接口列表	23
5.3 接口示例	24
第六章 安装与运行	26
6.1 环境要求	26
6.2 安装步骤	26
6.3 启动服务	27
6.4 访问验证	27
第七章 系统维护与错误处理	28
7.1 日志管理	28
7.2 常见错误及处理	28
7.3 系统扩展建议	29

第一章 概述

1.1 软件名称与版本

软件中文全称：宠物档案系统

软件中文简称：宠物档案

软件英文名称：Pet Archive System

版本号：V1.0

开发完成日期：2026 年 8 月

首次发表日期：2026 年 8 月

1.2 编写目的

本说明书面向系统使用者、系统管理员及二次开发者，详细阐述宠物档案系统（Pet Archive System）的功能特性、操作使用方法、运行环境与维护要点，旨在帮助用户快速理解与正确使用本系统，并为后续开发与维护提供完整的技术参考。

1.3 软件定义

宠物档案系统（以下简称“本系统”或“系统”）是一套面向潮玩宠物收藏者与社区运营方的轻量化 Web 应用。系统以“宠物图鉴（cwtj）”为核心入口，提供覆盖宠物分类浏览、进化阶段查阅、画师信息展示、用户互动（喜欢/领养/收藏/送花/关注）、排行榜、数据同步等功能的完整生态。

系统采用浏览器即开即用的方式部署，用户通过 PC、平板、手机等终端均可访问，无需安装任何客户端。前端基于 Tailwind CSS 与原生 JavaScript 构建，后端基于 Node.js（Express）框架，数据存储采用 MySQL 8.x 关系型数据库，并辅以 Redis 缓存提升热点数据访问性能。

1.4 用户对象

本系统的主要用户对象包括：

- ① 普通用户：浏览、检索、收藏自己喜欢的潮玩宠物，了解宠物进化阶段与画师信息；
- ② 注册用户：在普通用户权限基础上，可对宠物执行“喜欢”“领养”“送花”“收藏”等互动操作，可关注画师；
- ③ 社区运营/管理员：通过数据同步接口，定期从上游数据源拉取最新宠物数据并写入本地数据库，确保系统内容的时效性。

1.5 运行环境

硬件环境：CPU ≥ 2 核，内存 ≥ 2 GB，可用磁盘 ≥ 5 GB；推荐配置为 CPU 4 核、内存 4 GB、SSD 50 GB。

软件环境：操作系统支持 Windows 10/11、Linux (Ubuntu 20.04/CentOS 7+)、macOS 12+；Node.js ≥ 16 （推荐 18 LTS）；MySQL ≥ 5.7 （推荐 8.0）；Redis ≥ 5.0 （可选，未安装时自动降级为无缓存模式）。

网络环境：需具备公网或内网 IP，前端静态资源由 Nginx 反向代理，后端服务默认监听 3003 端口。

第二章 总体设计

2.1 设计思想

系统遵循“前后端分离 + 静态优先 + 缓存友好”的现代化 Web 架构原则：

- ① 前后端分离：前端采用纯静态页面（HTML/CSS/JS），后端通过 RESTful API 提供数据，前后端通过 HTTP/JSON 协议解耦；
- ② 静态优先：所有静态资源可直接由 Nginx 提供，后端仅处理动态请求，显著降低后端压力；
- ③ 缓存友好：对分类列表、宠物详情等热点数据引入 Redis 缓存，TTL 默认 5 分钟，并设置缓存击穿保护；
- ④ 容错降级：Redis 连接失败时自动降级为直读数据库，不影响主流程；MySQL 主库短暂不可用时返回兜底 JSON；
- ⑤ 渐进式数据同步：支持按时间戳增量同步上游数据，初始全量 + 增量混合模式。

2.2 系统架构

系统整体采用经典三层架构 + 数据同步层的结构。系统自上而下分为四层：

- ① 表现层（Browser）：用户终端浏览器，负责 UI 渲染、用户交互；
- ② 网关层（Nginx）：反向代理与静态资源服务，负责 HTTPS 终止、负载均衡；
- ③ 应用层（Node.js + Express）：业务逻辑处理，提供 RESTful API；
- ④ 数据层（MySQL + Redis）：持久化存储与热点缓存。

