CS 499 Capstone Narrative – Software Design and Engineering

Student Name: Emireth Castro
Course: CS 499 Computer Science Capstone
Enhancement: Software Design and Engineering (Java Console App → Full-Stack MERN Application)

A. Describe the Artifact

The original artifact was a Java-based console application created in IT 145. It managed rescue animals—specifically dogs and monkeys—using simple input and output via the terminal. The program included core classes such as `RescueAnimal`, `Dog`, and `Monkey`, and a monolithic `Driver.java` file that handled all logic inside the `main()` method.

This early project focused on object-oriented fundamentals, including class inheritance and encapsulation. While it functioned for simple data entry and printing, it lacked key design and usability principles expected in production-grade software. There was no persistent storage, modular separation, or interface for modern use cases such as web integration or REST APIs.

B. Justify the Inclusion

I selected this artifact because it marks the foundation of my programming education. It demonstrates how I evolved from static, procedural Java coding to designing scalable full-stack applications using modern JavaScript frameworks. The enhancement completely transformed the system into a full-stack MERN application (MongoDB, Express, React, Node.js), built with modular, RESTful architecture.

The **backend enhancement** features:

- Modular route/controller/model separation (`server.js`, `routes/animals.js`, `controllers`, `models/Dog.js`, `models/Monkey.js`)
- Schema validation using Mongoose
- Authentication using JWT tokens
- Centralized error handling and validation middleware
- Binary search logic for improved efficiency
- Unit testing with Jest and Supertest

The **frontend enhancement** features:

- A dynamic, mobile-responsive React interface

- Admin dashboard with form-based intake and edit functions
- Visual filtering, pagination, and CSV/JSON export
- Cloudinary image uploads for animals
- Dark mode toggle for accessibility

This end-to-end rework showcases my ability to apply secure design, client-server communication, and responsive interface principles. It satisfies multiple course outcomes related to software engineering and secure full-stack development.

C. Reflect on the Enhancement Process

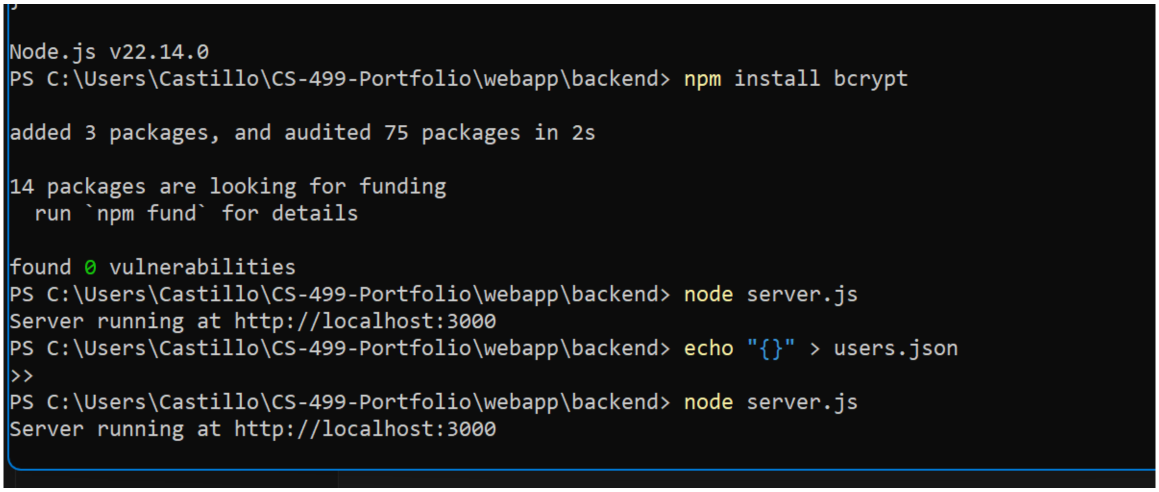
This was the most complex enhancement of the capstone. Refactoring the monolithic logic into modern backend routes and services pushed me to learn about REST patterns, request lifecycle, and asynchronous control flow. Writing middleware to validate animal data, sanitize inputs, and enforce unique IDs helped me understand real-world backend architecture.

