Design and implementation of pet adoption platform system based on Django

This paper describes in detail the design and implementation of a pet adoption platform system based on Django framework. With the improvement of people's awareness of pet welfare and the development of Internet technology, the traditional way of pet adoption can no longer meet the needs of modern society. The aim of this project is to build a fully functional and user-friendly online pet adoption platform that connects pets in need of a new owner with users interested in adopting a pet.

The system uses Django 3.2.7 as the back-end framework, combined with HTML, CSS and JavaScript front-end technology, the user registration login, pet information display, pet collection, adoption application and other core functions. The paper introduces the requirements analysis, architecture design, database design, module implementation and test evaluation process of the system in detail, highlighting the characteristics of the system in user experience, data security and function expansion.

Through the realization of this system, not only provides a practical pet adoption platform, but also provides a reference model for the development of similar social service Web applications. The research results show that the system can effectively meet the needs of users to adopt pets, improve the efficiency and success rate of pet adoption, and provide technical support for pet welfare.

Keywords: Pet adoption platform, Django framework, Web application development, responsive design, user experience

**Education Use Consent**

I hereby give my permission for this project to be shown to other University of w students and to be distributed in an electronic form.

Name: Signature:

**Acknowledgements**

We would like to thank Dr Kevin Bryson, Dr Mireilla Bikanga Ada, and Dr Marry Ellen Foster for their expanding guidance and supervision throughout the project.

Also, we would like to give many thanks to all the teaching assistants who were always there for us when we had problems to discuss.

Contents

[Education Use Consent 1](#_Toc1591067848)

[Acknowledgements 2](#_Toc610948892)

[Chapter 1 Introduction 4](#_Toc1098711537)

[1.2 Research Objectives 4](#_Toc1980405453)

[Chapter 2 Functional requirement 5](#_Toc825403718)

[2.1 User Story 5](#_Toc1963412453)

[2.2 Function module 5](#_Toc839377769)

[2.3 Nonfunctional requirement 6](#_Toc602086440)

[Chapter 3 System design 6](#_Toc323852416)

[3.1 System architecture 6](#_Toc1263994214)

[3.2 Module Division 8](#_Toc1042518574)

[3.3 System Deployment Architecture 8](#_Toc290597345)

[3.4 Database design 8](#_Toc691764137)

[3.5 Site map and wireframes 9](#_Toc3385701)

[Chapter 4 System implementation 11](#_Toc1068901885)

[4.1 General implementation method 11](#_Toc1333274040)

[4.2 Actual Implementation 11](#_Toc1492417482)

[4.3 Technology stack selection 12](#_Toc451623014)

[4.4 Implementation of Core functions 12](#_Toc1220787800)

[4.5 Solutions to Technical difficulties 14](#_Toc721791162)

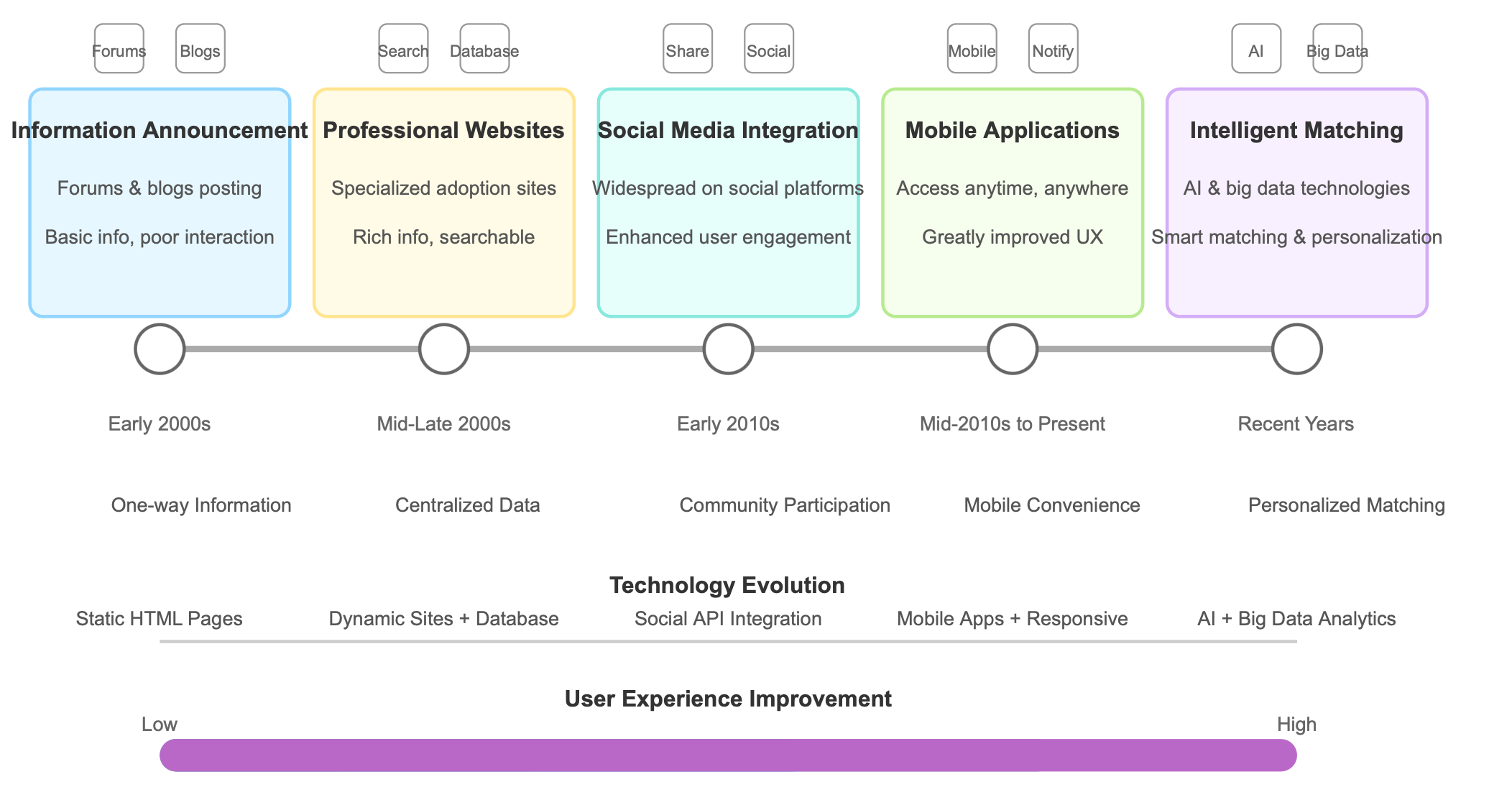
[Chapter 5 Testing 15](#_Toc8937831)

[Reference 18](#_Toc2041753974)

# Chapter 1 Introduction

## 1.1 Research background and motivation

With the rapid development of social economy and the improvement of people's living standards, pets have become an indispensable part of many family lives. At the same time, the number of stray pets and pets that need to be relocated is also increasing, and the problem of pet adoption and resettlement is becoming increasingly prominent. Traditional pet adoption methods, such as through pet shelters, pet stores or personal word-of-mouth recommendations, have problems such as limited information dissemination, lack of transparency and inefficiency.In this evolution process, the functions of the pet adoption platform are increasingly rich, the user experience is constantly optimized, and the technical means are constantly innovative, providing more and more powerful support for pet adoption.



1-1 img1:Development and evolution of pet adoption platform

The research motivation of this project stems from the concern of pet welfare and the exploration of the application of technology in the field of social welfare. By building a fully functional and user-friendly pet adoption platform, the efficiency and success rate of pet adoption can be effectively improved, suitable families can be found for pets in need, and more convenient and transparent services can be provided for users who are willing to adopt pets.

