

Web 602

Project 2- Pet Adoption web

Chloe

Project Overview

What the app does?

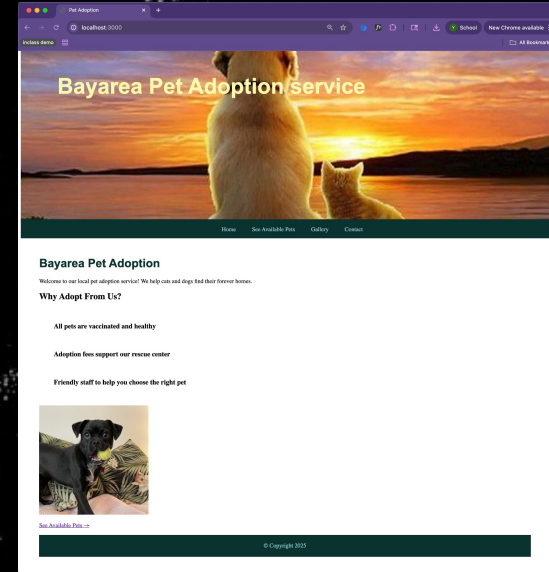
It's for people who want to adopt a pet and admin can add more data in the system

Who it's for

People who wanna a pet, and admin manage data

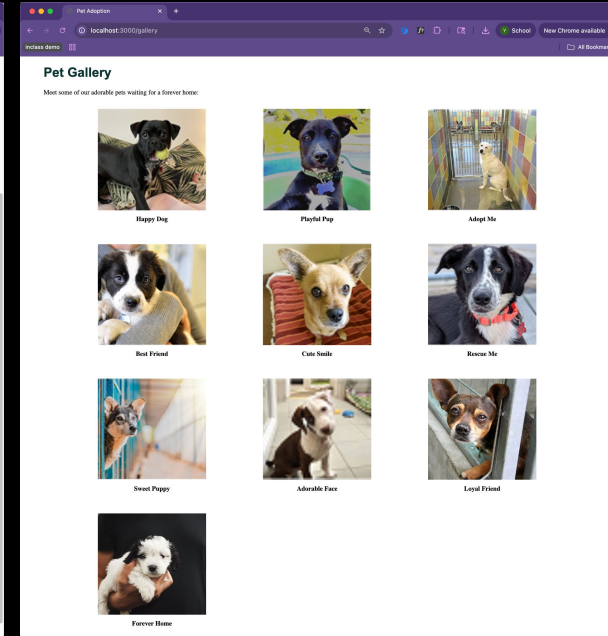
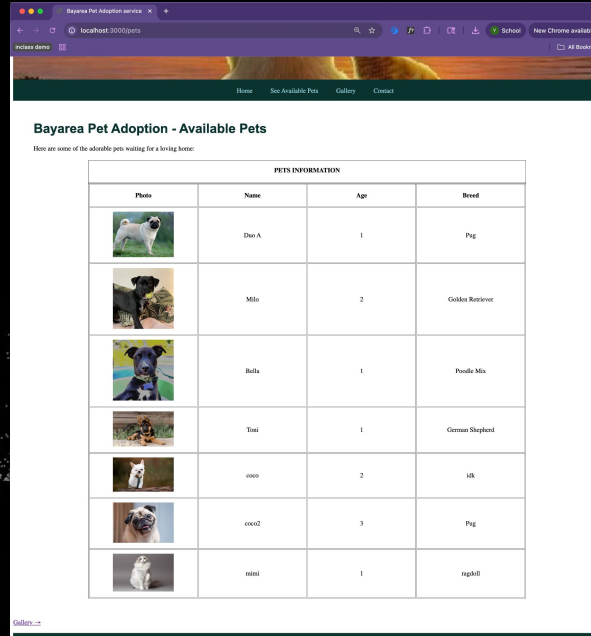
Main goal