The frontend taught me valuable lessons about state management, conditional rendering, and using reusable React components to improve maintainability. One major challenge was linking Cloudinary image uploads to backend schema objects while preserving form state across validations.

I incorporated feedback from testing tools, browser console warnings, and peer review. I also aligned the layout and button styling with professional UI standards. This work directly contributed to mastery in full-stack architecture, client-server coordination, and secure, modular code.

Screenshots

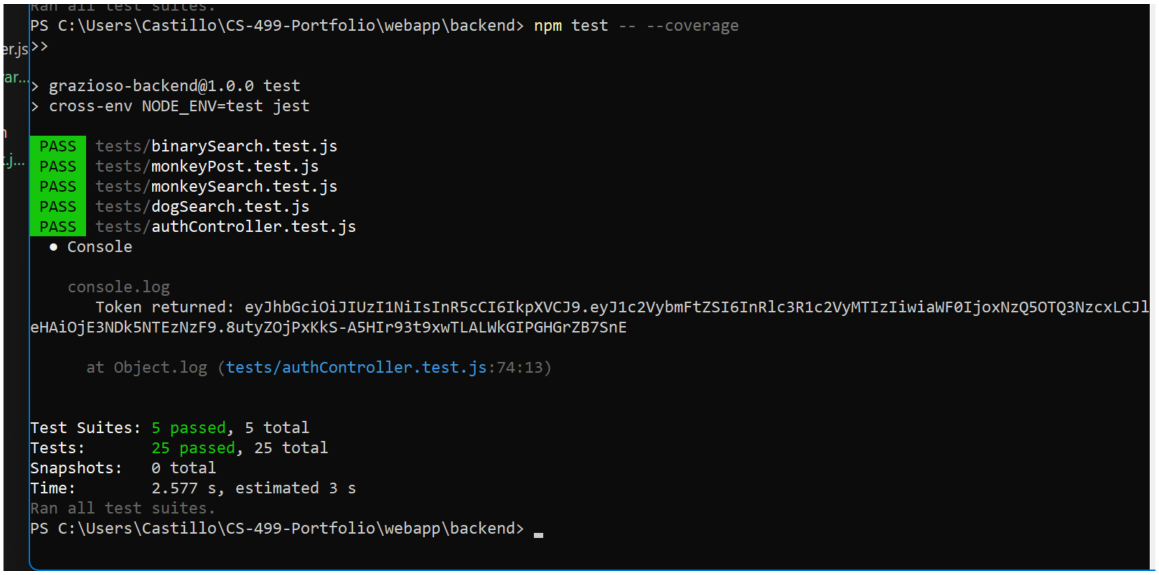
Terminal – Launching the Backend Server
This screenshot shows the use of `npm install` and `node server.js` to run the backend server on `http://localhost:3000`, confirming the Node.js API is active and listening.



```
Node.js v22.14.0
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> npm install bcrypt

added 3 packages, and audited 75 packages in 2s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> node server.js
Server running at http://localhost:3000
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> echo "{}" > users.json
>>
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> node server.js
Server running at http://localhost:3000
```

