Technical Documentation

1. Architecture:

The application follows a standard three-tier architecture:

- **Frontend (View):**
    - Implemented using Thymeleaf, HTML, CSS, and Bootstrap.
    - Utilizes JavaScript for client-side validation.
    - Provides a responsive and user-friendly interface.
- **Backend (Controller and Service):**
    - Built with Spring Boot, providing RESTful endpoints.
    - Utilizes Thymeleaf templates for server-side rendering.
    - Employs Spring Security for authentication and authorization.
    - Integrates Spring Validation for server-side input validation.
- **Data Layer (Model):**
    - Utilizes JPA (Java Persistence API) for data persistence.
    - Employs a relational database to store information about pets, pet articles, users, and other relevant entities.

2. Libraries and Frameworks:

- **Spring Boot:**
    - Used as the core framework for building the application.
    - Simplifies configuration and development of Spring applications.
- **Thymeleaf:**
    - Used for server-side templating, seamlessly integrated with Spring Boot.
- **Spring Security:**
    - Ensures secure authentication and authorization.
- **Bootstrap:**
    - Provides a responsive and visually appealing design.

3. Authentication and Authorization:

- **Spring Security:**
    - Configured for user authentication and authorization.
    - Admins have additional privileges for CRUD operations.

4. Validation:

- **Client-side Validation (JavaScript):**
    - Implemented for enhanced user experience during form interactions.
- **Server-side Validation (Spring Validation):**
    - Ensures data integrity and correctness on the server.

5. CRUD Operations:

- **Pets and Pet Articles:**
    - Create, Read, Update, and Delete operations are available.
    - Admins have exclusive access to these operations.

6. Data Storage:

- **Relational Database:**
    - Employs a relational database (e.g., MySQL, PostgreSQL) for data storage.
    - Entities include Pet, PetArticle, User, etc.

7. Testing:

- **Postman:**
    - Used for testing CRUD operations on the RESTful endpoints.

8. Security:

- **HTTPS:**
    - Enforces secure communication between the client and the server.

10. Build Tool:

- **Maven:**
    - Used for project management and build automation.

11. Development Environment:

- **IDE:**
    - Developed using IntelliJ IDEA