

Web 602

# Project 2- Pet Adoption web

Chloe

---

# Project Overview

What the app does?

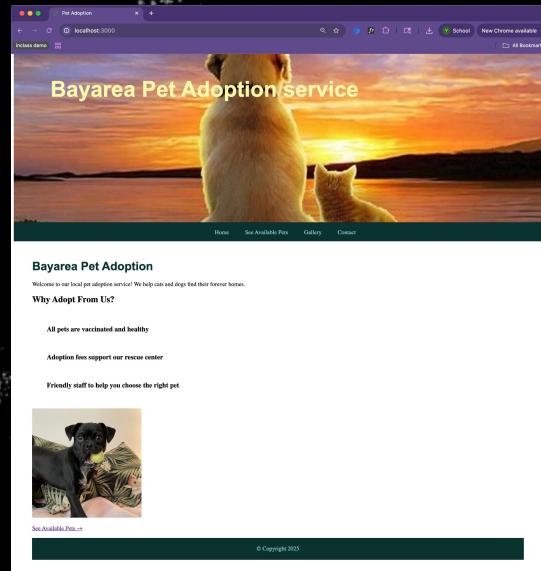
It's for people who want to adopt a pet and admin can add more data in the system

Who it's for

People who wanna a pet, and admin manage data

Main goal

Using new tech knowledge connect front-end and back-end, make the web app to a dynamic page.



# Core Features

- User authentication
- Pet listing & dashboard
- Database integration
- Session-based access control

The screenshot shows a web browser window titled "Bayarea Pet Adoption service" with the URL "localhost:3000/pets". The page has a header with links for Home, See Available Pets, Gallery, and Contact. Below the header is a banner image of a dog. The main content area is titled "Bayarea Pet Adoption - Available Pets" and contains a table titled "PETS INFORMATION". The table lists seven pets with their photos, names, ages, and breeds:

Photo	Name	Age	Breed
	Duo A	1	Pug
	Milo	2	Golden Retriever
	Bella	1	Poodle Mix
	Toni	1	German Shepherd
	coco	2	shih tzu
	coco2	3	Pug
	mini	1	ragdoll

At the bottom of the page, there is a link labeled "Gallery".

The screenshot shows a web browser window titled "Pet Adoption" with the URL "localhost:3000/gallery". The page has a header with links for Home, See Available Pets, Gallery, and Contact. The main content area is titled "Pet Gallery" and contains a heading "Meet some of our adorable pets waiting for a forever home:". Below the heading is a grid of nine pet photos, each with a caption:

- Happy Dog
- Playful Pup
- Adopt Me
- Best Friend
- Cute Smile
- Rescue Me
- Sweet Puppy
- Adorable Face
- Loyal Friend

At the bottom of the page, there is a link labeled "Forever Home".

# Overall Architecture

- Express server-side rendering
- MongoDB + Mongoose
- EJS templates



# Frontend & Backend Connection

```
app.get("/pets", async (req, res) => {
  const pets = await Pet.find();
  console.log("PETS FROM DB:", pets);
  res.render("pets", { pets });
});
```

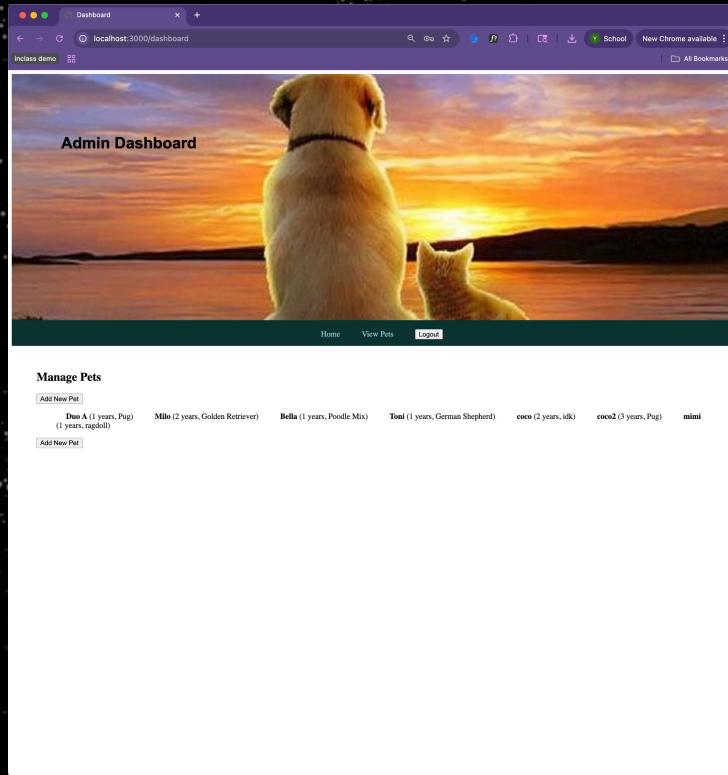
```
<% if (pets && pets.length > 0) { %>
<% pets.forEach(pet => { %>
<tr>
  <td>
    " width="150">
  </td>
  <td><%= pet.name %></td>
  <td><%= pet.age %></td>
  <td><%= pet.breed %></td>
</tr>
<% }) %>
<% } else { %>
<tr>
  <td colspan="4">No pets available yet.</td>
</tr>
<% } %>
```

Data is retrieved from MongoDB and passed to EJS templates using Express routes

# Authentication & Authorization

```
// dashboard
const requireLogin = (req, res, next) => {
  if (!req.session.userId) {
    return res.redirect("/login");
  }
  next();
};
```

A screenshot of a web browser window titled "Login". The URL bar shows "localhost:3000/login". The page content is a "Login" form with fields for "Email" containing "ve@gmail.com" and "Password" containing "....". Below the form is a "Login" button and a link "Don't have an account? [Sign up](#)".