Jest Test Execution

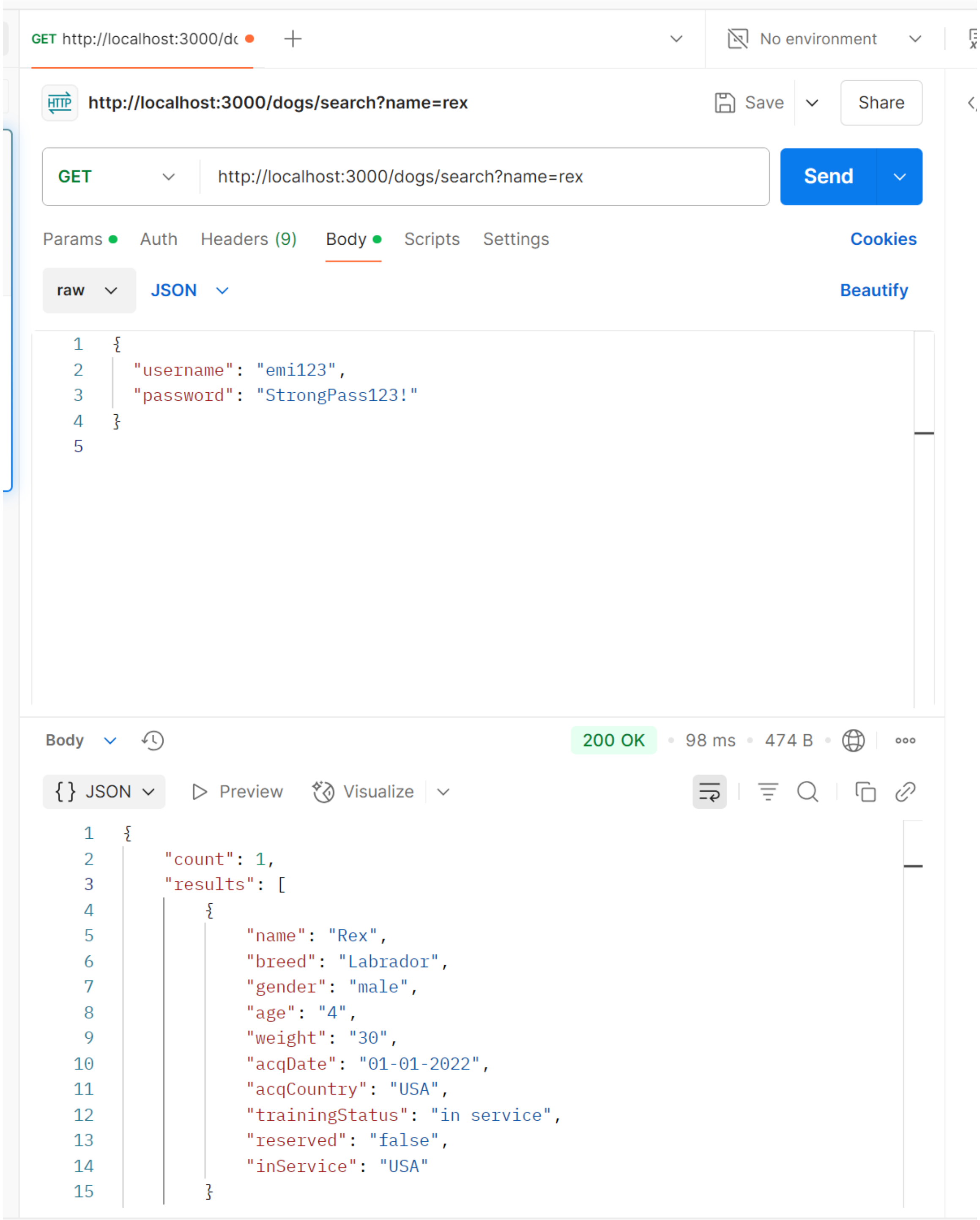
Running `npm test` triggers Jest, which tests multiple backend components including binary search logic, monkey/dog endpoints, and authentication. All tests pass, ensuring stable API behavior.



```
Ran all test suites.
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> npm test -- --coverage
>>

> grazioso-backend@1.0.0 test
> cross-env NODE_ENV=test jest

 PASS  tests/binarySearch.test.js
 PASS  tests/monkeyPost.test.js
 PASS  tests/monkeySearch.test.js
 PASS  tests/dogSearch.test.js
 PASS  tests/authController.test.js
  ● Console

    console.log
      Token returned: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InRlc3R1c2VyMTIzIiwiaWF0IjoxNzQ5OTQ3NzcxLCJl
eHAiOjE3NDk5NTEzNzF9.8utyZOjPxKkS-A5HIr93t9xwTLALWkGIPGHGrZB7SnE

      at Object.log (tests/authController.test.js:74:13)


Test Suites: 5 passed, 5 total
Tests:       25 passed, 25 total
Snapshots:   0 total
Time:        2.577 s, estimated 3 s
Ran all test suites.
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> 
```

Postman – GET and POST Requests

A successful `GET` and `POST` request is shown in Postman. The POST request demonstrates schema validation by accepting only correctly formatted data for a new animal object.



