

Scenario: Virtual Pet Adoption Platform

Overview:

Welcome to Pawsome Pets Adoption Platform, where digital companions find their forever homes! As the backend developer, your mission is to create a system that allows users to browse, adopt, and care for virtual pets. Administrators will manage the inventory of available pets and ensure their well-being.



Schema and Validations:

Field	Validation
petId	Unique identifier (automatically generated)
name	Required, String
species	Required, String (e.g., cat, dog, rabbit)
age	Required, Number (positive integer)
gender	Required, String (male, female, other)

adoptionStatus	String (default: available)
vaccinated	Boolean (default: false)
description	String
image	String (URL format)

Routes:

1. Basic CRUD Routes:

- GET /pets: Get all available pets
- GET /pets/:petId: Get details of a specific pet
- POST /pets: Add a new pet to the inventory
- PUT /pets/:petId: Update details of a pet
- DELETE /pets/:petId: Remove a pet from the inventory

2. Additional Routes for MongoDB Commands:

- GET /pets/species/:species: Get pets by species
- GET /pets/age/:minAge/:maxAge: Get pets within an age range
- GET /pets/available: Get all available pets for adoption
- GET /pets/vaccinated: Get all vaccinated pets
- PUT /pets/:petId/adopt: Mark a pet as adopted
- PUT /pets/:petId/vaccinate: Update vaccination status of a pet
- PUT /pets/:petId/change-image: Update the image URL of a pet

Technical Implementation:

1. Node.js and Express.js:

- Use Node.js for server-side logic.
- Utilize Express.js to create RESTful APIs for managing pets.

2. Mongoose:

- Define a Mongoose schema for pets with the specified fields and validations.
- Implement CRUD operations using Mongoose methods to interact with the MongoDB database.

3. Routes Implementation:

- Define route handlers for each endpoint, implementing the necessary logic to handle requests and responses.
- Use MongoDB queries to perform search operations and update pet details.

4. Error Handling:

- Implement error handling middleware to catch and handle errors.
- Return appropriate error responses with meaningful messages.