## 1.2 Research Objectives

The main goal of this study is to design and implement a pet adoption platform system based on Django framework to meet the needs of all parties in the process of pet adoption. Specific objectives include:

1. **Build a well-functioning platform,** 2. **Provide user-friendly interface,** 3. **Ensure data security and privacy,** 4. **Optimize pet matching efficiency,** 5. **Provide system scalability,** 6. **Realize technical innovation and practice.**

By achieving the above goals, this study aims to promote the digitization and networking of the pet adoption process, provide technical support for pet welfare undertakings, and provide a reference model for the development of similar social service Web applications.

# Chapter 2 Functional requirement

## 2.1 User Story

Based on the information collected, we put together a series of user stories that show the multilayered needs of ordinary users (adopters). As potential adopters, they first hope to register an account and log in to the system, so that they can successfully use the functions provided by the platform; Next, they expect to be able to browse lists of pets for adoption so they can quickly locate someone they are interested in. In addition, by fine screening of pet type, age, region and other conditions, they can more efficiently find pets that meet individual conditions; After determining the target, the user also wants to view the pet's details and photos in order to get a full picture of the pet's condition; At the same time, the collection function enables them to save pets of interest for easy comparison and review later; Once the intention is confirmed, the potential adopter submits an application to start the adoption process and wants to be able to see the status of the application in real time to keep track of the progress of the process; Finally, in order to ensure that personal information is updated and accurate, they also need to manage their personal information. Taken as a whole, this series of user stories builds a complete service chain from initial exploration to in-depth understanding, from intention collection to process tracking, and then to personal information management, just like a well-planned building, from the entrance to each functional area to provide users with seamless and convenient services.

## 2.2 Function module

### 1. User management module

The user management module is equivalent to the access control of the system, and is responsible for user registration, login, personal information management and password retrieval. Strict authentication mechanism ensures that only legitimate users can access internal system resources.

### 2. Pet information management module

The module carries a detailed display of pet information, including basic information about the pet, multiple pictures, detailed descriptions and status updates. Users can classify and manage the information according to the type of pet (such as cat, dog, rabbit, etc.), and update the status of the pet in real time, so that the data display is always fresh and accurate.

### 3. Adoption process and interactive module

This module provides users with the whole process from browsing, saving, submitting adoption applications to tracking the progress of the application. The user can make a collection decision based on the details and pictures of the pet and submit an adoption application after confirming the intention. The system feeds back the application status in real time, providing users with a transparent adoption experience.

### 4. Background management and data analysis module

The back-end management module of the platform is similar to the control center of a building, where administrators can review pet information, manage user accounts, handle complaints, and monitor adoption applications. At the same time, the data analysis and report function carries out statistics and analysis on the platform operation data, providing data support for system optimization and decision-making.

## 2.3 Nonfunctional requirement

While meeting basic functions, the platform must also meet the following non-functional requirements to ensure overall system performance, user experience, and security:

### 1. Performance requirements

The system response time requires that the page load does not exceed 3 seconds, the operation response does not exceed 1 second, and supports multi-user concurrent operations, which is like configuring a high-efficiency engine for the building to ensure that it can still respond quickly to user requests under high loads. At the same time, the system architecture supports horizontal scaling to cope with the growing number of users.

### 2. Safety requirements

In order to prevent malicious attacks and data leaks, the platform adopts measures such as password encryption, form verification and data disinfection, and protects against common vulnerabilities such as SQL injection, XSS and CSRF. Strict permission control ensures that different roles can only access the authorization function, and builds a multi-layer protective network for the system.

### 3. Availability and reliability requirements

The platform design requires the system to be available 7×24 hours, with fault tolerance and regular data backup mechanism. No matter what kind of emergency, the system can restore normal operation in time through the perfect emergency plan, thus minimizing the risk of service interruption.

### 4. User experience and maintenance requirements

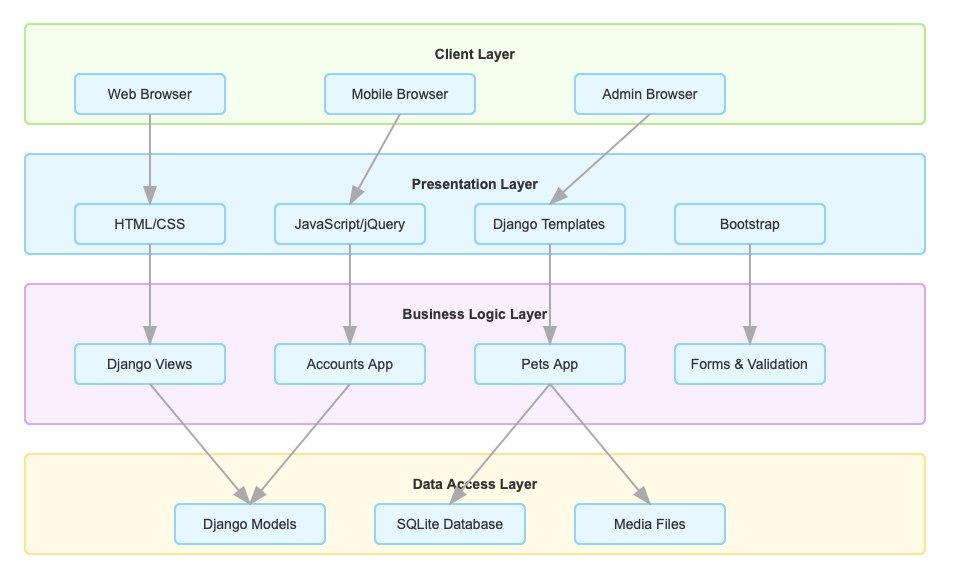
The interface design pursues simple, intuitive and easy operation, while taking into account both mobile and desktop users through responsive design. The modular code structure, detailed documentation and coverage of unit tests make the system highly operable in maintenance and extension.

## Chapter 3 System design

This chapter will introduce the design scheme of the pet adoption platform system in detail, including system architecture, technology stack selection, database design and detailed design of each functional module, to provide a complete blueprint for system implementation.

## 3.1 System architecture

The pet adoption platform uses a classic three-tier architecture, consisting of a presentation layer, a business logic layer, and a data access layer, combined with the MVC (Model-View-Controller) pattern of the Django framework (commonly referred to as MVT: Model-View-Template in Django). Figure 3-1 shows the system architecture.