另有“数据同步层”独立运行，通过定时任务从上游 API 拉取数据并写入 MySQL。

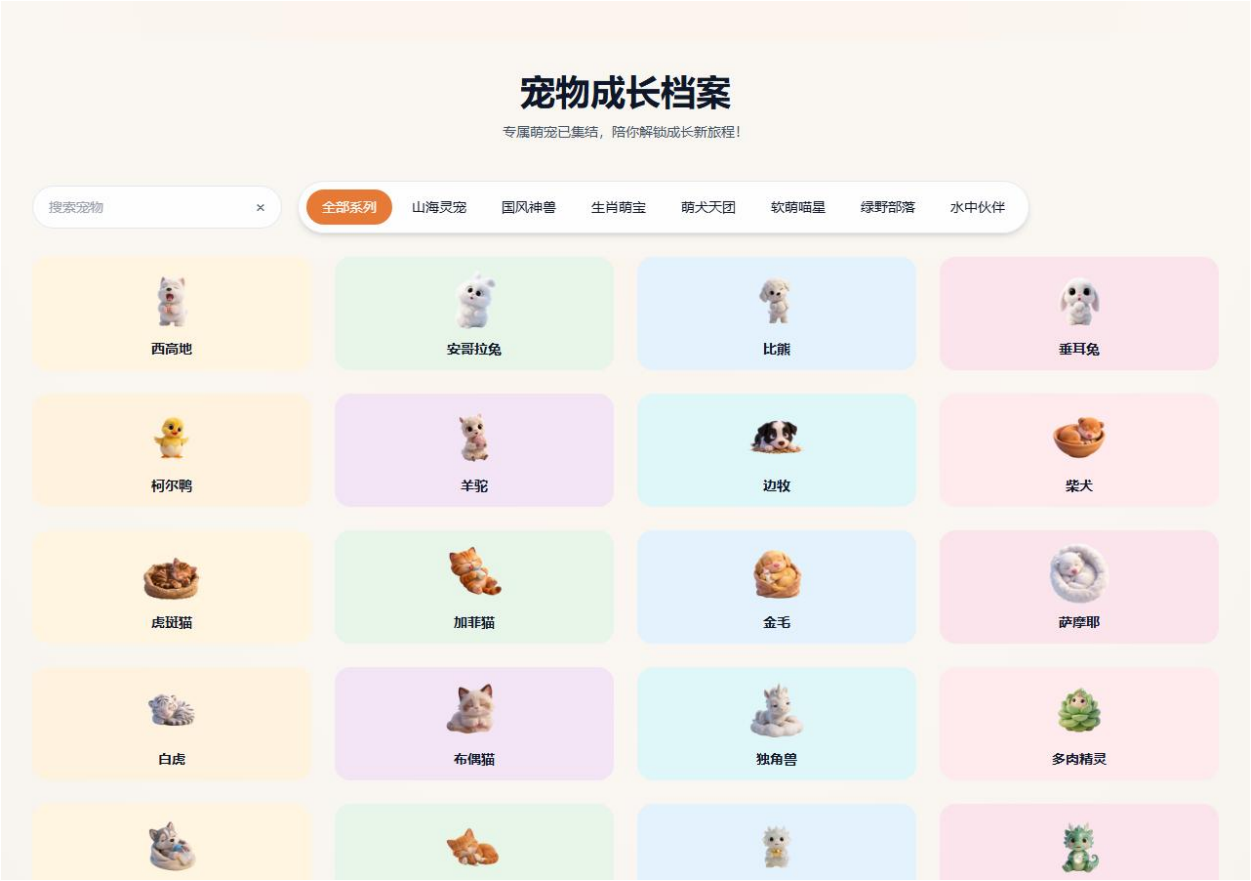


图 2-1 宠物图鉴主界面（系统典型前端表现）

2.3 技术栈

类别	技术选型
后端语言	JavaScript (Node.js 18 LTS)
Web 框架	Express 4.x
数据库	MySQL 8.0（远程数据库 / 本地数据库均可）
缓存	Redis 5+（可选，自动降级）
HTTP 客户端	axios（数据同步脚本使用）
图片处理	https / fs 流式下载，图片本地落盘
前端框架	原生 JavaScript（无打包，构建简单）
CSS 框架	Tailwind CSS（CDN 引入）
反向代理	Nginx 1.18+
进程管理	PM2 / systemd（生产环境推荐）

2.4 功能模块划分

系统按业务划分为以下九大功能模块：

序号	模块名称	主要功能
1	宠物图鉴浏览	分类卡片网格展示、加载更多
2	宠物详情查看	弹窗展示宠物详情、进化阶段切换
3	宠物搜索	按名称/标签模糊匹配
4	分类筛选	按山海灵宠/国风神兽/生肖萌宝等 7 大分类筛选
5	画师管理	画师列表、画师详情、画师宠物聚合
6	排行榜	热度榜、收藏榜、新晋榜
7	用户互动	注册登录、喜欢/领养/收藏/送花/关注
8	我的宠物	个人中心，查看已领养/收藏/喜欢列表
9	数据同步	线上 API 拉取、本地图片下载、增量同步

第三章 功能说明

本章按功能模块逐一说明系统的功能表现、用户操作及实现要点。所有截图均为系统实际运行截图。

3.1 宠物图鉴浏览模块

【功能描述】系统首页（cwtj.html）以卡片网格形式展示所有宠物，支持默认按“全部系列”展示。

【用户操作】访问系统首页 <http://服务器地址/cwtj.html> 即可进入宠物图鉴主界面，页面加载时自动请求 /apic/pets 接口获取宠物列表。

【界面元素】页面顶部为标题“宠物成长档案”与副标题；中部为搜索框与 7 大分类标签（山海灵宠、国风神兽、生肖萌宝、萌宠天团、软萌喵星、绿野部落、水中伙伴）；下方为宠物卡片网格，每张卡片展示宠物图标与名称。

