Pet Adoption Application

Overview

The **Pet Adoption Application** is a web-based platform designed to help users browse, search, and apply for adopting pets from shelters. Built using **React** for the frontend, **Node.js (Express)** for the backend, and **MongoDB** for the database, the application provides a smooth user experience for viewing pet profiles, filtering by category, and submitting adoption requests.

Shelter administrators can add new pets, manage existing profiles, view adoption requests, and update adoption statuses.
The application is deployed using **Vercel** for the frontend, **Render** for the backend, and **MongoDB Atlas** for secure cloud-hosted data storage

Components of the Project

Frontend

Technology: React.js
Functionality:

- Displays available pets using cards.
- Search and filter pets by category (Dog, Cat, Bird, etc.).
- Pet details page showing age, breed, gender, description, health info, and photos.
- Adoption Request form for users to submit application information.
- Responsive design for all devices.

Backend

Technology: Node.js with Express.js
Functionality:

- RESTful APIs for pets and adoption request management.
- CRUD operations for shelter admins (add, update, delete pets).
- User submission handling for adoption requests.
- Authentication and role-based authorization (optional Admin Login).
- Middleware for validating form submissions.

Database

Technology: MongoDB
Collections:

1. Pets

Fields:

- name
- type (Dog/Cat/Bird/etc.)
- breed
- age
- gender
- description
- healthStatus
- imageUrl
- createdAt

2. AdoptionRequests

Fields:

- petId
- applicantName
- email
- phone
- address
- reasonForAdoption
- status (Pending/Approved/Rejected)
- submittedAt

3. AdminUsers

Fields:

- name
- email
- password
- role
- createdAt

Hosting Platforms

Frontend Hosting:

Hosted on **AWS S3 + CloudFront**, providing fast, secure, globally distributed static website delivery.

Backend Hosting:

Deployed on **AWS EC2** (or AWS Elastic Beanstalk) for scalable Node.js backend hosting with automated deployment and monitoring.

Database Hosting:

Managed using **AWS DynamoDB** or **AWS RDS (MongoDB-compatible via DocumentDB)** for secure, highly available cloud-based data storage.

Frontend Components

(matching the structure of your Assignment Tracker document)

1. Components Folder

- **Navbar.jsx** – Navigation menu (Home, Pets, About, Contact).
- **PetCard.jsx** – Displays a single pet's image, name, and type.
- **PetList.jsx** – Lists all pets using PetCard.
- **FilterBar.jsx** – Buttons or dropdown to filter by category.

2. Context Folder

- **PetContext.jsx** – Manages global state for pets and adoption requests.

3. Pages Folder

- **HomePage.jsx** – Landing page with banner + featured pets.
- **PetDetailsPage.jsx** – Shows detailed information about a selected pet.
- **AdoptionFormPage.jsx** – Users fill details to request adoption.
- **AdminDashboardPage.jsx** – Admin panel to manage pets and requests.
- **AddPetPage.jsx** – Admin page for adding new pets.
- **ManageRequestsPage.jsx** – Admin can see all adoption submissions.

Backend Components

Controllers

- **PetController** – Handles fetching, adding, editing, and deleting pets.
- **RequestController** – Manages adoption request submissions.
- **AuthController** – For admin login (optional).

Routes

- `/pets` – Get/Add/Update/Delete pets.
- `/adoption` – Handle adoption requests.
- `/auth` – Admin authentication.

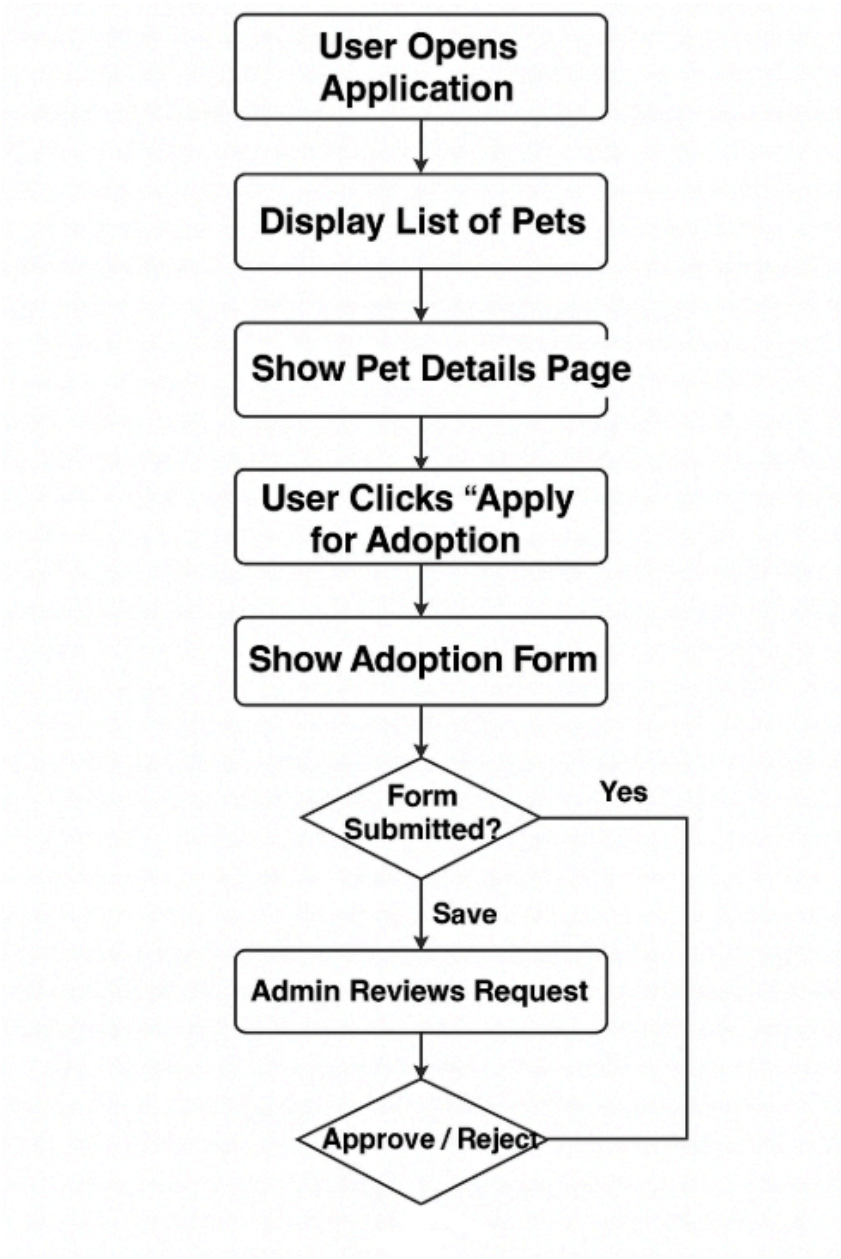
Models

- **Pet Model** – Schema for pet details.
- **AdoptionRequest Model** – Schema for adoption request forms.
- **AdminUser Model** – Schema for admin credentials.

Middleware

- Authentication (JWT).
- Role-based authorization.
- Form validation.

Flowchart


User Opens Application
Display List of Pets
Show Pet Details Page
User Clicks “Apply for Adoption
Show Adoption Form
Form Submitted?
Yes
Save
Admin Reviews Request
Approve / Reject