3-2 img2:System Architecture Diagram

### 3.1.1 Overall architecture

Our system uses a browser/server (B/S) architecture, users access the system through the browser, and the server is responsible for processing various requests and returning the corresponding response data. In this architecture, the system is divided into multiple layers, each with unique and important responsibilities. The first is the presentation layer, which is mainly responsible for the display of the user interface and the interaction with the user. It uses front-end technologies such as HTML, CSS, JavaScript to build an intuitive and friendly user interface, and ensures good display effects on different devices through responsive design. Then there is the business logic layer, which realizes the core business logic of the system, not only handles user requests, executes business rules, but also plays a coordinating role between various functional modules to ensure the smooth and efficient operation of the whole system. The last is the data access layer, which focuses on the interaction with the database and realizes the operation of adding, deleting, modifying and checking the data to ensure the consistency and security of the data. At the same time, in the Django framework, we use the MVT (Model-View-Template) Model to design the system, in which the Model part defines the data structure and the relationship between the database tables, and implements the data access logic. For example, UserProfile, PetsType, PetsInfo, etc. The View layer mainly processes HTTP requests and responses, is responsible for implementing specific business logic, calls the Model layer to obtain data and passes it to the Template layer for display, such as user registration, pet list display, pet details and other functions; The Template layer is responsible for defining the HTML structure of the page, rendering dynamic data through the template language provided by Django, and realizing page reuse and inheritance, such as base.html, index.html, pets.html and other template files. Together to build a fully functional and good user experience electric scooter rental platform。

## 3.2 Module Division

First of all, **the account management module** (accounts) is mainly responsible for the implementation of user registration, login and information management and other functions, while undertaking the management of user rights and identity verification, to ensure that only legitimate users can access the internal resources of the system. Secondly, **the Pet management module (pets)** focuses on the management, display and search of pet information, and supports the processing of pet collection and adoption applications, so as to provide users with convenient pet information query and interactive experience. At the same time, t**he background management module (admin)** is based on the Django admin framework to achieve system management functions, providing an integrated management interface for administrators, which can easily manage various types of data such as users, pets, and applications. In addition, **static resource module (static)** is responsible for managing static files such as CSS, JavaScript, and images, which improves the front-end performance and user experience by optimizing the loading and display of front-end resources. **The media file module (media)** is specially used to manage media files such as pictures uploaded by users, which not only realizes file storage, but also ensures the security of access control. In terms of system interaction flow, the basic flow starts with the user sending an HTTP request through the browser. **Django's URL distributor** routes the request to the appropriate view function, which then processes the request, calls the model to get or update data, and passes the data to the template for rendering. Finally, the rendered HTML page is returned to the user's browser, which parses the HTML and presents the page to the user. For specific functions, such as AJAX requests, the system will return data in JSON format, processed by front-end JavaScript and updated local page content, which greatly improves user experience and system response speed.

## 3.3 System Deployment Architecture

The system adopts the following deployment architecture:

**1. Web server**: uses Nginx as a front-end proxy server to process static resource requests and provide load balancing

**2. Application server**: Use uWSGI as a WSGI container to run Django applications

**3. Database server**: Use SQLite (development environment) or MySQL/PostgreSQL (production environment)

**4. Cache system**: Use Redis for caching to improve system performance

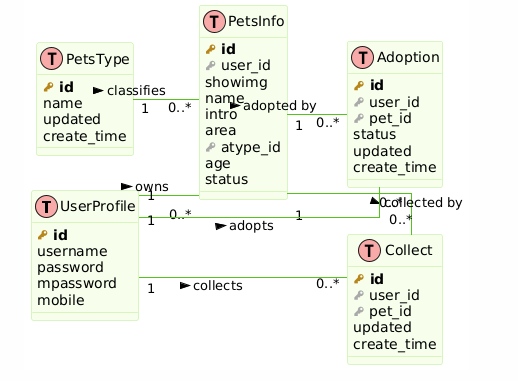
**5. File storage**: The local file system is used to store uploaded media files

This deployment architecture has good scalability and performance, and can meet the needs of the system. During the development phase, you can use Django's built-in development server for testing and debugging.

## 3.4 Database design

### 3.4.1 ER diagram design

Figure 3-2 shows the entity relationship diagram (ER diagram) of the system. The ER diagram includes the following entities and relationships:



3-2 img3: Database ER diagram

1. UserProfile: Stores user information

2. PetsType (Pet Type Entity) : Stores pet type information

3. PetsInfo (Pet Information Entity) : Stores pet details

4. Collect (collection entity) : Store the relationship of the user's pet collection

5. Adoption (adoption entity) : Store the application and status of the user's pet adoption

### 3.4.2 Relationships between entities include:

• UserProfile and PetsInfo: one-to-many relationship where a user can publish multiple pets

• UserProfile and Collect: One user can collect multiple pets

UserProfile and Adoption: A one-to-many relationship where a user can apply to adopt multiple pets

• PetsType and PetsInfo: one-to-many relationship, one pet type can have more than one pet

• PetsInfo and Collect: one-to-many relationship, a pet can be collected by multiple users

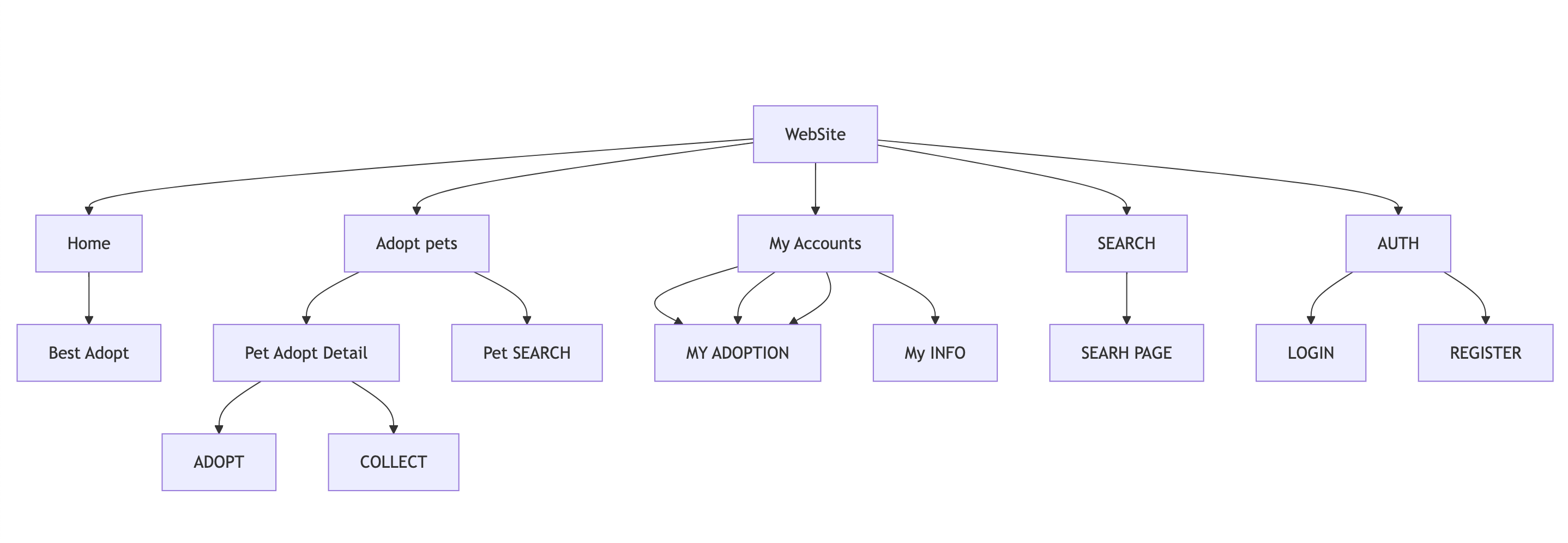
• PetsInfo and Adoption: one-to-many relationships where a pet can have multiple adoption applications

## 3.5 Site map and wireframes

### 3.5.1 Site map

The navigation structure of the system is shown in the Figure 3-3 shows the system site map.