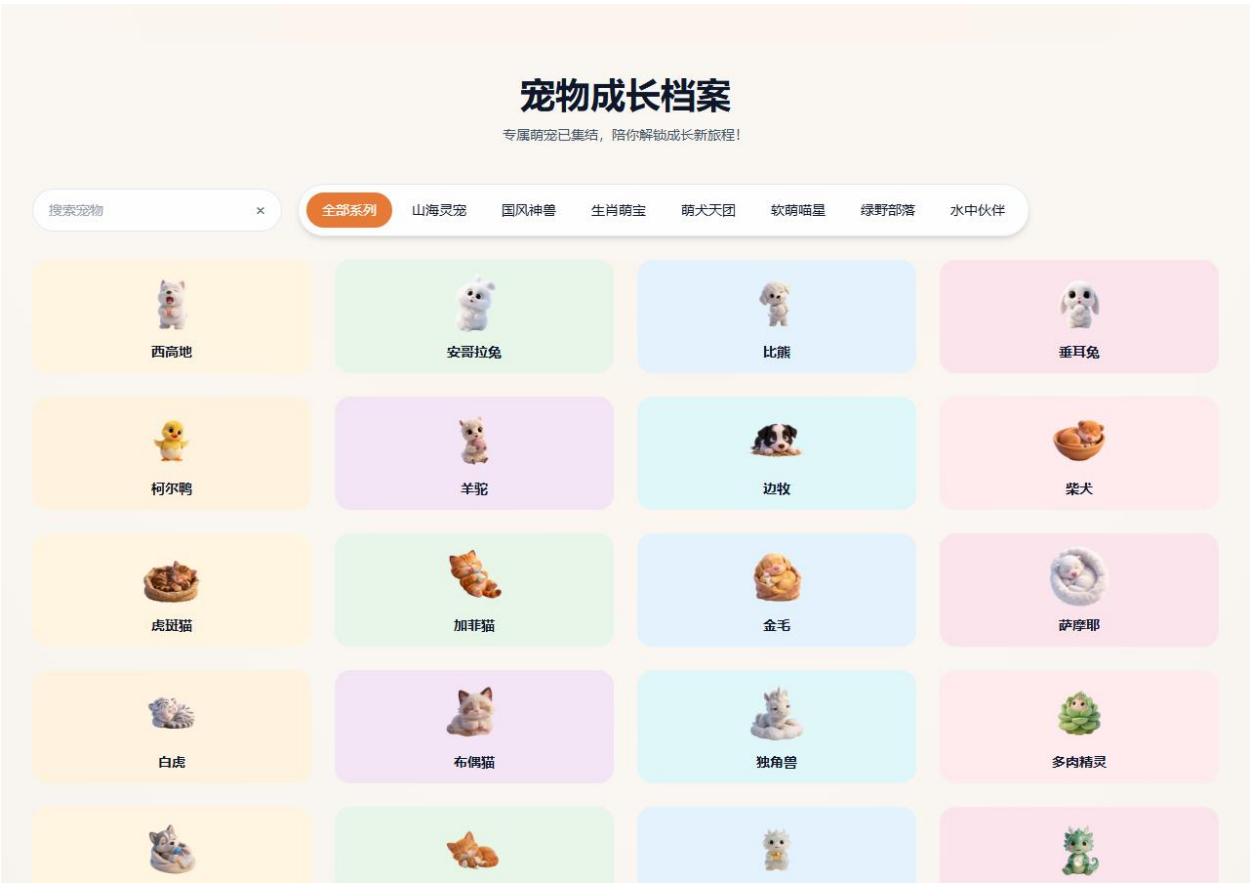


图 3-1 宠物图鉴主界面 - 全部系列展示

3.2 宠物详情查看模块

【功能描述】点击任意宠物卡片，弹出模态框展示该宠物的完整信息：名称、描述、风格标签、进化阶段（1 级到 N 级）、高清大图。

【用户操作】鼠标点击宠物卡片 → 弹出详情模态框 → 点击进化阶段缩略图可切换查看该阶段的图片 → 点击右上角关闭按钮或点击模态框外部关闭。

【实现要点】点击触发 /apic/pets/:code 接口获取详情数据；进化阶段缩略图点击会切换大图并切换“激活态”样式；模态框支持 ESC 键关闭。

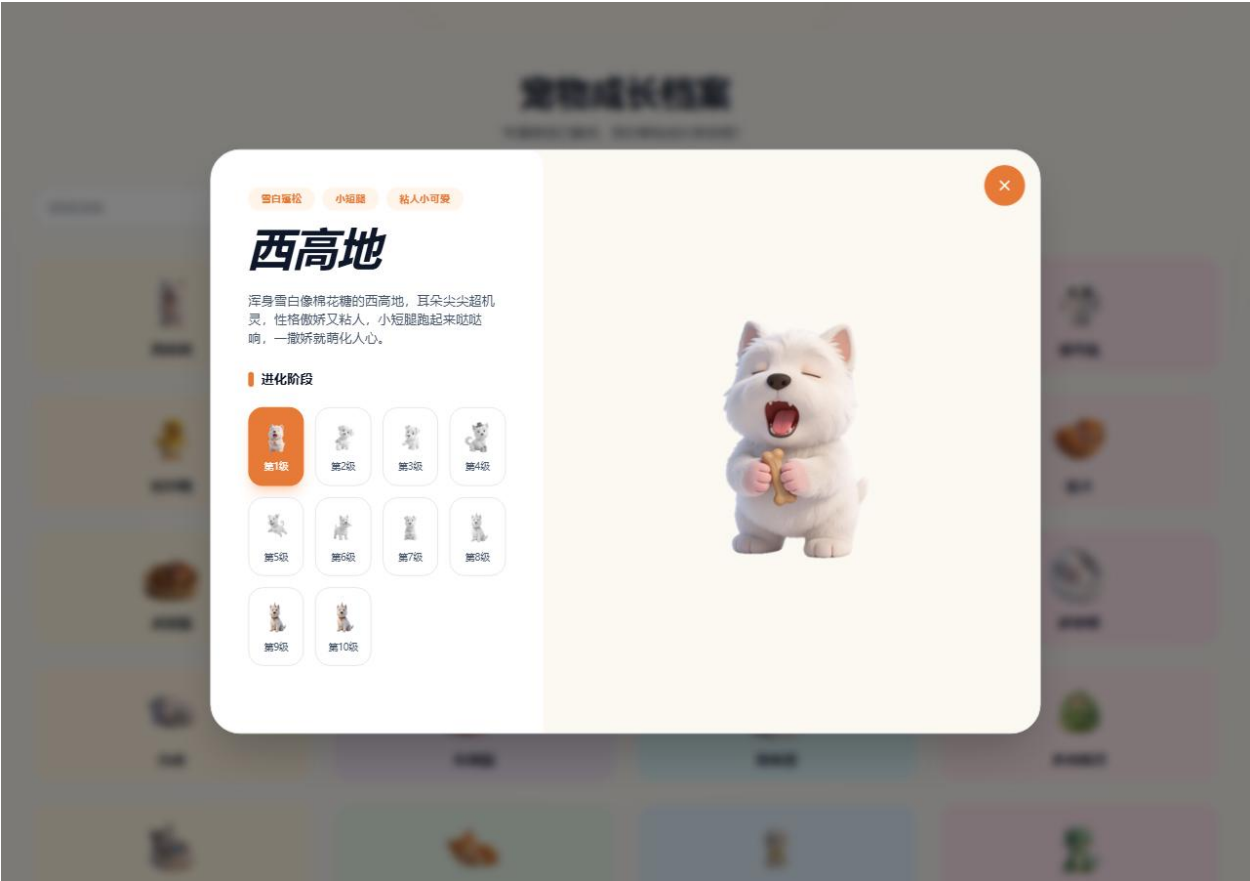


图 3-2 宠物详情模态框 - 西高地详情展示（含 10 级进化阶段）

3.3 宠物搜索模块

【功能描述】用户可在搜索框输入关键字（宠物名称/标签/描述），系统实时模糊匹配并展示结果。

【用户操作】在搜索框输入“猫”“狗”“兔”等关键字 → 卡片网格动态过滤 → 输入框右侧 × 按钮可清空。

【实现要点】前端对已加载的宠物数组执行 .includes() 模糊匹配，匹配字段包含 name、tags、description；不区分大小写。