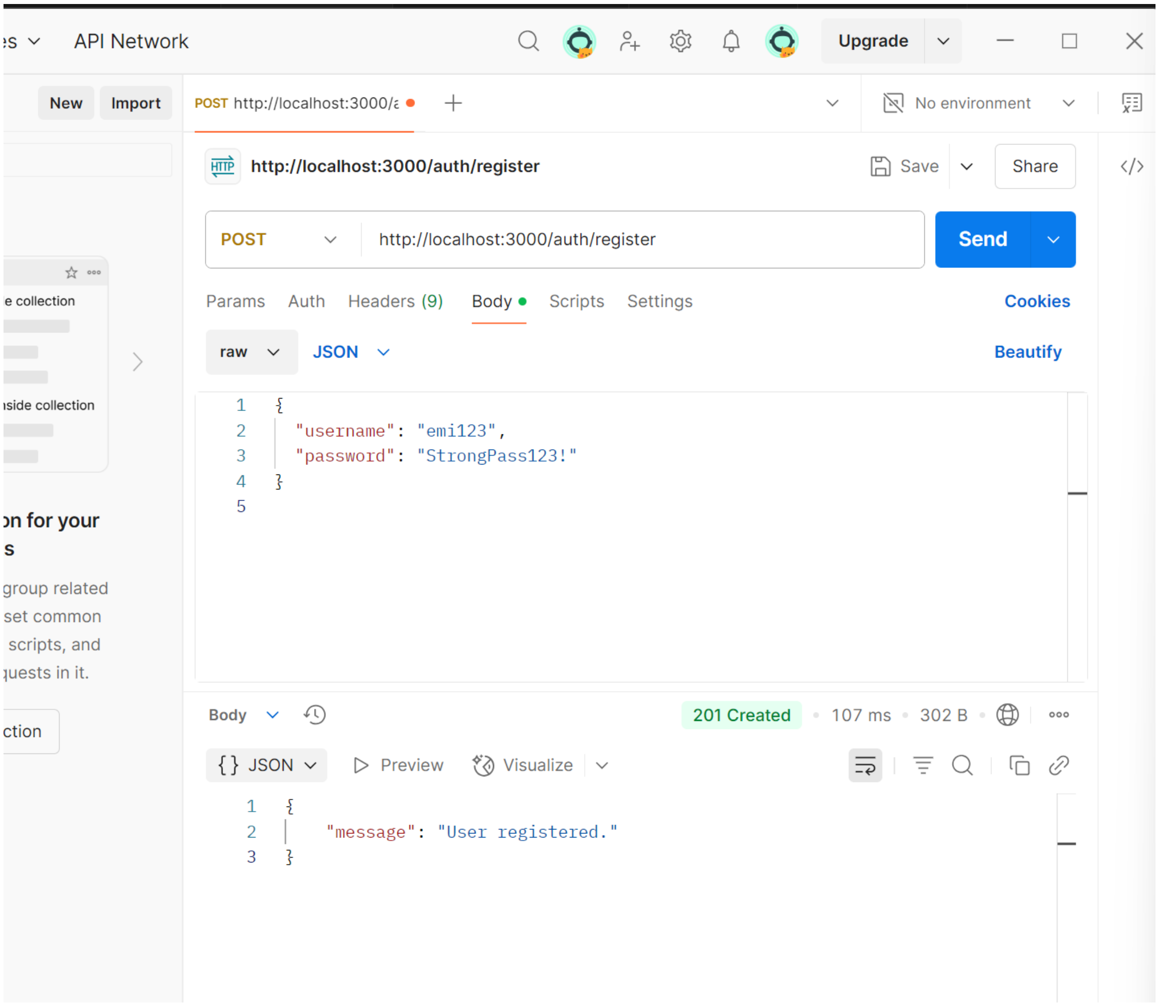
http://localhost:3000/dogs/search?name=rex

