PetPaw

Pet Adoption Management System

College Project Assessment Document • February 2026

Project Title	PetPaw – Pet Adoption Management System
Technology Stack	HTML, CSS, Bootstrap 5, JS, Python (Flask), Supabase
Project Type	Web-Based Pet Adoption Portal
Last Updated	February 2026

1. Executive Summary

PetPaw is a comprehensive pet adoption management system designed to connect animal shelters with potential pet adopters through a seamless, modern web interface. It bridges the gap between shelters managing their listings and users seeking to adopt pets. The platform supports secure authentication with email confirmation, role-based interactions between adopters and shelter staff, a donation platform, real-time messaging, and a wishlist/favorites system.

2. Project Objectives

Primary Objectives

- User-friendly pet adoption browsing and application platform
- Secure authentication with email confirmation via Supabase Auth
- Shelter management with dedicated shelter profiles and pet listings
- Pet wishlist / favourites functionality for registered users
- Donation capabilities for general fund and shelter-specific giving

Secondary Objectives

- Real-time chat/messaging system between users and shelters
- Adoption application status tracking with visual indicators
- Fully responsive design using Bootstrap 5 for all devices
- Flask backend integrated with Supabase PostgreSQL
- Row-Level Security (RLS) enforced on all user-owned tables

3. System Architecture

Frontend Architecture

The frontend is a multi-page HTML/CSS/JavaScript application styled with Bootstrap 5.3 and Font Awesome 6. Pages include: *index.html* (home/browsing), *shelters.html*, *shelter-detail.html*, *pet-detail.html*, *adopt.html*, *donate.html*, *my-adoptions.html*, *my-favorites.html*, *contact-shelter.html*, *login.html*, and *register.html*. A shared *main.js* handles auth state, pet card rendering, and favorites logic.

Backend Architecture (Flask + Supabase)

The backend is a Python Flask REST API (*app.py*) with Flask-CORS. It exposes endpoints for pets, shelters, adoption requests, donations, favorites, messages, and authentication. All data is persisted in Supabase PostgreSQL with Supabase Auth handling email/password sign-up and JWT issuance.

REST API Endpoint	Method	Purpose
/api/pets	GET	Browse & filter pets
/api/shelters	GET	Shelter listings
/api/shelters/<id>	GET	Single shelter + pets
/api/adoption-request	POST	Submit application
/api/my-adoptions/<id>	GET	User's applications
/api/donations	POST	Record donation
/api/favorites	POST	Add to favorites
/api/favorites/<id>	DELETE	Remove favourite
/api/messages	POST	Send message
/api/messages/<u>/<s>	GET	Fetch conversation
/auth/register	POST	User registration
/auth/login	POST	Login + JWT

4. Features & Modules

Home / Pet Browsing
Grid of all available pets with photo, name, breed, and age. Heart button saves pets to favourites. Responsive layout — 3 columns desktop, 1 on mobile.

Shelter Management
Shelter listing page with contact info. Detail page shows all available pets at that shelter and a direct Contact button linking to messaging.

Pet Detail & Adoption
Full pet profile with images, description, and shelter info. Adoption application form with personal details; tracked in My Adoptions.

Donations
Donate to PetPaw General Fund or a specific shelter. Amount presets (■100–■5000) plus custom input. Three payment methods: UPI, Card, Net Banking (simulated).

My Favourites / Wishlist
Heart button instantly saves pets (login required). My Favourites page displays saved pets with animated cards and one-click removal.

My Adoptions Tracker
Displays all adoption applications. Color-coded status bars: Gold = Pending, Teal = Approved, Red = Rejected. Stats grid for each status.

Contact / Messaging
Chat-style interface to contact any shelter. Shelter selector with live info card. Messages as chat bubbles; Enter key support and auto-scroll.

Authentication
Registration with email confirmation via Supabase Auth. JWT stored in localStorage; cleared on logout. All personal pages protected with login-required guard.

5. Database Design

Table	Description	Access
profiles	User profile info synced from auth.users via DB trigger	User owns data
pets	Pet listings with breed, age, image, shelter link	Public read
shelters	Shelter info: name, address, phone, email, logo	Public read
adoption_requests	User adoption applications with status tracking	User owns data
donations	Donation records linked to donor and optional shelter	Auth users
favorites	User pet wishlist with unique user+pet constraint	User owns data
messages	Chat messages between users and shelters	Auth users
species / breeds	Animal species and breed catalogs	Public read

6. Security Implementation

6.1 Authentication Security

- Email/password authentication via Supabase Auth
- Email confirmation required before login is permitted
- Minimum 6-character password enforced by Supabase
- JWT tokens stored in localStorage; cleared on logout

6.2 Data Security

- Row Level Security (RLS) policies on all user-owned tables
- Profiles table references auth.users via auto-trigger
- Favorites & adoption_requests enforce ownership via user_id

6.3 Frontend Guards

- Personal pages check login status on load
- Unauthenticated users see a lock screen with login redirect
- Donate button requires login before submission
- Favourite toggle redirects to login if not authenticated

7. User Roles & Permissions

- Registered User: browse, apply, favourite, donate, chat
- Guest: browse pets & shelters only (read-only access)

8. Authentication Flow

Registration Flow

① User fills Register Form (name, email, password) → ② Supabase Auth creates auth.users entry → ③ DB trigger auto-creates profiles row → ④ Confirmation email sent to user → ⑤ User clicks link → Account activated → User can log in.