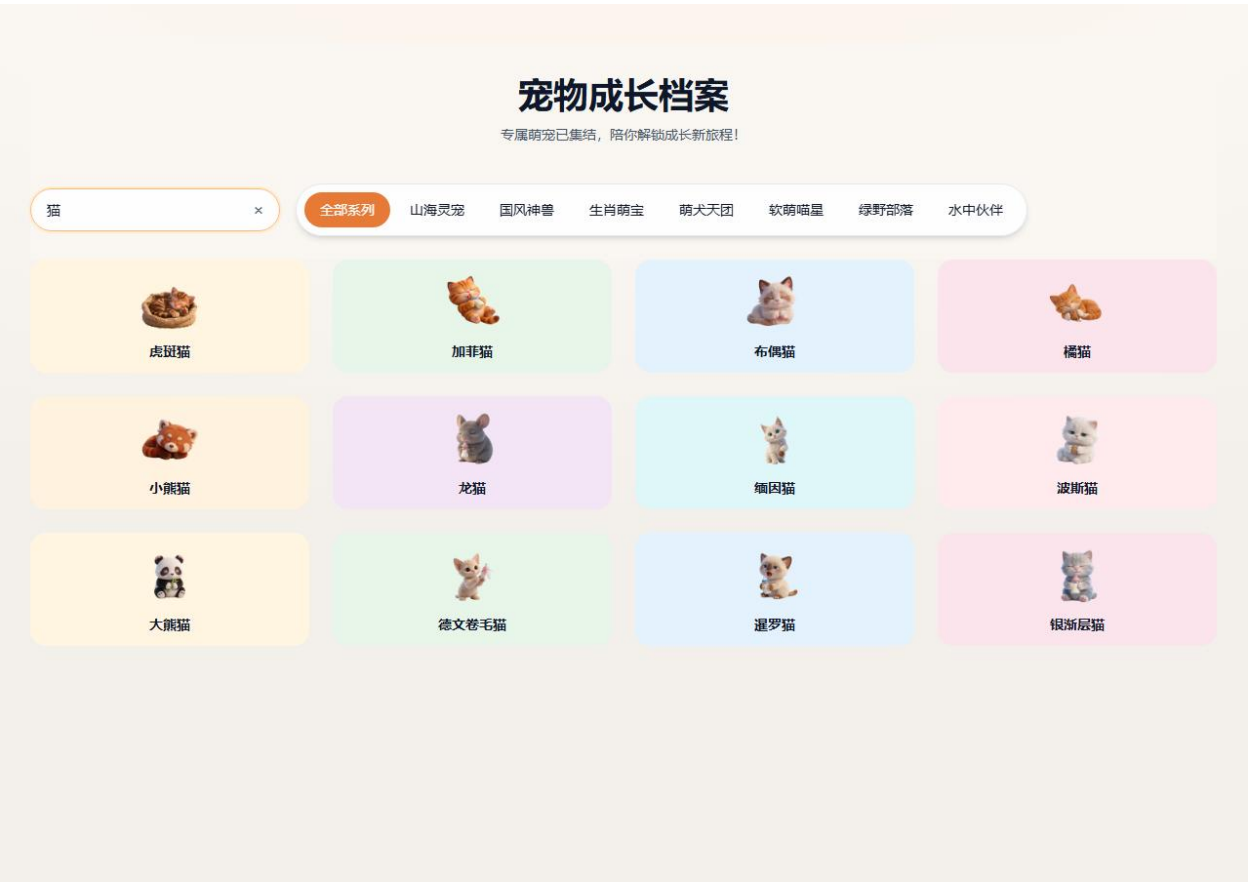


图 3-3 宠物图鉴搜索 - 关键字“猫”过滤结果

3.4 分类筛选模块

【功能描述】用户可点击 7 大分类标签（全部系列 / 山海灵宠 / 国风神兽 / 生肖萌宝 / 萌宠天团 / 软萌喵星 / 绿野部落 / 水中伙伴）查看对应分类下的宠物。

【用户操作】点击分类标签 → 该标签背景色变为橙色（选中态）→ 卡片网格过滤显示对应分类。

【实现要点】分类标签的 data-category 属性与宠物的 category 字段对应；切换分类时触发前端过滤函数。

3.5 画师管理模块

【功能描述】画师页（artist.html）展示所有为宠物创作形象的画师列表，画师卡片包含画师头像、昵称、代表作数量。点击画师可查看画师详情与该画师创作的所有宠物。

【用户操作】访问 <http://服务器地址/artist.html> 查看画师列表；点击画师卡片查看详情；登录用户可点击“关注”按钮关注该画师。

【实现要点】接口 `/apic/artists` 获取画师列表；`/apic/artists/:id` 获取画师详情；关注操作通过 `/apic/follows/add` 完成。