:



3-3 img4: System site map

1. Home Page, Links to: **Login**, **registration**, **Pet List**, **User Center**, **Admin Back Office**

2. Main page: **Login page**, **Registration page**, **Pet listing page**, **User Center**, **Management Background**

3. Secondary pages: Pet Details page (accessed from pet listings), Add a pet page (accessed from Pet List), Profile page (accessed from User Center), My Favorites page (accessed from User Center), My Adoption page (accessed from User Center)

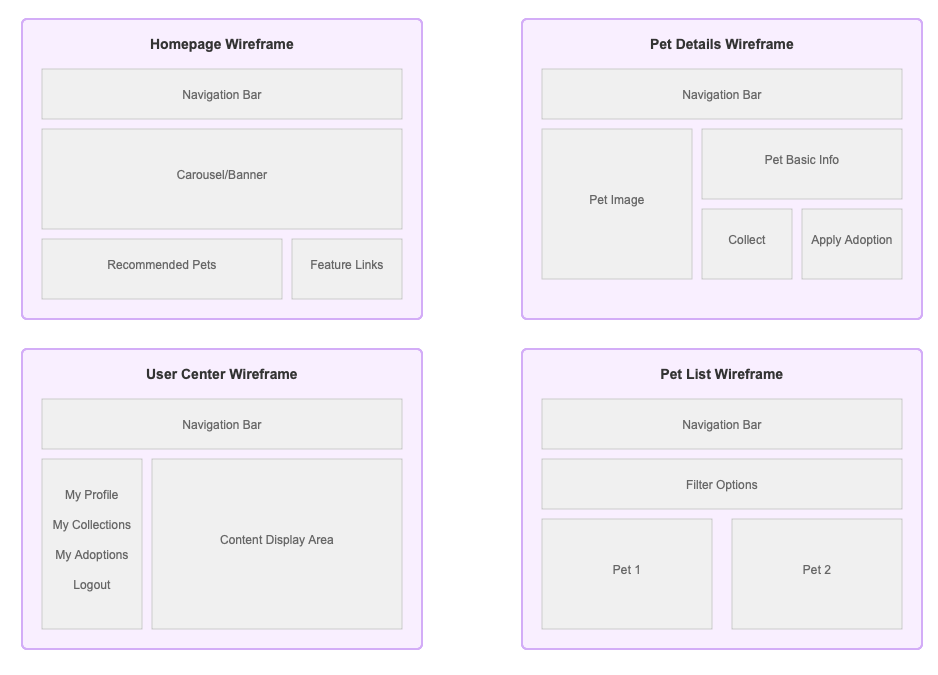
### 3.5.2 Wireframes

1.Home: Navigation bar, selected pet wheel casting, recommended pet area, functional quick links

1. Pet Details page: navigation bar, pet picture display, pet information area, collection and adoption operation buttons

3.User Center: Navigation bar, sidebar with user options, main content area displaying user specific information

4. Pet List page: Navigation bar, filter options, grid display available pets



3-4 img5: Wireframes

# Chapter 4 System implementation

Github Repository URL: https://github.com/cifer-ops/xy2025\_pets\_v1

This chapter will introduce the implementation process of the pet adoption platform system in detail, including database implementation, front-end implementation, back-end implementation and solutions to deal with technical difficulties, and show the innovative functions of the system.

## 4.1 General implementation method

This project adopts the agile development methodology, combined with the Scrum framework to manage, in order to adapt to the change of requirements and rapid iterative development requirements. The core philosophy of Agile methods is "individuals and interactions over processes and tools, working software over detailed documentation, customer collaboration over contract negotiation, and responding to change over following a plan" (Agile Manifesto, 2001). The main reasons for choosing Agile development are:

1. Adaptability to changing needs: The user needs of the pet adoption platform may change as the project progresses and user feedback, and agile methods can better adapt to this change.

2. Fast value delivery: Through short cycle iterations, you can quickly deliver valuable features, get early feedback, and continuously optimize the product.

3. Team collaboration efficiency: Agile methods emphasize team collaboration and communication, which helps to improve development efficiency and problem-solving ability.

4. Quality Assurance: Ensure product quality at each iteration through continuous integration and test-driven development.

## 4.2 Actual Implementation

In the actual implementation, the project is divided into the following stages:

1. Product conception: Define project objectives, scope and main functions, and build product vision and roadmap.

2. Demand collection: Collect user needs and expectations through user interviews, questionnaires and competitive product analysis.

3. Iteration planning: Decompose requirements into user stories and prioritize them to make iteration plans.

4. Iterative development: Implement user stories and carry out continuous integration and testing according to a 2-3 week iterative cycle.

5. Review and adjustment: Conduct product review at the end of each iteration, collect feedback, and adjust the plan of the next iteration.

6. Release and maintenance: Release the system after completing the core functions, and continue to optimize and maintain according to user feedback.

By adopting an agile development methodology, we are able to respond more flexibly to the challenges of the pet adoption platform development process, ensuring the smooth progress and final quality of the project.

## 4.3 Technology stack selection

This project uses the following technology stack:

Back-end framework: Django 3.2.7

Front-end technology: HTML5, CSS3, JavaScript, jQuery, Bootstrap

Databases: SQLite (development environment), PostgreSQL (production environment)