Login Flow

① User submits Login Form (email + password) → ② Supabase validates credentials → ③ On Success: JWT stored in localStorage, navbar updates to 'Welcome, [name]', redirect to home. On Failure: error message displayed to user.

9. Technical Specifications

9.1 Frontend Technologies

Technology	Version	Purpose
HTML5 / CSS3	Latest	Page structure, styling & animations
Bootstrap	5.3.3	Responsive layout and UI components
Font Awesome	6.5.1	Icons throughout the application
Google Fonts	Latest	Fraunces + Plus Jakarta Sans typography
JavaScript (ES6+)	Latest	Frontend interactivity and API calls

9.2 Backend Technologies

Technology	Purpose
Python (Flask)	REST API server handling all route logic
Flask-CORS	Cross-Origin Resource Sharing for frontend-backend comms
Supabase Python Client	Database queries and auth operations
Supabase / PostgreSQL	Backend-as-a-Service with Auth, relational DB

9.3 Development Tools

Git + GitHub	Version control & hosting
Supabase Dashboard	DB management, SQL editor, RLS
VS Code	Primary code editor
Python venv / pip	Dependency management

10. UI Highlights

Home / Pet Browsing	Hero section with gradient and floating emoji animations. Pet cards grid with heart button, shelter badge, and View Details link. Responsive: 3 columns desktop, 1 mobile.
Shelter Pages	Shelters list with teal/navy gradient hero. Detail page shows logo, contact info, and all available pets. Contact Shelter button links to messaging.
Donation Page	Purple gradient hero. Fund selection (General / Specific Shelter), amount preset grid, payment methods. Impact sidebar + recent donations list.
My Adoptions	Stats grid (Total / Pending / Approved / Rejected). Each card has a color-coded status bar. Pet photo, name, breed, applicant details, and status badge.
My Favourites	Deep red/coral hero. Saved pets with hover zoom effect. Remove button fades out card with smooth animation.
Contact Shelter	Dark navy hero with coral chat header. Left panel: shelter selector + info card. Right panel: chat bubbles (sent right, received left). Enter key + auto-scroll.

11. Future Enhancements

Phase 2

- Admin dashboard for shelter staff to manage pets and applications
- Push notifications when application status changes
- Pet search and filter by species, breed, age, and location
- Reviews and ratings system for shelters
- My Profile page for users to update personal information

Phase 3

- Mobile app (React Native) for iOS and Android
- Real payment gateway integration (Razorpay / Stripe)
- Email notifications for adoption approvals and shelter replies
- Advanced analytics dashboard for shelter managers
- Machine learning model to suggest pets based on user preferences

Appendix: Installation & Setup

Requirements: Python 3.9+, pip, a Supabase project with required tables.

```
# 1. Clone the repository
git clone https://github.com/hemajanu04/pet-adoption-system.git
cd pet-adoption-system

# 2. Install Python dependencies
pip install flask flask-cors supabase

# 3. Set environment variables
export SUPABASE_URL=your_supabase_url
export SUPABASE_KEY=your_supabase_key

# 4. Start Flask development server
python app.py

# 5. Open browser → http://127.0.0.1:5000/static/index.html
```

Variable	Description
SUPABASE_URL	Supabase project URL
SUPABASE_KEY	Supabase service role or anon key
SECRET_KEY	Flask secret key for session management

12. Conclusion

PetPaw represents a comprehensive full-stack solution for connecting animal lovers with shelters and pets in need of homes. The project successfully demonstrates proficiency in modern web development across the entire stack:

Robust Authentication: Email confirmation with Supabase Auth and JWT-based session management

Full Pet Management: Browse, filter, favourite, and apply to adopt pets from multiple shelters

Shelter Communication: Real-time chat messaging between users and shelters

Donation Platform: General and shelter-specific donations with simulated payment flow

Application Tracking: Status-based tracker with color-coded visual indicators

Modern UI/UX: Vibrant coral/purple theme with animations and fully responsive layout

Secure Backend: Flask REST API with Supabase PostgreSQL and RLS data protection

The system demonstrates real-world applicability as a scalable adoption platform, with a clear roadmap for expansion into mobile apps, real payment gateways, and AI-powered pet recommendations in future phases.