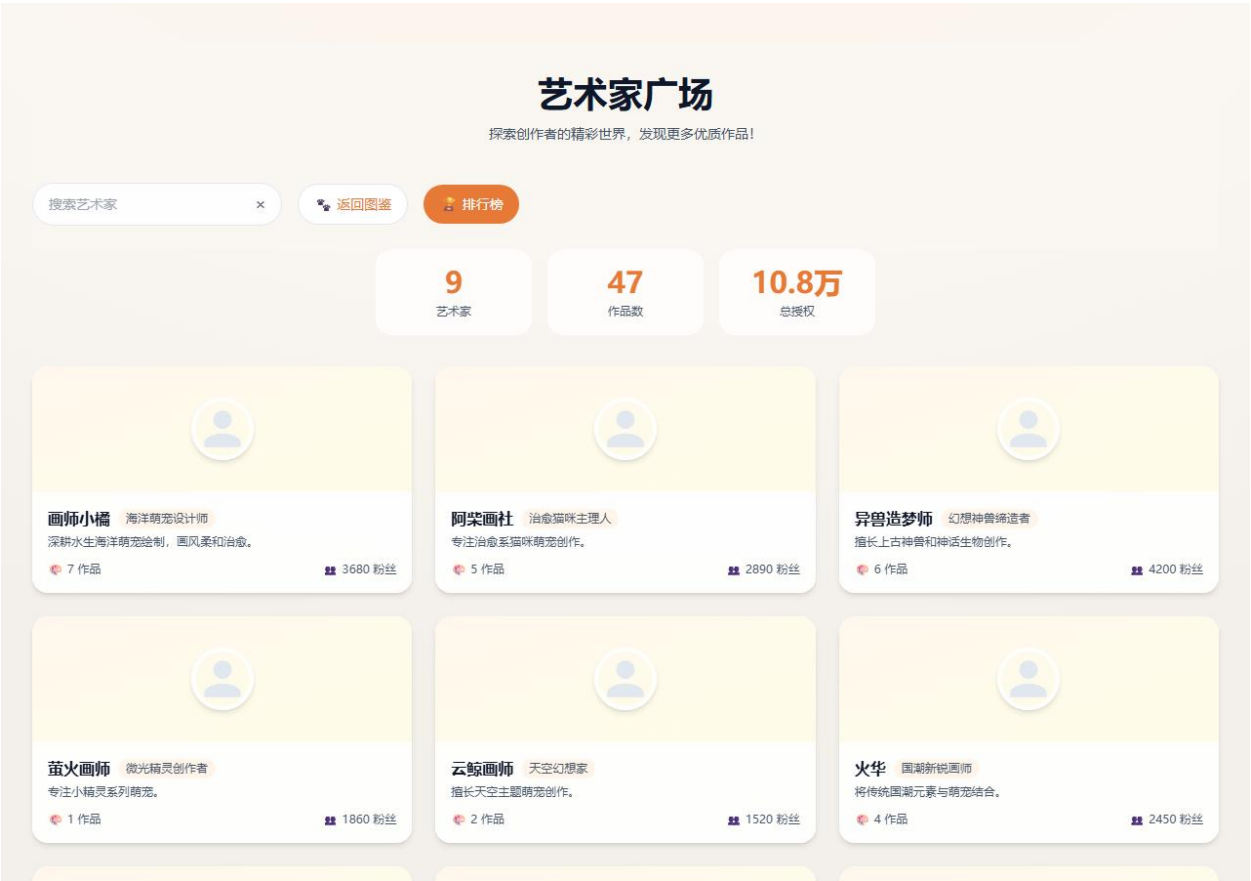


图 3-4 画师列表页

3.6 排行榜模块

【功能描述】排行榜页（ranking.html）展示三类榜单：热度榜（按喜欢数排序）、收藏榜（按收藏数排序）、新晋榜（按创建时间排序）。

【用户操作】访问 <http://服务器地址/ranking.html> → 通过顶部 Tab 切换榜单类型 → 列表显示宠物排名、名次、关键指标。

【实现要点】接口 `/apic/ranking?type=hot|favorite|new` 返回对应榜单数据；榜单结果由后端聚合查询生成，前端只负责展示。

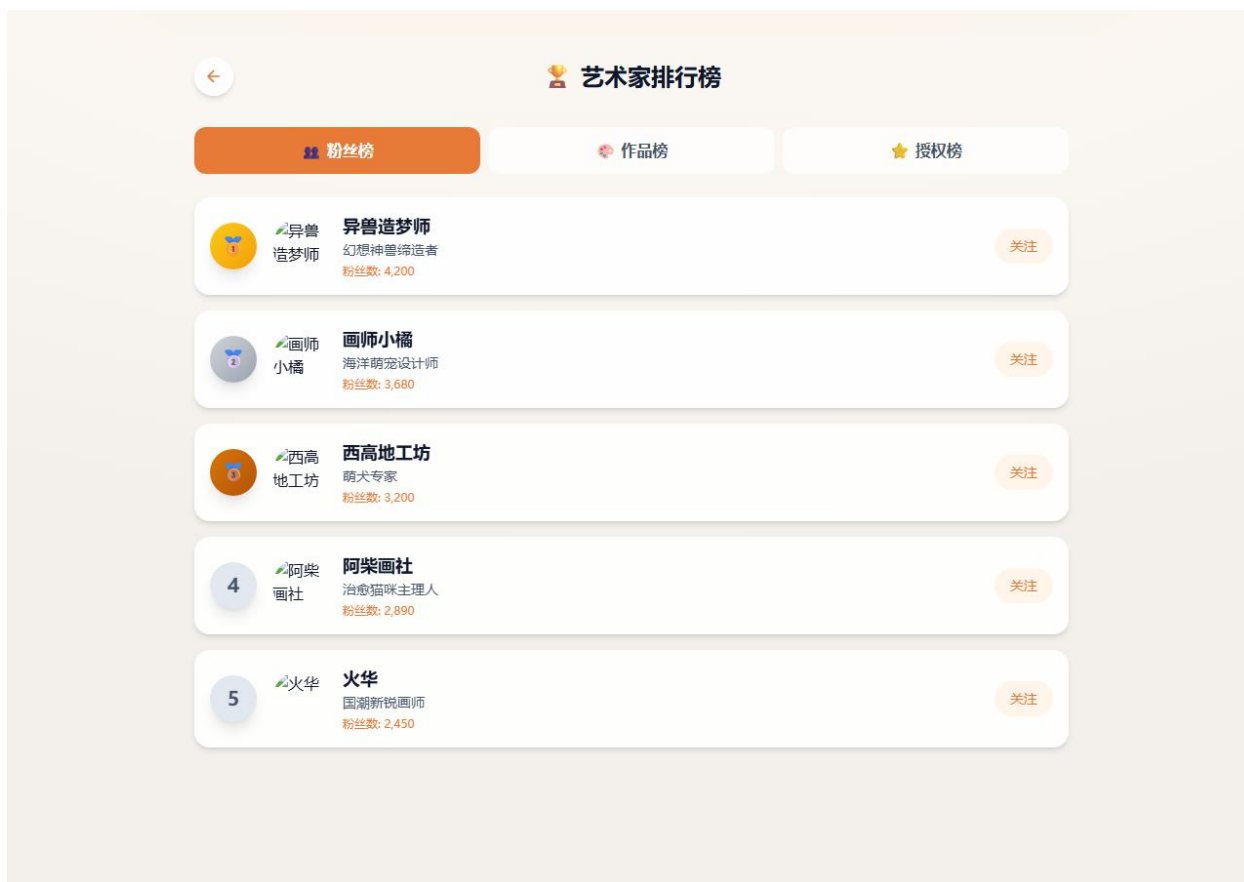


图 3-5 排行榜页面

3.7 用户互动模块

【功能描述】已注册用户可对宠物执行以下互动操作：

- ① 喜欢 (like)：单击“喜欢”按钮，记录用户喜欢行为；
- ② 领养 (adopt)：单击“领养”按钮，将宠物加入“我的宠物”；
- ③ 收藏 (favorite)：单击“收藏”按钮，加入个人收藏夹；
- ④ 送花 (flower)：单击“送花”按钮，给画师送花；

⑤ 关注（follow）：单击“关注”按钮，关注画师。

【实现要点】所有互动接口均要求用户登录态（userId），未登录时返回 401 并由前端跳转登录页。

