**Pet Adoption Website Project – Full Documentation**

**1. Project Overview**

A modern, full-stack **Pet Adoption Website** and e-commerce platform that allows users to browse, adopt pets, and purchase pet-related products. It includes customer and admin functionalities, secured authentication, and real-time database integration.

**2. Project Structure**

text

pet-adoption-website/

├── frontend/

│ ├── index.html # Homepage

│ ├── adoption.html # Pet adoption page (lists available pets)

│ ├── shop.html # Product shop page

│ ├── customer.html # Customer dashboard/account page

│ ├── admin.html # Admin dashboard (secured)

│ ├── admin-products.html # Admin product management

│ ├── customer-login.html # Customer login page

│ ├── admin-login.html # Admin login page

│ ├── register.html # Customer registration page

│ ├── css/ # Styling files (responsive and design)

│ ├── js/ # Frontend JavaScript (API calls, interactivity)

│

├── backend/

│ ├── RestApiServer.java # Java HTTP server providing REST API

│ ├── Main.java # Backend business logic and DB operations

│ ├── lib/ # Required libraries (gson, mysql connector)

│

├── database/

│ ├── schema.sql # SQL for creating all database tables

│

└── docs/

└── setup\_instructions.md # This documentation and setup guide

**3. Technologies Used**

* **Frontend**: HTML5, CSS3 (responsive design), JavaScript (fetch API, DOM manipulation)
* **Backend**: Java (HttpServer), Gson (JSON parsing), JDBC + MySQL Connector for DB access
* **Database**: MySQL, relational schema with tables for pets, products, customers, orders, adoption logs
* **Tools**: Java 17+, MySQL 8+, Postman (API testing), Git (version control)

**4. Database Design**

**Key tables:**

* customer – Stores customer profile information.
* customer\_login – Stores authentication info (email, hashed password) linked to customer.
* pet – Stores pet details available for adoption.
* product – Stores product info in the shop.
* orders – Stores order transactions (purchase history).
* adopts – Records pet adoption events.

Relationships:

* One-to-many between customers and orders.
* One-to-many between customers and adopts.
* Foreign keys ensure data integrity.

**5. Backend Development**

* Set up **Java HTTP server** (RestApiServer) that listens on port 8080.
* Created RESTful endpoints for:
  + **Customer registration**: accepts JSON, inserts into customer and customer\_login tables with hashed password.
  + **Pet and product retrieval**: retrieve lists from database.
  + **Product addition, pet addition, and purchase APIs**.
* Implemented **CORS filter** to enable frontend communication.
* Leveraged **Gson** for JSON serialization/deserialization.
* Managed **database connections and transactions** with JDBC.
* Ensured **secure password storage** using SHA-256 hashing (recommend bcrypt in production).

**6. Frontend Development**

* Developed responsive, user-friendly pages:
  + **Login pages** for customers and admins, with validation and session storage.
  + **Registration page** for new users, form validation, and submission to backend.
  + **Customer dashboard** showing profile, adopted pets, orders, and wishlist.
  + **Admin dashboard** with product and pet management.
  + **Adoption page** that fetches and displays pets dynamically with filtering and adoption actions.
  + **Shop page** displaying products fetched from backend, with shopping cart functionality.
* Used JavaScript to:
  + Call backend APIs with fetch.
  + Update UI reactively on data changes.
  + Perform client-side validation.
  + Manage user session state via localStorage/sessionStorage.

**7. Security Considerations**

* Passwords are hashed before storage.
* CORS headers configured to allow frontend access.
* Login sessions managed on the frontend with local/session storage.
* Admin and customer pages include JavaScript checks to ensure login status.
* Backend designed for easy extension to implement more robust token-based auth (e.g., JWT).

**8. Testing and Deployment**

* Tested API endpoints via Postman and browser.
* Debug logs included in the backend for troubleshooting.
* Verified database operations with live data insertion and retrieval.
* Frontend tested on desktop and mobile views for responsiveness.
* Instructions for compiling and running backend server and accessing frontend pages provided.

**9. Future Enhancements**

* Implement token-based authentication (JWT) for secure backend API calls.
* Add password reset and email verification flows.
* Expand admin reporting tools (sales, adoption trends).
* Integrate payment gateway for purchase processing.
* Improve UI with React or Vue.js for better scalability.
* Add real-time notifications and chat support.

**10. How to Run the Project**

**Prerequisites**

* Java 17+ installed
* MySQL server running, database schema created
* Required JARs (gson, mysql-connector) placed in backend/lib

**Steps**

1. Compile backend:

text

javac -cp ".;lib/gson-2.9.0.jar;lib/mysql-connector-java-8.0.33.jar" Main.java RestApiServer.java

1. Run backend:

text

java -cp ".;lib/gson-2.9.0.jar;lib/mysql-connector-java-8.0.33.jar" RestApiServer

1. Open frontend pages in browser (e.g., index.html, customer-login.html).
2. Register and login as customer or admin.
3. Use admin panel for pet/product management.
4. Browse pets, adopt pets, buy products as customer.

**11. Summary**

This project successfully:

* Built a **full-stack pet adoption and product store website**.
* Designed a **structured relational database** with clear relationships.
* Developed **Java REST API backend** for data management.
* Created **modern, responsive frontend web pages**.
* Implemented **registration and login flows** with secure password handling.
* Enabled **real-time dynamic data loading and interaction**.
* Provided a foundation to **expand features and scale** the platform.