**PawsConnect Pet Adoption Website**

**Setup and Installation Instructions**

**Table of Contents**

1. [Overview](https://www.perplexity.ai/search/make-html-code-for-this-HKLpFaLSTD2myuZIW43yCw#overview)
2. [Prerequisites](https://www.perplexity.ai/search/make-html-code-for-this-HKLpFaLSTD2myuZIW43yCw#prerequisites)
3. [Database Setup](https://www.perplexity.ai/search/make-html-code-for-this-HKLpFaLSTD2myuZIW43yCw#database-setup)
4. [Backend Server Setup](https://www.perplexity.ai/search/make-html-code-for-this-HKLpFaLSTD2myuZIW43yCw#backend-server-setup)
5. [Frontend Setup](https://www.perplexity.ai/search/make-html-code-for-this-HKLpFaLSTD2myuZIW43yCw#frontend-setup)
6. [Running the Application](https://www.perplexity.ai/search/make-html-code-for-this-HKLpFaLSTD2myuZIW43yCw#running-the-application)
7. [Common Issues and Troubleshooting](https://www.perplexity.ai/search/make-html-code-for-this-HKLpFaLSTD2myuZIW43yCw#common-issues-and-troubleshooting)
8. [Further Development](https://www.perplexity.ai/search/make-html-code-for-this-HKLpFaLSTD2myuZIW43yCw#further-development)

**Overview**

PawsConnect is a full-stack pet adoption website, featuring functionalities such as user registration and login, pet adoption, product shop, order management, and an admin dashboard. The backend is done in Java with an HTTP server and MySQL database. The frontend consists of responsive HTML, CSS, and JavaScript pages.

**Prerequisites**

Before setup, ensure you have the following installed:

* **Java Development Kit (JDK) 11 or higher**  
  [Download JDK](https://adoptium.net/)
* **Apache Maven** (optional, if you use build tools)
* **MySQL Server 8.x** or higher
* **MySQL Workbench or CLI client** for DB management
* **Web browser** for frontend testing (Chrome, Firefox, Edge)

**Libraries:**

* Gson (for JSON handling) — gson-2.8.9.jar
* MySQL JDBC Connector — mysql-connector-java-8.0.x.jar

*Ensure the lib/ folder contains these JAR files.*

**Database Setup**

1. **Start MySQL Server**  
   Make sure your MySQL server is running.
2. **Create the Project Database**  
   Open MySQL Workbench or CLI and run:

sql

**CREATE** **DATABASE** miniproject;

**USE** miniproject;

1. **Create Required Tables**  
   Execute the SQL schema below to create tables:

sql

*-- Customer Table*

**CREATE** **TABLE** customer (

customer\_id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

customer\_name **VARCHAR**(100) NOT NULL,

address **TEXT**,

email\_address **VARCHAR**(100),

mobile\_number **VARCHAR**(20)

);

*-- Customer Login*

**CREATE** **TABLE** customer\_login (

login\_id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

customer\_id **INT** NOT NULL,

email **VARCHAR**(100) NOT NULL **UNIQUE**,

password\_hash **VARCHAR**(255) NOT NULL,

is\_active **BOOLEAN** **DEFAULT** TRUE,

last\_login **DATETIME**,

**FOREIGN** **KEY** (customer\_id) **REFERENCES** customer(customer\_id)

);

*-- Admin Login*

**CREATE** **TABLE** admin\_login (

admin\_id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

username **VARCHAR**(50) NOT NULL **UNIQUE**,

password\_hash **VARCHAR**(255) NOT NULL,

admin\_name **VARCHAR**(100) NOT NULL,

email **VARCHAR**(100),

is\_active **BOOLEAN** **DEFAULT** TRUE,

created\_date **DATETIME** **DEFAULT** **CURRENT\_TIMESTAMP**,

last\_login **DATETIME**

);

*-- Pet Table*

**CREATE** **TABLE** pet (

pet\_id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

pet\_name **VARCHAR**(50) NOT NULL,

pet\_type **VARCHAR**(30) NOT NULL,

breed **VARCHAR**(50),

age **VARCHAR**(20),

date\_of\_rescue **DATE**,

is\_adopted **BOOLEAN** **DEFAULT** FALSE

);