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



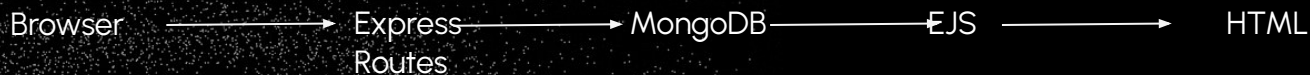
Core Features

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



Overall Architecture

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



Frontend & Backend Connection

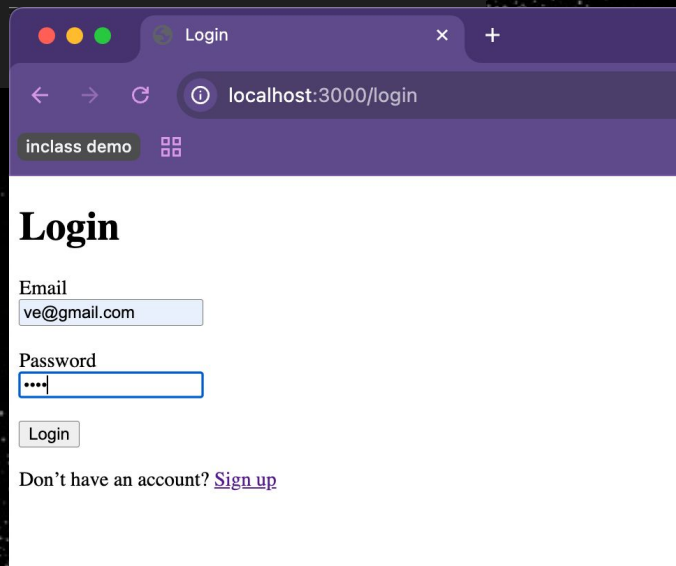
```
app.get("/pets", async (req, res) => {  
  const pets = await Pet.find();  
  console.log("PETS FROM DB:", pets);  
  res.render("pets", { pets });  
});
```

```
<% 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();
};
```



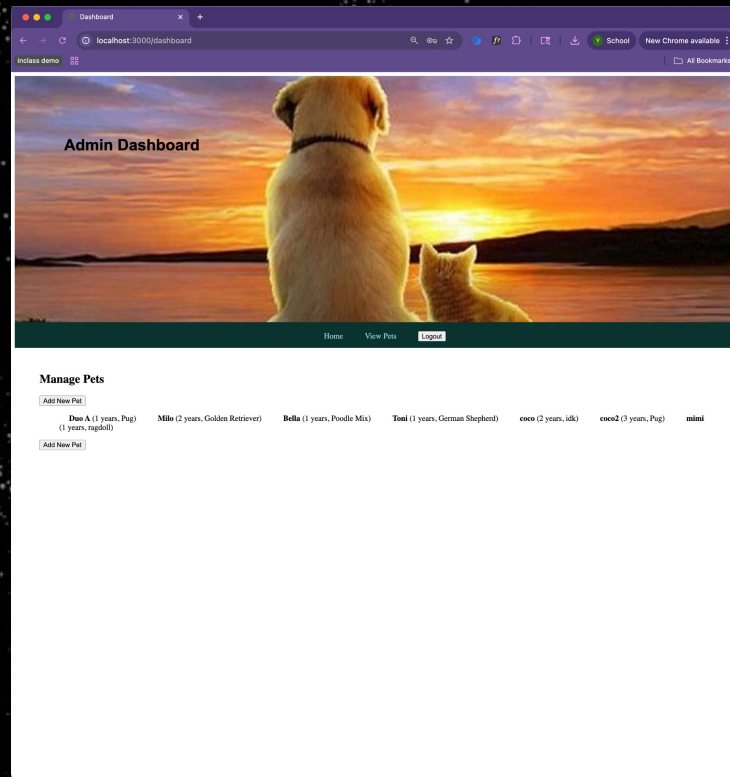
Login

Email
ve@gmail.com

Password

Login

Don't have an account? [Sign up](#)