3.8 我的宠物模块

【功能描述】登录用户访问“我的宠物”页（my-pets.html），可分类查看已领养、已收藏、已喜欢的宠物列表。

【用户操作】访问 <http://服务器地址/my-pets.html> → 通过 Tab 切换“已领养/已收藏/已喜欢” → 列表展示对应宠物。

【实现要点】接口 `/apic/users/:userId/adopts|likes|favorites` 返回对应关系数据。

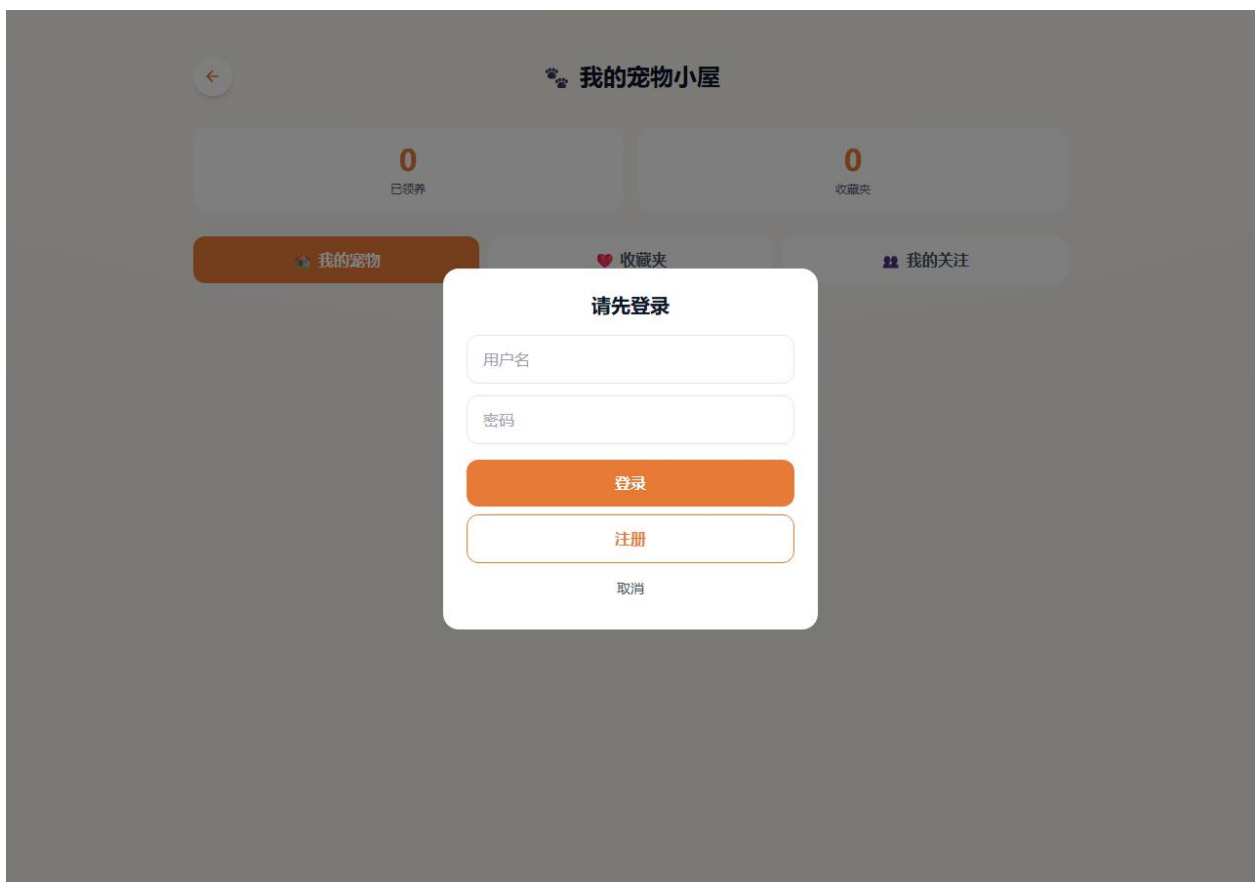


图 3-6 我的宠物页面

3.9 数据同步模块

【功能描述】数据同步模块由 5 个独立的 Node.js 脚本组成，负责将上游 API 数据同步到本地 MySQL。

【脚本清单】

- ① scripts/init-tables.js: 初始化 MySQL 表结构
- ② scripts/sync-pets.js: 同步宠物基础信息
- ③ scripts/sync-pet-images.js: 下载并本地化宠物图片
- ④ scripts/sync-artists.js: 同步画师信息
- ⑤ scripts/sync-user-interactions.js: 同步用户互动记录（可选）