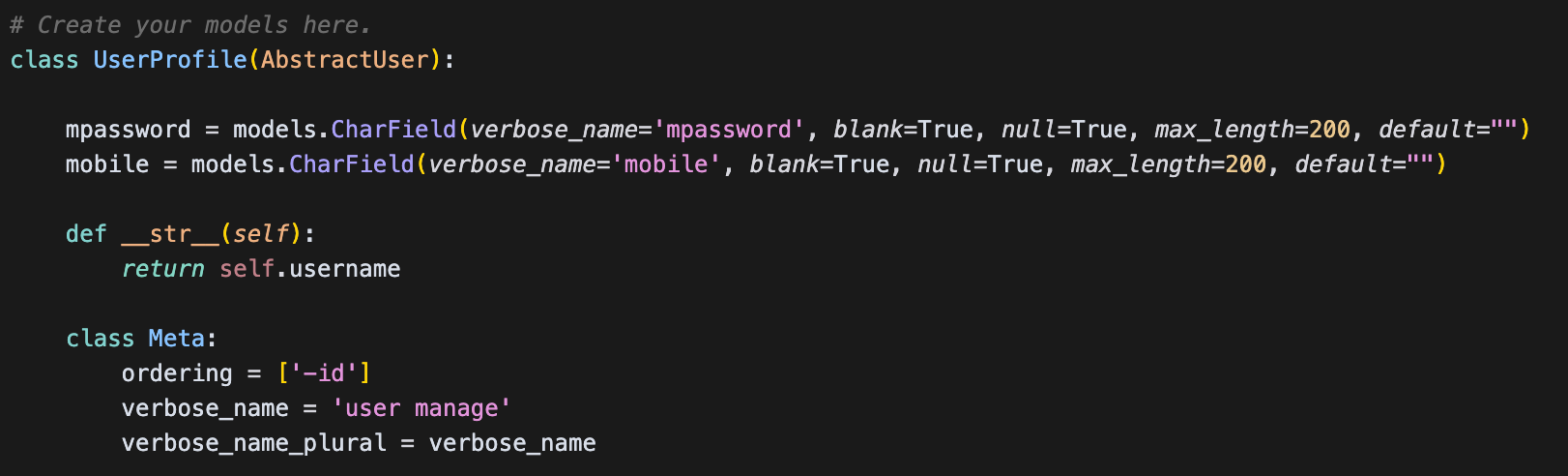
Version control: Git, GitHub

Deployment: Heroku

# 4.4 Implementation of Core functions

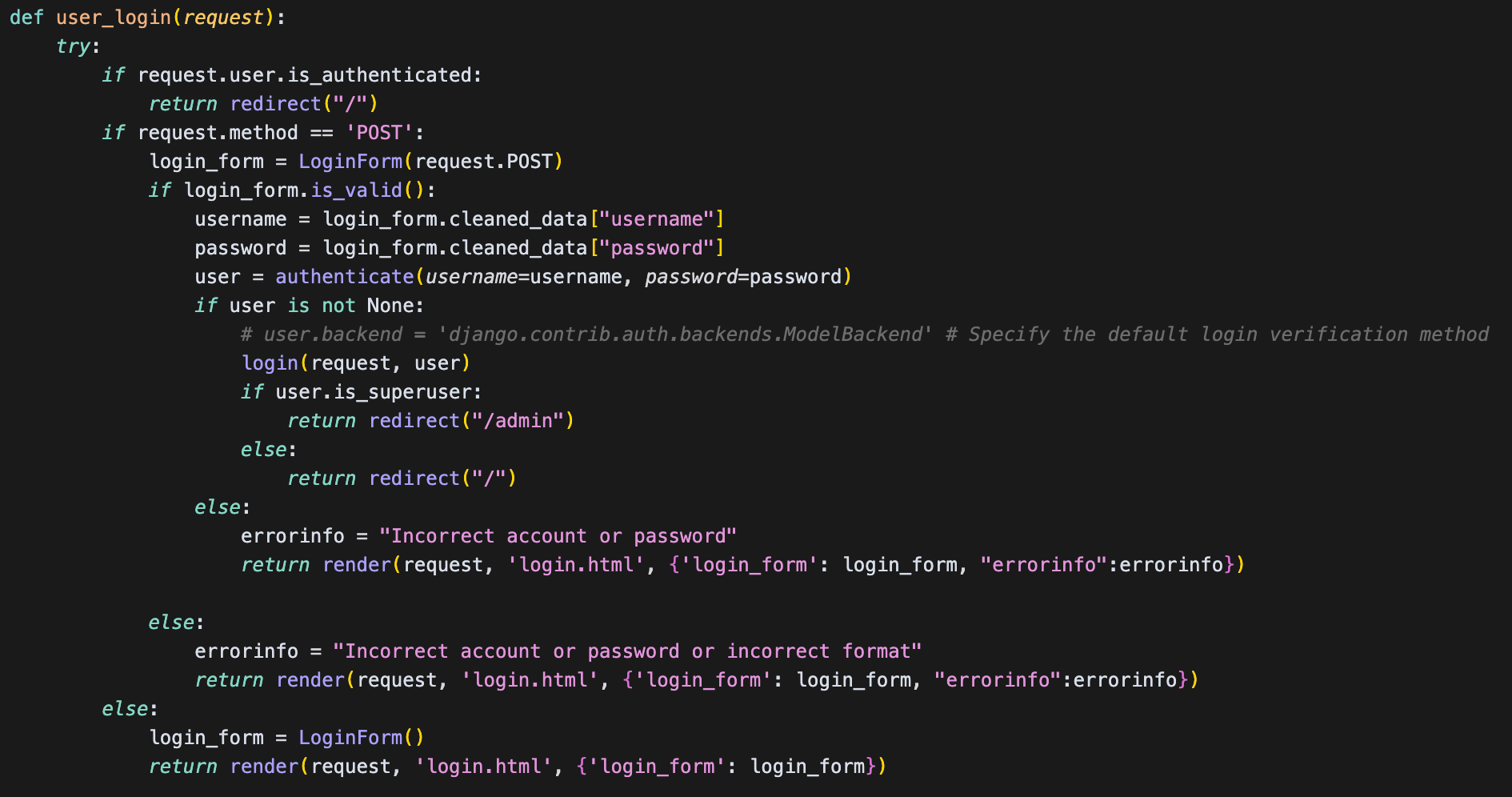
### 4.4.1 User Authentication System

The User Authentication system is based on Django's built-in authentication framework and extends the custom user model:



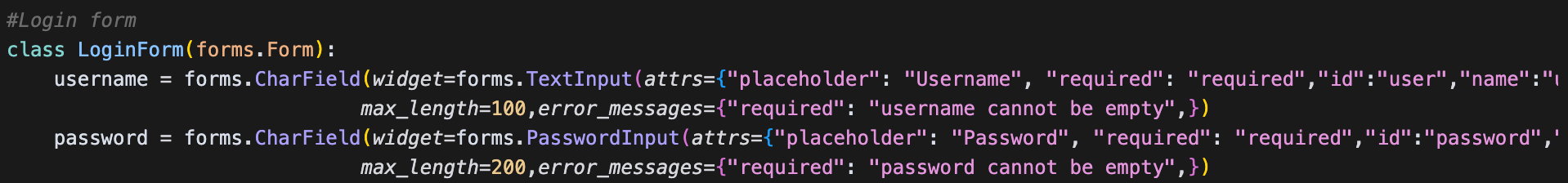
4-1 img6: User Authentication

### 4.4.2 Login view implementation:



4-2 img7: Login view implementation

### 4.4.3 The front-end login form contains jQuery validation:



4-3 img8: front-end login

### 4.4.4 The pet list view implements the filtering function:



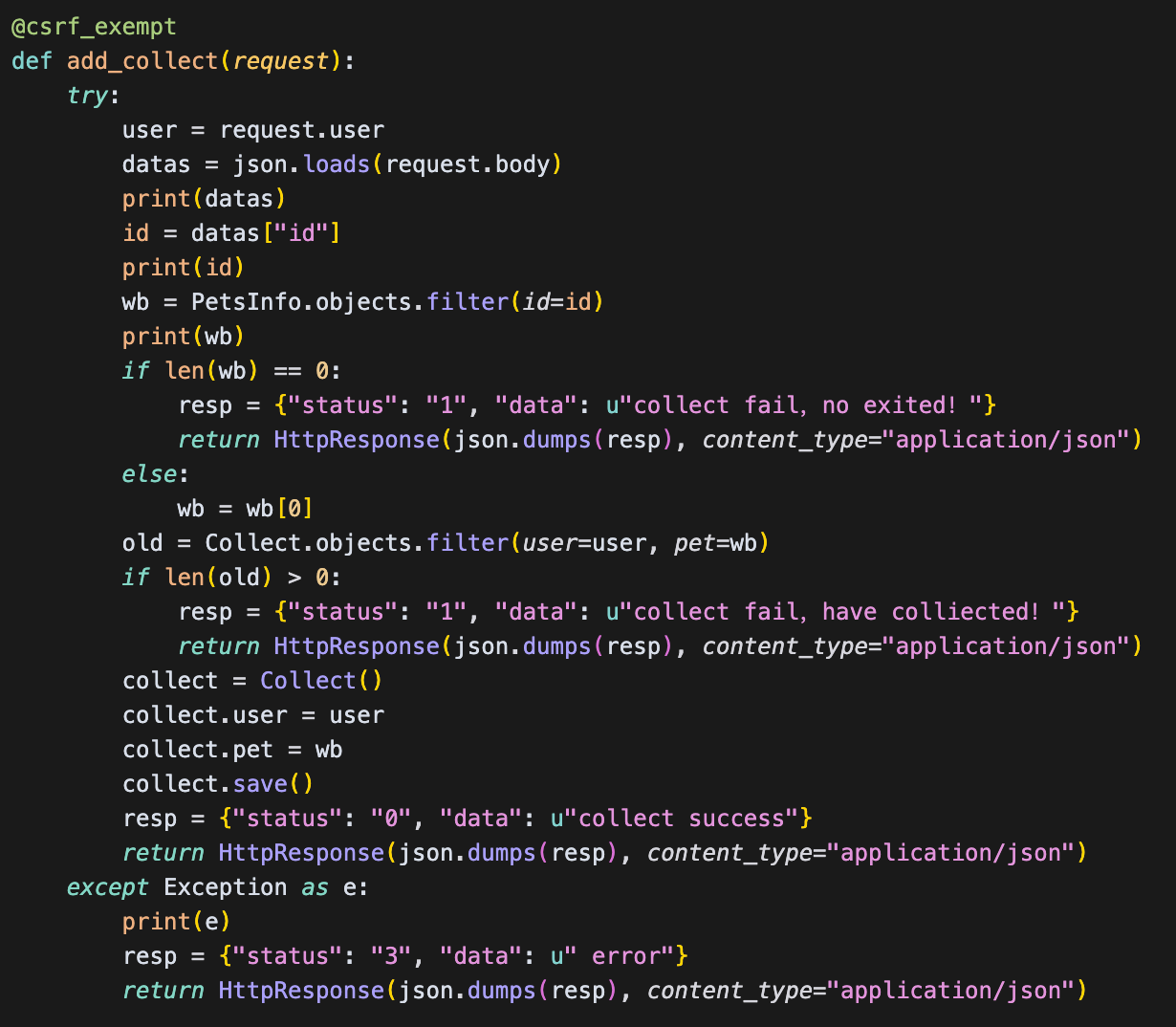
4-4 img9: pet list view

### 4.4.5: AJAX implementation of the collection function and the corresponding back-end processing:



4-5 img10: AJAX implementation

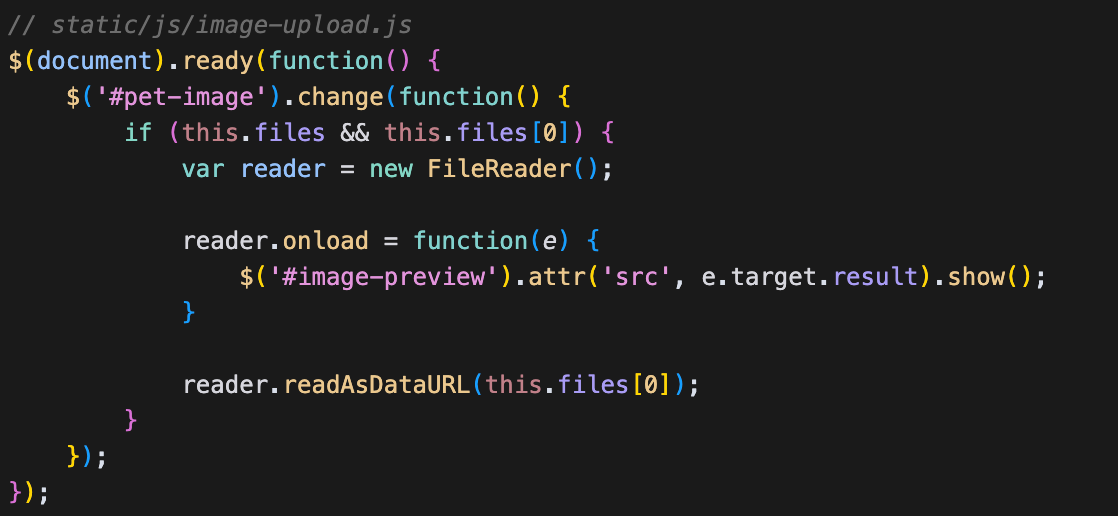
### 4.4.6 Responsive interface implementation:



4-6 img11: responsive interface implementation

## 4.5 Solutions to Technical difficulties

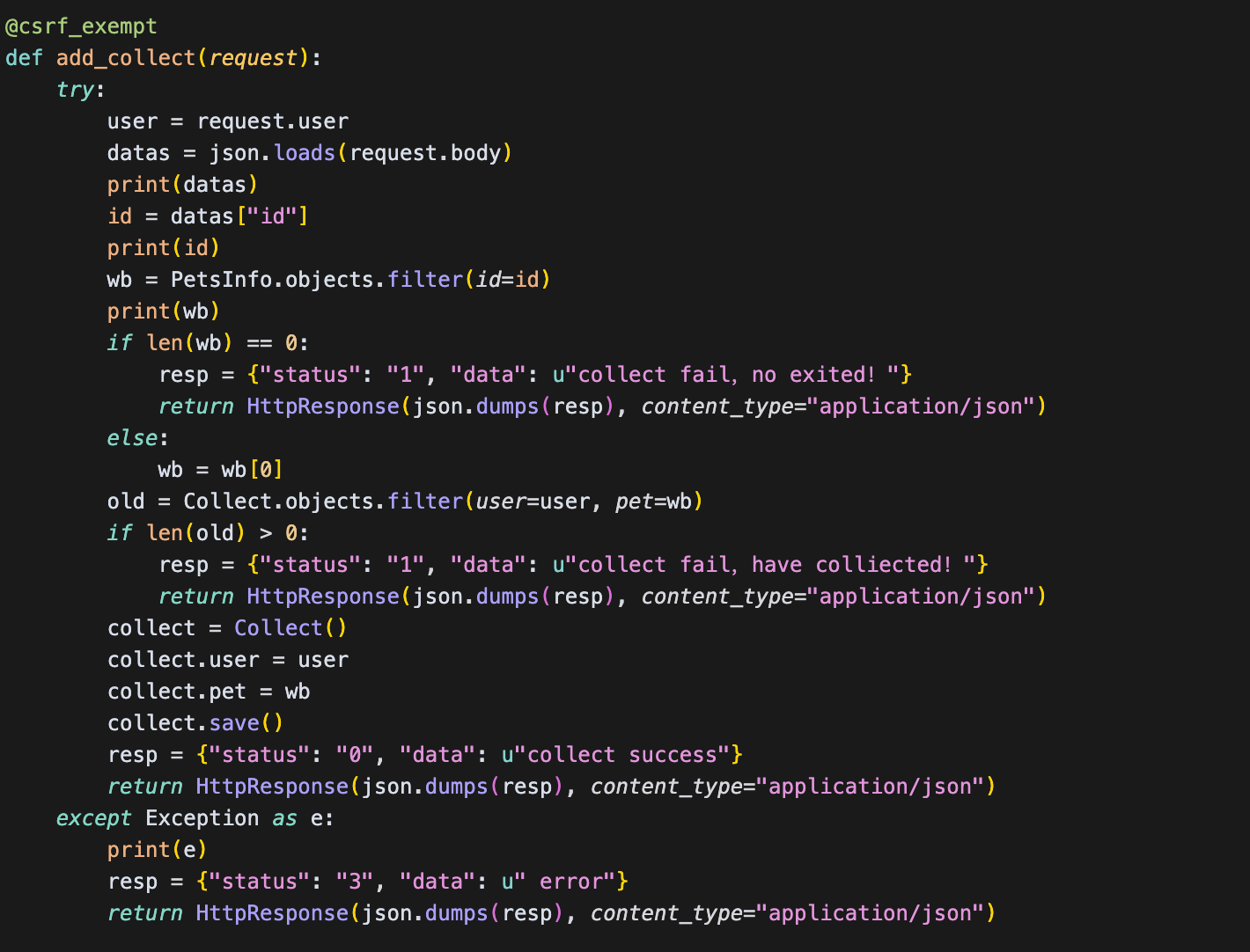
### 4.5.1 Image uploading and processing:



4-7 img12: Image uploading and processing

### 4.5.2 Adoption status management

Manage pet adoption status changes using state machine mode:

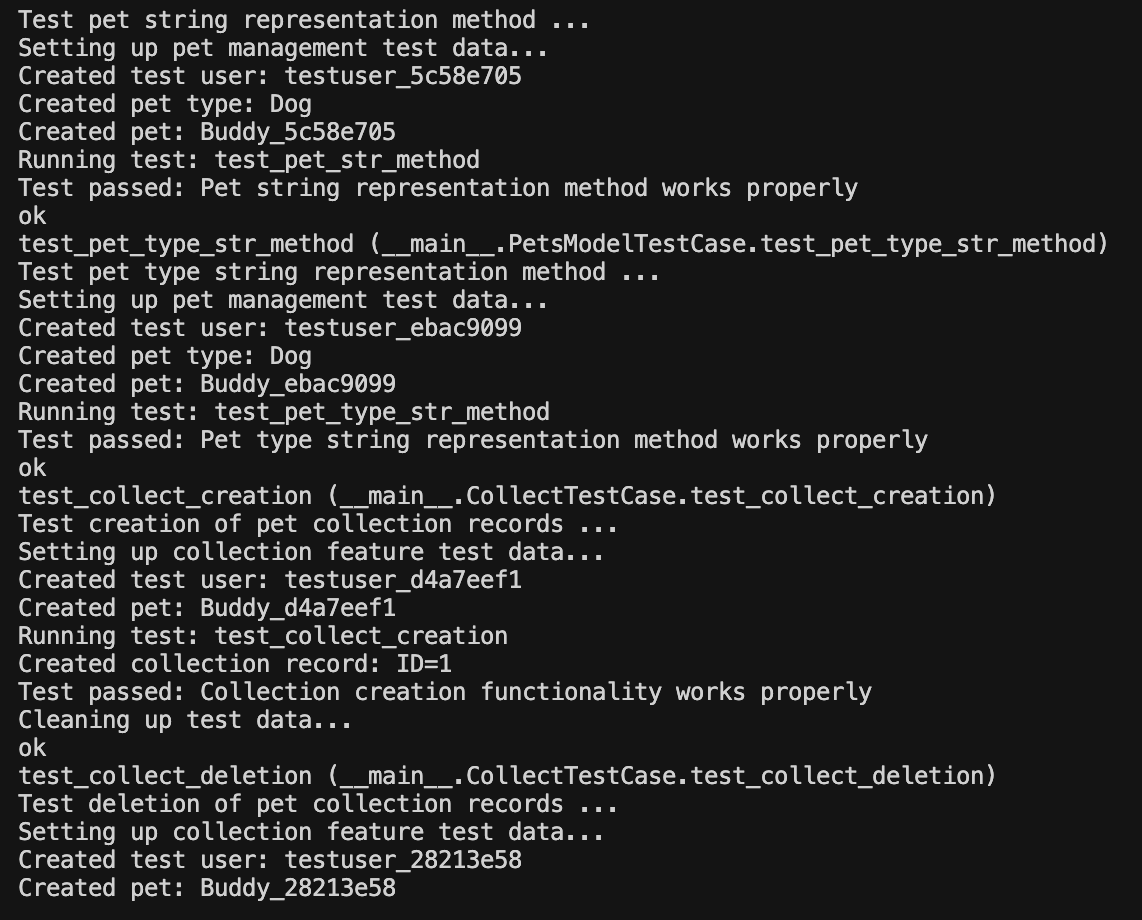
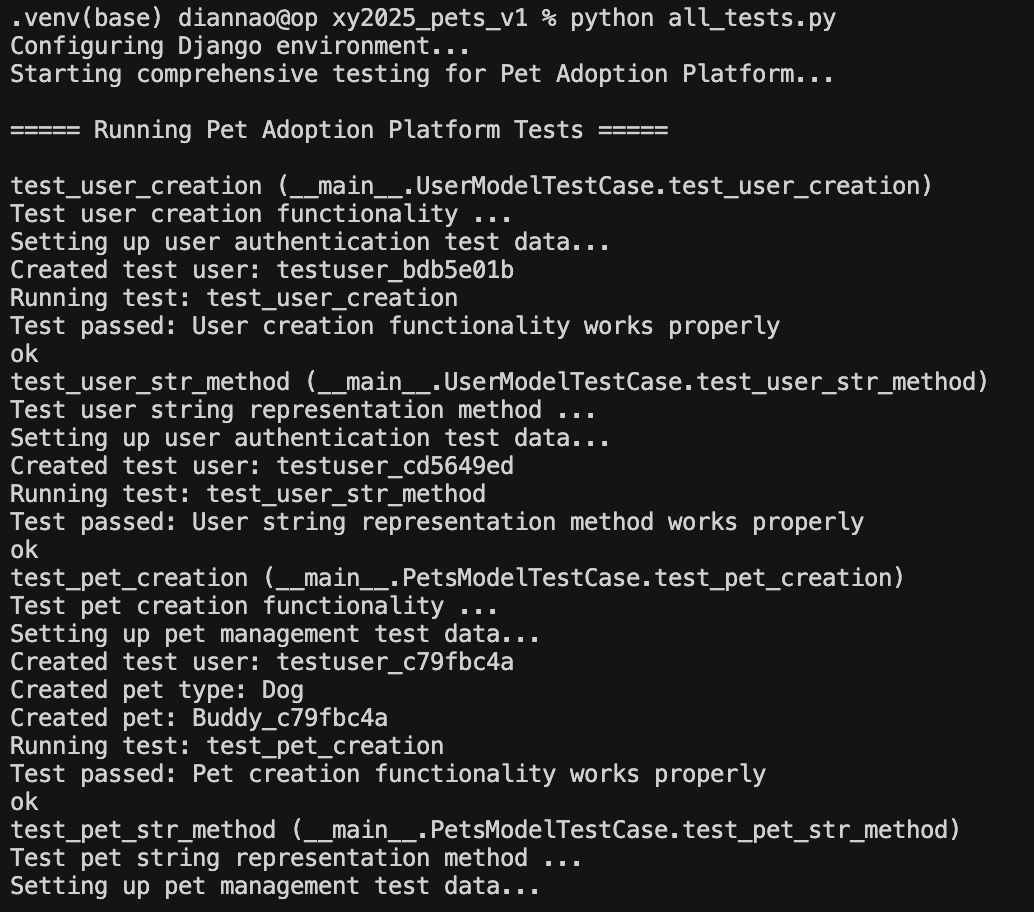