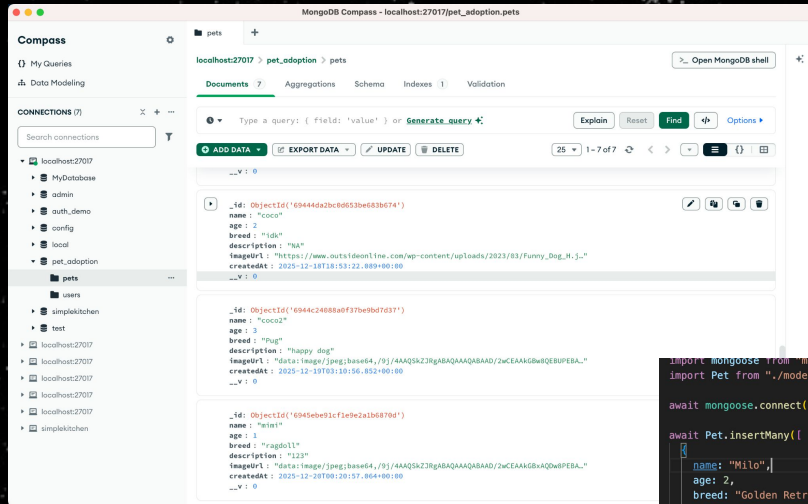


Database & Models

```
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);
```



```
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: 2,
    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");
  }

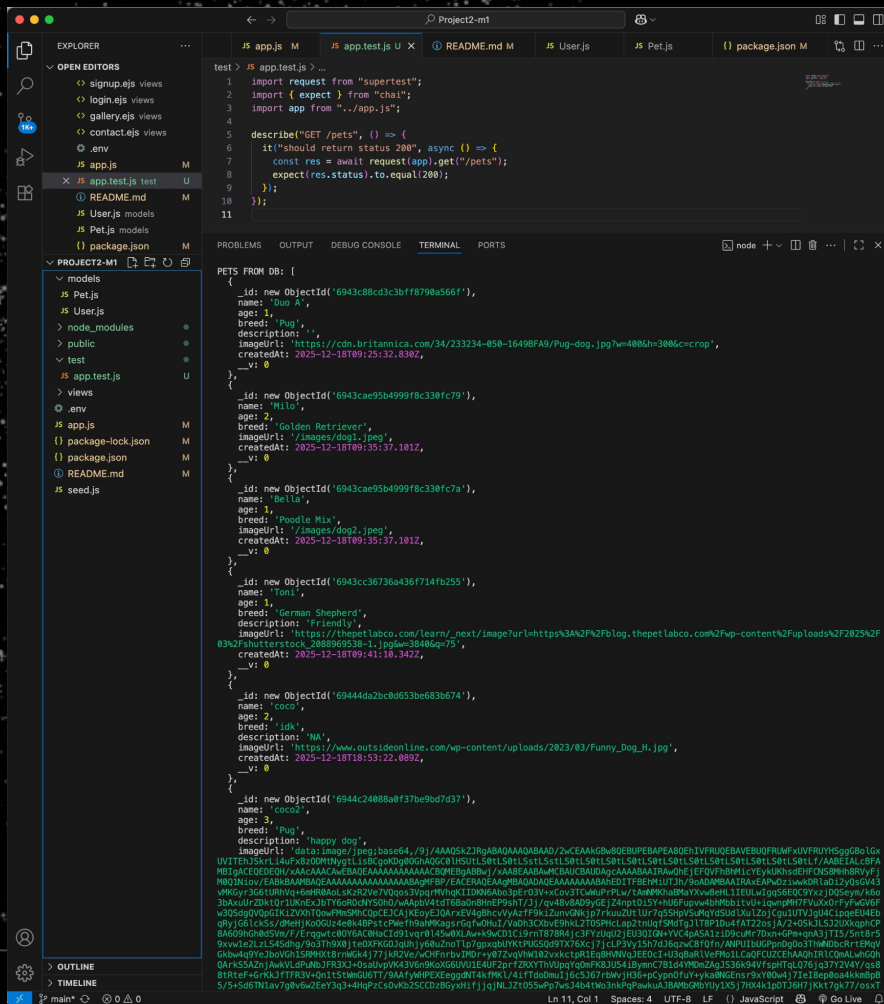
  await Pet.create({
    name,
    age,
    breed,
    description,
    imageUrl
  });

  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);
  });
});
```



Technologies Used

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