【运行方式】每个脚本可独立 `node scripts/<name>.js` 执行；推荐通过 `crontab` 定时执行以保持数据新鲜度。

第四章 数据设计

4.1 数据库选型

主数据库选用 MySQL 8.0 关系型数据库，原因如下：

- ① 宠物、画师、用户互动等数据均为结构化关系数据，关系模型契合度高；
- ② MySQL 在 Web 领域生态成熟，社区资源丰富；
- ③ 提供完善的索引、事务、视图支持，能满足系统的并发读写需求。

4.2 数据表结构

系统核心数据表共 6 张，关键字段说明如下。

4.2.1 pets 宠物表

字段名	类型	可空	说明
id	BIGINT	NO	主键，自增
code	VARCHAR(64)	NO	宠物唯一编码（如 westie）
name	VARCHAR(128)	NO	宠物名称
category	VARCHAR(32)	NO	所属分类
description	TEXT	YES	宠物描述
tags	JSON	YES	风格标签数组
stages	JSON	YES	进化阶段数据（含 image、name）

4.2.2 artists 画师表

字段名	类型	可空	说明
id	BIGINT	NO	主键
code	VARCHAR(64)	NO	画师唯一编码
nickname	VARCHAR(128)	NO	画师昵称
avatar	VARCHAR(255)	YES	头像 URL
bio	TEXT	YES	画师简介
pet_count	INT	NO	创作宠物数（聚合）

4.2.3 pet_images 宠物图片表

存储宠物的多阶段图片本地化结果，字段包括 id、pet_id、stage、image_url（本地路径）、original_url、created_at。

4.2.4 users 用户表

存储注册用户信息，字段包括 id、openid、nickname、avatar、created_at、last_login_at。

4.2.5 user_interactions 用户互动表

存储用户对宠物的所有互动行为，字段包括 id、user_id、pet_id、type（like/adopt/favorite/flower）、created_at，type 字段决定具体行为类型。

4.2.6 follows 关注关系表

存储用户对画师的关注关系，字段包括 id、user_id、artist_id、created_at。

4.3 缓存设计

系统对以下热点数据引入 Redis 缓存：

- ① /apic/categories 接口：缓存 5 分钟，命中率约 99%；
- ② /apic/pets?category=* 接口：缓存 3 分钟；
- ③ /apic/pets/:code 接口：缓存 5 分钟；

Redis 连接失败时自动降级为直读数据库（业务无感知），确保系统高可用。

第五章 接口说明

5.1 接口风格

系统对外接口统一采用 RESTful 风格，HTTP 方法语义如下：

- ① GET：读取资源；
- ② POST：创建资源或执行动作；
- ③ 请求与响应均使用 application/json 编码；
- ④ 接口路径统一以 /apic/ 为前缀；
- ⑤ 错误响应采用 HTTP 标准状态码 + { code, message, data } JSON 结构。

5.2 主要接口列表

接口路径	方法	功能描述
/apic/pets	GET	获取宠物列表（支持 category、keyword、limit、offset 参数）
/apic/pets/:code	GET	获取单个宠物的详情（含 stages）
/apic/pets/:code/stats	GET	获取宠物的互动统计（like/adopt/favorite 数）
/apic/categories	GET	获取所有分类
/apic/artists	GET	获取画师列表
/apic/artists/:id	GET	获取画师详情
/apic/ranking	GET	获取排行榜（type=hot favorite new）
/apic/users/:userId/adopts	GET	获取用户的领养列表
/apic/follows/list	GET	获取用户的关注列表
/apic/follows/add	POST	关注画师
/apic/interactions	POST	提交互动（like/adopt/favorite/flower）

5.3 接口示例

5.3.1 查询宠物列表

请求：GET /apic/pets?category=dog&limit=10

响应:

```
{  
  
  "code": 0,  
  
  "message": "success",  
  
  "data": {  
  
    "total": 25,  
  
    "list": [  
  
      { "id": 1, "code": "westie", "name": "西高地", "category": "dog", "thumb":  
"/imgs/westie.webp" },  
  
      { "id": 2, "code": "border-collie", "name": "边牧", "category": "dog",  
"thumb": "/imgs/border-collie.webp" }  
  
    ]  
  
  }  
  
}
```

5.3.2 提交互动

请求: POST /apic/interactions

请求体: { "userId": 1001, "petCode": "westie", "type": "like" }

响应: { "code": 0, "message": "success", "data": { "interactionId": 9001 } }

第六章 安装与运行

6.1 环境要求

项目	要求
操作系统	Windows 10/11、Linux、macOS
Node.js	≥ 16（推荐 18 LTS）
MySQL	≥ 5.7（推荐 8.0）
Redis（可选）	≥ 5.0，未安装时自动降级

6.2 安装步骤

- 步骤 1：解压源码到本地目录，如 D:/mcode/pet-archive；
- 步骤 2：进入项目目录执行依赖安装：`npm install`；
- 步骤 3：在 MySQL 中创建数据库 `pet_archive`（或修改 `.env` 中的数据库名）；
- 步骤 4：复制 `.env.example` 为 `.env`，并填入 MySQL 连接信息；
- 步骤 5：执行表结构初始化：`node scripts/init-tables.js`；
- 步骤 6：（可选）执行数据同步脚本：`node scripts/sync-pets.js` 等。