*-- Product Table*

**CREATE** **TABLE** product (

product\_id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

product\_name **VARCHAR**(100) NOT NULL,

**type** **VARCHAR**(50) NOT NULL,

used\_by\_pet **VARCHAR**(30),

cost **DECIMAL**(10,2) NOT NULL,

quantity **INT** NOT NULL **DEFAULT** 0

);

*-- Order Table*

**CREATE** **TABLE** orders (

order\_id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

customer\_id **INT** NOT NULL,

product\_id **INT** NOT NULL,

quantity **INT** NOT NULL,

order\_date **DATE** NOT NULL,

total\_amount **DECIMAL**(10,2),

**status** **VARCHAR**(20) **DEFAULT** 'pending',

**FOREIGN** **KEY** (customer\_id) **REFERENCES** customer(customer\_id),

**FOREIGN** **KEY** (product\_id) **REFERENCES** product(product\_id)

);

*-- Adopt Table*

**CREATE** **TABLE** adopts (

adoption\_id **INT** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

customer\_id **INT** NOT NULL,

pet\_id **INT** NOT NULL,

adoption\_date **DATE** NOT NULL,

**FOREIGN** **KEY** (customer\_id) **REFERENCES** customer(customer\_id),

**FOREIGN** **KEY** (pet\_id) **REFERENCES** pet(pet\_id)

);

1. **Add Default Admin User**

sql

**INSERT** **INTO** admin\_login (username, password\_hash, admin\_name, email, is\_active)

**VALUES** ('admin', '48bb9e7f2f8e7fc18cd536a30c0aeb6a9d096f3e54398c5b6f8652bc5642ae1c', 'System Administrator', 'admin@pawsconnect.com', TRUE);

*The password is admin123 hashed with SHA-256.*

**Backend Server Setup**

1. **Configure Database Credentials**  
   Open RestApiServer.java and update the following variables with your MySQL credentials:

java

**private** **static** **final** String DB\_URL = "jdbc:mysql://localhost:3306/miniproject";

**private** **static** **final** String DB\_USER = "root";

**private** **static** **final** String DB\_PASSWORD = "your\_mysql\_password";

1. **Place Required JARs**  
   Place gson-2.8.9.jar and mysql-connector-java-8.0.x.jar inside a lib/ folder.
2. **Compile the Java Code**  
   Open terminal inside your backend folder and run:

bash

javac -cp ".;lib/gson-2.8.9.jar;lib/mysql-connector-java-8.0.x.jar" Main.java RestApiServer.java

1. **Run the Server**

bash

java -cp ".;lib/gson-2.8.9.jar;lib/mysql-connector-java-8.0.x.jar" RestApiServer

1. **Server Output**  
   On success, you should see output indicating successful DB connection and server running on port 8080.

**Frontend Setup**

1. Open the frontend/ folder.
2. Deploy your website files (index.html, customer-login.html, register.html, etc.) on any web server or open them directly in your browser.
3. The frontend will interact with the backend via HTTP requests to http://localhost:8080.

**Running the Application**

1. **Start MySQL Server.**
2. **Run backend server** as described above.
3. **Open frontend pages** in a modern browser:
   * Homepage: index.html
   * Register: register.html
   * Login: customer-login.html or admin-login.html
   * Shop: shop.html
   * Adoption: adoption.html
4. **Register users** and test the full workflow:
   * Register a new customer
   * Login as customer/admin
   * Browse pets, adopt pets
   * Shop products and checkout (if implemented)
   * Use admin dashboard for management

**Common Issues & Troubleshooting**

* **"Cannot connect to database"**:
  + Check database credentials in RestApiServer.java.
  + Ensure MySQL server is running.
* **"Access denied for user"**:
  + Check MySQL user permissions.
* **"API requests fail (CORS)"**:
  + Ensure backend CORS headers are set.
  + Use the backend’s provided CORS filter.
* **Compilation errors with Date class**:
  + Ensure import java.sql.Date and use java.sql.Date.valueOf(...) for date conversion.
* **Port conflict on 8080**:
  + Stop other services using port 8080 or change port in RestApiServer.java.

**Further Development**

* Implement **secure password hashing** (consider bcrypt instead of SHA-256).
* Add **JWT or session management** for authentication.
* Integrate **payment gateway** in checkout.
* Expand **admin reporting and management** features.
* Enhance **frontend with React/Vue** for dynamic UI.

**Support & Contributions**

If you face any issues or have suggestions, feel free to reach out or contribute to the project!

Happy coding and thank you for making a difference for pets and people with PawsConnect! 🐾❤️