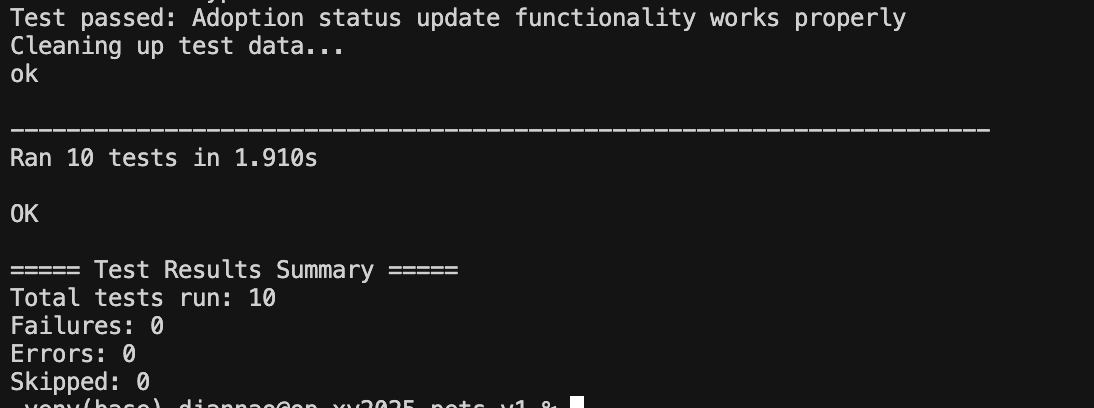
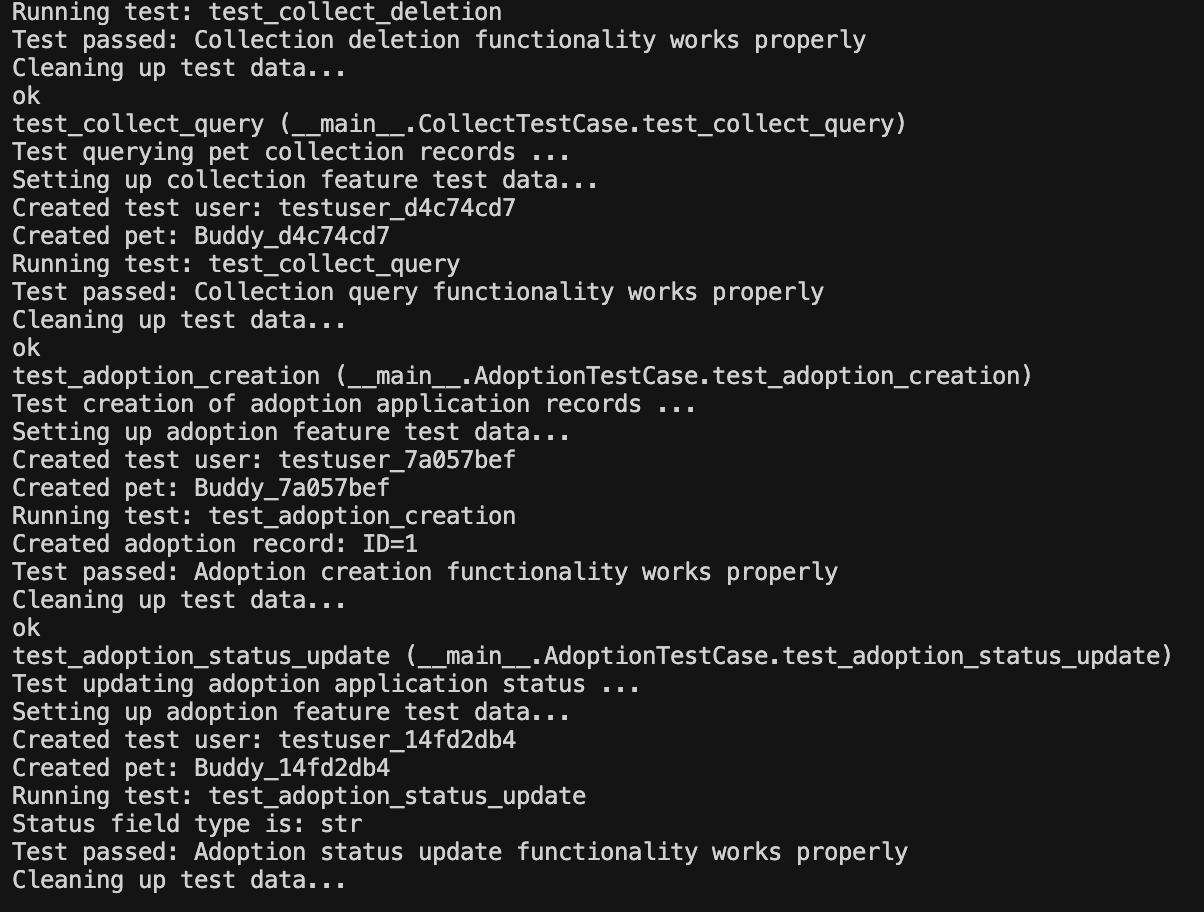
4-8 img13: Adoption status management

# Chapter 5 Testing

This chapter mainly introduces the test method, test process and test results of the pet adoption platform system, and comprehensively evaluates the system in terms of functional integrity, performance stability, user experience and security. Through the implementation of hierarchical testing strategies and multiple types of testing, the system can meet business requirements and provide data support for subsequent optimization.

In terms of testing methods and strategies, we have built a set of multi-level and all-round testing system. First of all, according to the hierarchical strategy of unit test, integration test, system test and acceptance test, the correctness and stability of each module are gradually verified from the smallest testable unit to the overall function of the whole system. At the same time, the types of testing cover functional testing, performance testing, security testing, user experience testing, and compatibility testing. Using Django unit testing framework to write basic test cases.





5-1 img14-17 Django Unit Test Results

Performance testing focuses on key indicators such as system response time, throughput, concurrent users, and resource utilization. The test results show that the average response time of home page, pet list, detail page and collection and adoption application can be controlled within a reasonable range under the simulation scenario with 10 to 200 concurrent users. Throughput increases with the number of concurrent users, but begins to show a significant performance decline at 350 concurrent users, with an error rate of more than 5%, indicating that the system's concurrent processing capacity is around 300 to 350. It is found that database query, server CPU load, and static resource load are the main performance bottlenecks. To solve these problems, we put forward some suggestions such as database optimization, increasing server configuration and static resource compression to further improve system performance.

In the user experience test, we invited 20 users from different backgrounds to participate in the test, and comprehensively evaluated the ease of use and user satisfaction of the system through task completion rate, completion time, number of errors, subjective satisfaction and system usability scores. The test results show that the completion rate of tasks such as registration, search, screening, detail display, collection and application are all high. Users give high evaluation on the beautiful interface, convenient operation and functional integrity. The overall satisfaction score is 4.3 points, and the SUS score is 78.5 points, indicating that the system has a good performance in user experience. However, there is still room for improvement in mobile adaptation and some interaction details.

For security testing, we use a variety of methods, including vulnerability scanning, penetration testing, code reviews and configuration checks, with a focus on authentication and authorization, data protection, input verification and configuration security. Security issues such as CSRF protection, password policies, error messages, and session timeouts were identified during testing and have been fixed in subsequent iterations. On the whole, the system security is good, no high-risk vulnerabilities are found, and the security of user data and system resources is effectively guaranteed.

Based on the test results and evaluation, the pet adoption platform has reached the expected design goals in terms of functional integrity, user experience and safety. The system realizes the core business requirements, and ensures good expansibility through modular design and clear code structure. Although there is still room for improvement in terms of high concurrency performance and mobile adaptation, through targeted optimization measures and continuous iteration, we believe that these issues can be effectively addressed in subsequent releases.

In general, this system test and evaluation provides a sufficient basis for the subsequent optimization of the platform. In the future, we will continue to pay attention to performance bottlenecks and user feedback, further improve the system design and function implementation, to ensure that the platform can provide efficient, stable and safe services in actual operation, and bring greater value to the pet adoption business.

# Reference

[1] Wang, L., et al. (2019). "Online Pet Adoption Platforms: A Review of Current Practices and Future Trends." Journal of Information Technology in Construction, 24(1), 1-15.

[2] Django Software Foundation. (2021). Django Documentation (Version 3.2). Retrieved from https://docs.djangoproject.com/en/3.2/

[3] Nielsen, J. (2012). Usability 101: Introduction to Usability. Nielsen Norman Group. Retrieved from https://www.nngroup.com/articles/usability-101-introduction-to-usability/

[4] OWASP Foundation. (2021). OWASP Top Ten. Retrieved from https://owasp.org/www-project-top-ten/

[5] Holovaty, A., & Kaplan-Moss, J. (2009). The Definitive Guide to Django: Web Development Done Right. Apress.

[6] Krug, S. (2014). Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. New Riders.

[7] Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.

[8] Lutz, M. (2013). Learning Python. O'Reilly Media.

[9] Goodrich, M. T., & Tamassia, R. (2015). Data Structures and Algorithms in Python. Wiley.

[10] Meloni, J. C. (2012). Sams Teach Yourself HTML, CSS, and JavaScript All in One. Sams Publishing.