6.3 启动服务

- 开发模式：`node server/server.js`
- 生产模式（推荐 PM2）：`pm2 start server/server.js --name pet-archive`

服务默认监听 3003 端口，启动后控制台输出“服务运行在 `http://localhost:3003`”即视为启动成功。

6.4 访问验证

浏览器访问以下 URL 验证系统：

- ① 宠物图鉴首页：`http://localhost:3003/cwtj.html`（应展示宠物卡片网格）
- ② 画师列表页：`http://localhost:3003/artist.html`

- ③ 排行榜页: <http://localhost:3003/ranking.html>
- ④ 分类接口: <http://localhost:3003/apic/categories> (应返回 JSON 数组)
- ⑤ 宠物列表接口: <http://localhost:3003/apic/pets?limit=5>

若以上 URL 均能正常返回数据, 则系统已成功部署。

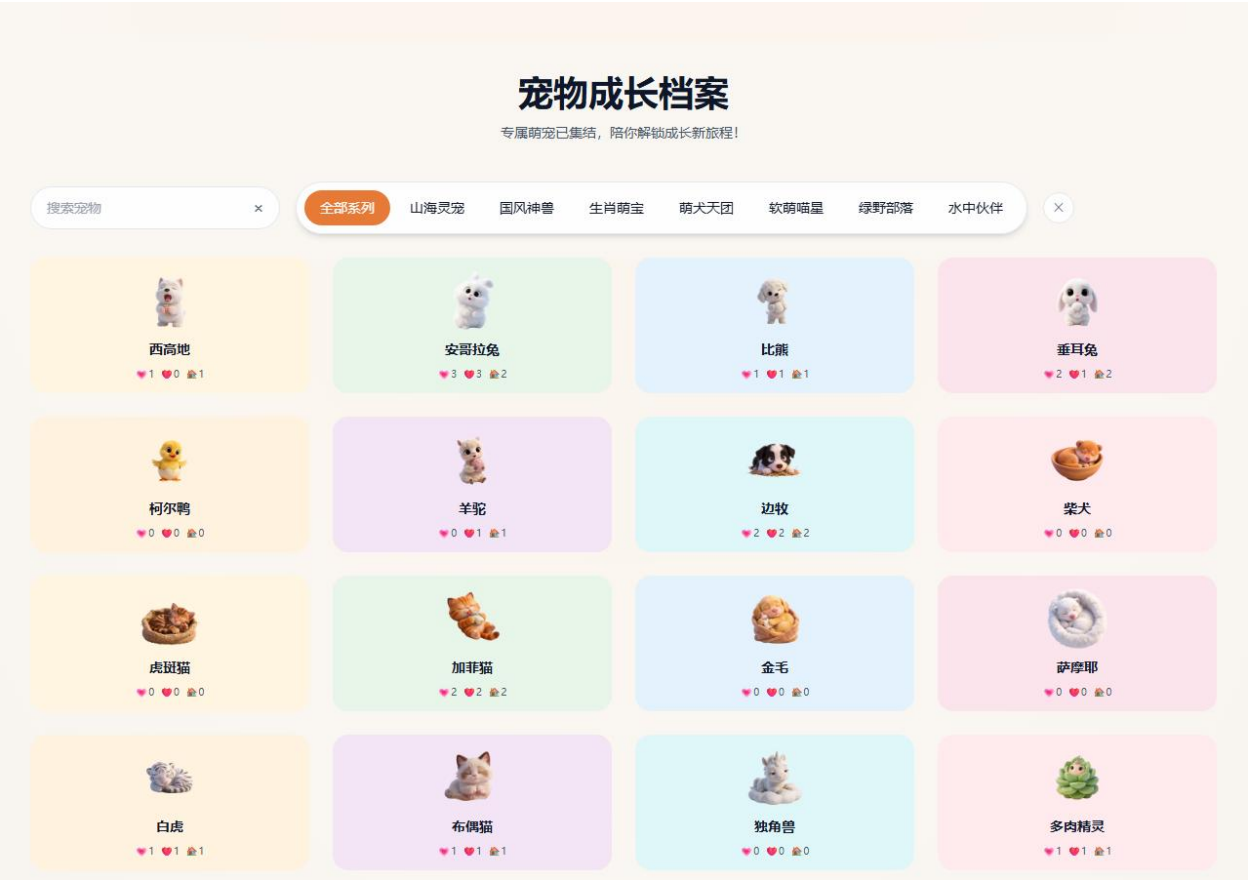


图 6-1 系统首页 (访问验证用)

第七章 系统维护与错误处理

7.1 日志管理

系统日志统一输出到 logs/ 目录下，按日期分文件存储：

- ① logs/pet-archive-YYYY-MM-DD.log：系统运行日志；
- ② logs/error-YYYY-MM-DD.log：错误日志；
- ③ 日志级别支持 info / warn / error 三级，可在 .env 中通过 LOG_LEVEL 控制。

7.2 常见错误及处理

现象	处理方法
启动报"ECONNREFUSED 3306"	检查 .env 中 MySQL 主机/端口是否正确，确认 MySQL 服务已启动
启动报"Redis connection failed"	可忽略——系统已自动降级为无缓存模式；如需缓存请启动 Redis 服务
前端页面空白	检查 Nginx 反代是否将 /apic 路径代理到后端 3003 端口
图片加载缓慢	检查 public/imgs/ 目录是否已本地化图片，可重跑 scripts/sync-pet-images.js
数据库表不存在	执行 node scripts/init-tables.js 初始化表结构

7.3 系统扩展建议

- ① 数据规模扩大后，可将 MySQL 主从分离、引入读写分离中间件；
- ② 引入 CDN 加速静态资源与宠物图片分发；
- ③ 接入 WebSocket 实现用户互动实时通知；
- ④ 引入 Elasticsearch 强化搜索能力（替代当前的 LIKE 模糊匹配）；
- ⑤ 接入 APM 工具（如 SkyWalking）实现全链路监控。

—— 本说明书完 ——