# Database & Models

```
import mongoose from "mongoose";

const petSchema = new mongoose.Schema({
  name: { type: String, required: true },
  age: Number,
  breed: String,
  description: String,
  imageUrl: String,
  createdAt: { type: Date, default: Date.now }
});

export default mongoose.model("Pet", petSchema);
```

The screenshot shows the MongoDB Compass interface connected to a local host at port 27017. The 'pet\_adoption' database is selected, and the 'pets' collection is viewed. Three documents are listed:

- ObjectID('69444da2be8d0530e683b674'): name: "coco", age: 2, breed: "Lab", description: "Max", imageUrl: "https://www.outsidetheonline.com/wp-content/uploads/2023/03/Funny\_Dog\_H.j..."
- ObjectID('6944c24688a0f37be9bd7d37'): name: "coco", age: 2, breed: "Pug", description: "Happy dog", imageUrl: "data:image/jpeg;base64,/9j/4AAQSkZJRgABQAAAQkBAAD/2wCEAAKGBwEQEPUPEBA..."
- ObjectID('6945eb01cf1e9e2a1b6870d'): name: "Milo", age: 1, breed: "Magle", description: "L", imageUrl: "data:image/jpeg;base64,/9j/4AAQSkZJRgABQAAAQkBAAD/2wCEAAKGBwEQEPUPEBA..."

Below the table, a code editor displays a JavaScript file for inserting multiple documents into the 'pets' collection:

```
import Mongoose from "mongoose";
import Pet from "./models/Pet.js";

await mongoose.connect("mongodb://127.0.0.1:27017/pet_adoption");

await Pet.insertMany([
  {
    name: "Milo",
    age: 1,
    breed: "Golden Retriever",
    imageUrl: "/images/dog1.jpeg"
  },
  {
    name: "Bella",
    age: 1,
    breed: "Poodle Mix",
    imageUrl: "/images/dog2.jpeg"
  },
  {
    name: "Rocky",
    age: 3,
    breed: "German Shepherd",
    imageUrl: "/images/dog12.png"
  }
]);

console.log("Seed data inserted");
process.exit();
```

# Form Handling (POST request)

```
app.post("/pets", requireLogin, async (req, res) => {
  const { name, age, breed, description, imageUrl } = req.body;

  if (!name || !imageUrl) {
    return res.send("Name and image are required");
  }

  await Pet.create({
    name,
    age,
    breed,
    description,
    imageUrl
  });

  res.redirect("/dashboard");
});
```

# Unit Testing

```
import request from "supertest";
import { expect } from "chai";
import app from "../app.js";

describe("GET /pets", () => {
  it("should return status 200", async () => {
    const res = await request(app).get("/pets");
    expect(res.status).to.equal(200);
  });
});
```

The screenshot shows a code editor with two tabs open: `JS app.test.js` and `JS app.js`. The `app.test.js` tab contains a Mocha test for the `/pets` endpoint using the `supertest` module. The `app.js` tab shows the actual application logic.

The database, visible in the sidebar under `PROJECT2-M1`, contains the following data:

- `Buddy`:
  - `name: "Buddy", age: 1, breed: "Pug", description: "", imageUtl: "https://cdn.britannica.com/34/233234-050-1649BF09/Pug-dog.jpg?w=400&h=300&c=crop", createdAt: 2025-12-18T09:25:32.830Z, _v: 0`
  - `name: "Milo", age: 2, breed: "Golden Retriever", imageUtl: "/images/dog2.jpeg", createdAt: 2025-12-18T09:35:37.101Z, _v: 0`
  - `name: "Bella", age: 1, breed: "Poodle Mix", imageUtl: "/images/dog2.jpeg", createdAt: 2025-12-18T09:35:37.101Z, _v: 0`
  - `name: "Tom", age: 1, breed: "German Shepherd", description: "Friendly", imageUtl: "https://www.outsideonline.com/wp-content/uploads/2023/03/Funny_Dog_H.jpg", createdAt: 2025-12-18T09:41:10.342Z, _v: 0`
  - `name: "Coco", age: 2, breed: "Husky", description: "NA", imageUtl: "https://www.outsideonline.com/wp-content/uploads/2023/03/Funny_Dog_H.jpg", createdAt: 2025-12-18T09:41:22.089Z, _v: 0`
  - `name: "Cooper", age: 3, breed: "Pug", description: "Happy dog", imageUtl: "data:image/png;base64,...", createdAt: 2025-12-18T09:41:22.089Z, _v: 0`

# Technologies Used

- **JavaScript ES6**
- **Express**
- **MongoDB / Mongoose**
- **EJS**
- **bcrypt**
- **express-session**
- **Mocha / Chai**
- **GitHub**