GET http://localhost:3000/dogs/search?name=rex

```
{
  "username": "emi123",
  "password": "StrongPass123!"
}

```

200 OK • 98 ms • 474 B

```
{
    "count": 1,
    "results": [
        {
            "name": "Rex",
            "breed": "Labrador",
            "gender": "male",
            "age": "4",
            "weight": "30",
            "acqDate": "01-01-2022",
            "acqCountry": "USA",
            "trainingStatus": "in service",
            "reserved": "false",
            "inService": "USA"
        }
```



http://localhost:3000/auth/register

POST http://localhost:3000/auth/register

raw JSON

```
{
  "username": "emi123",
  "password": "StrongPass123!"
}
```

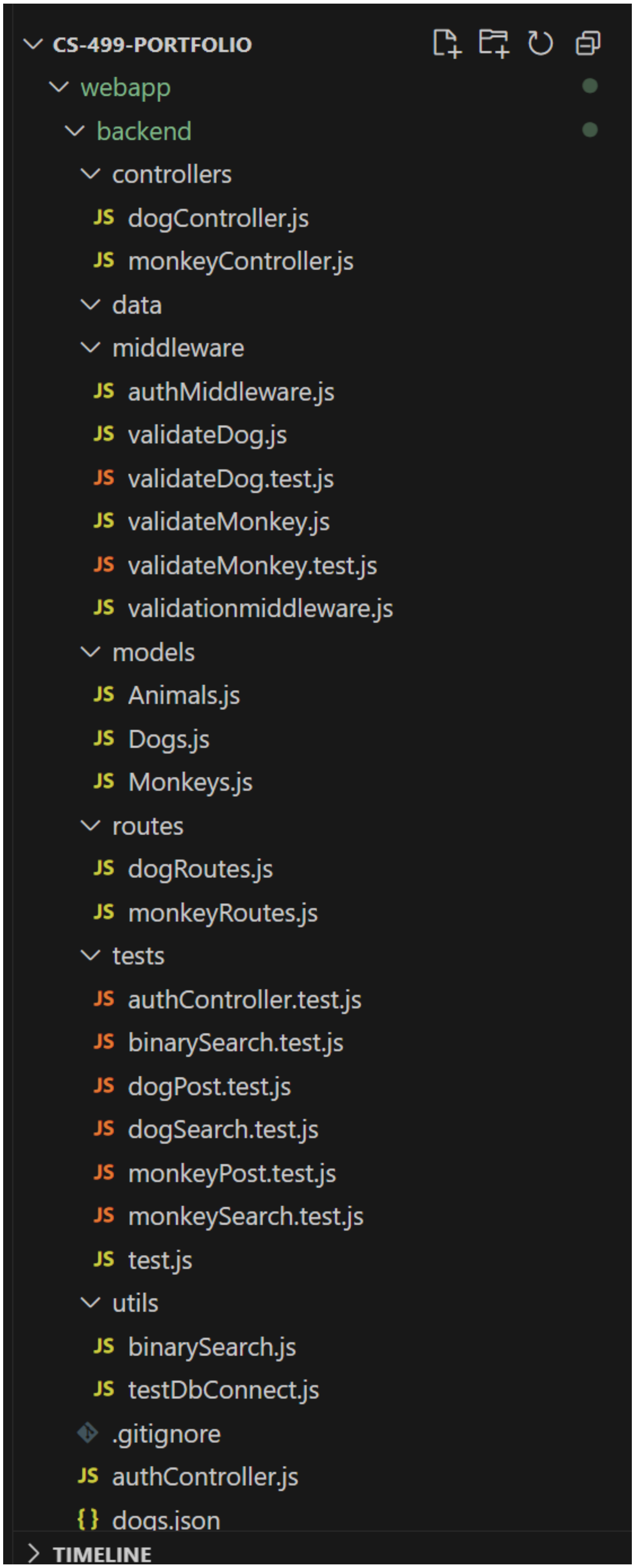
Body · 201 Created · 107 ms · 302 B

{} JSON

```
{
    "message": "User registered."
}
```

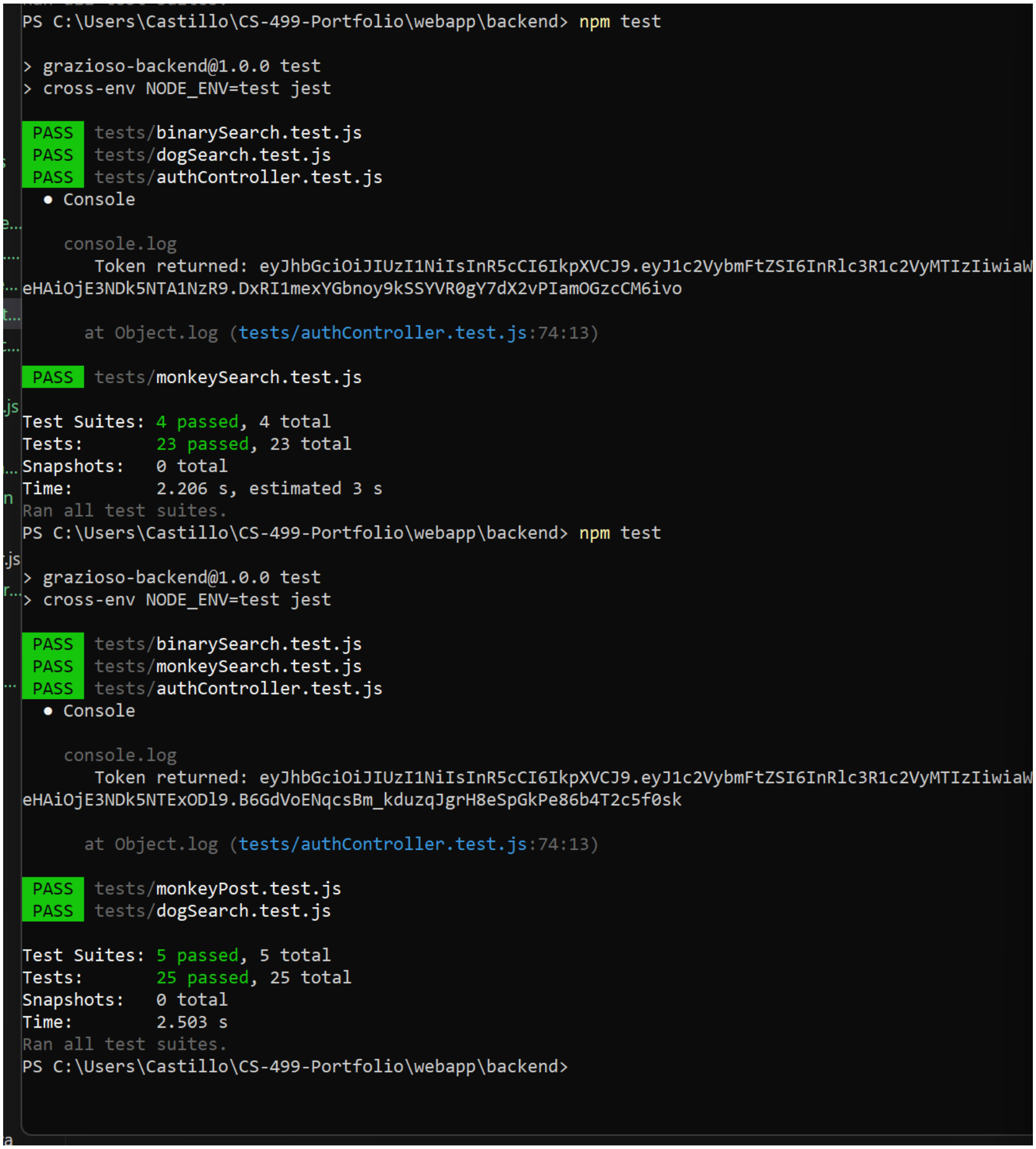
Project Folder Tree

The folder structure includes `controllers`, `routes`, `models`, and `middleware`, demonstrating modular separation of logic. This aligns with professional backend application patterns.

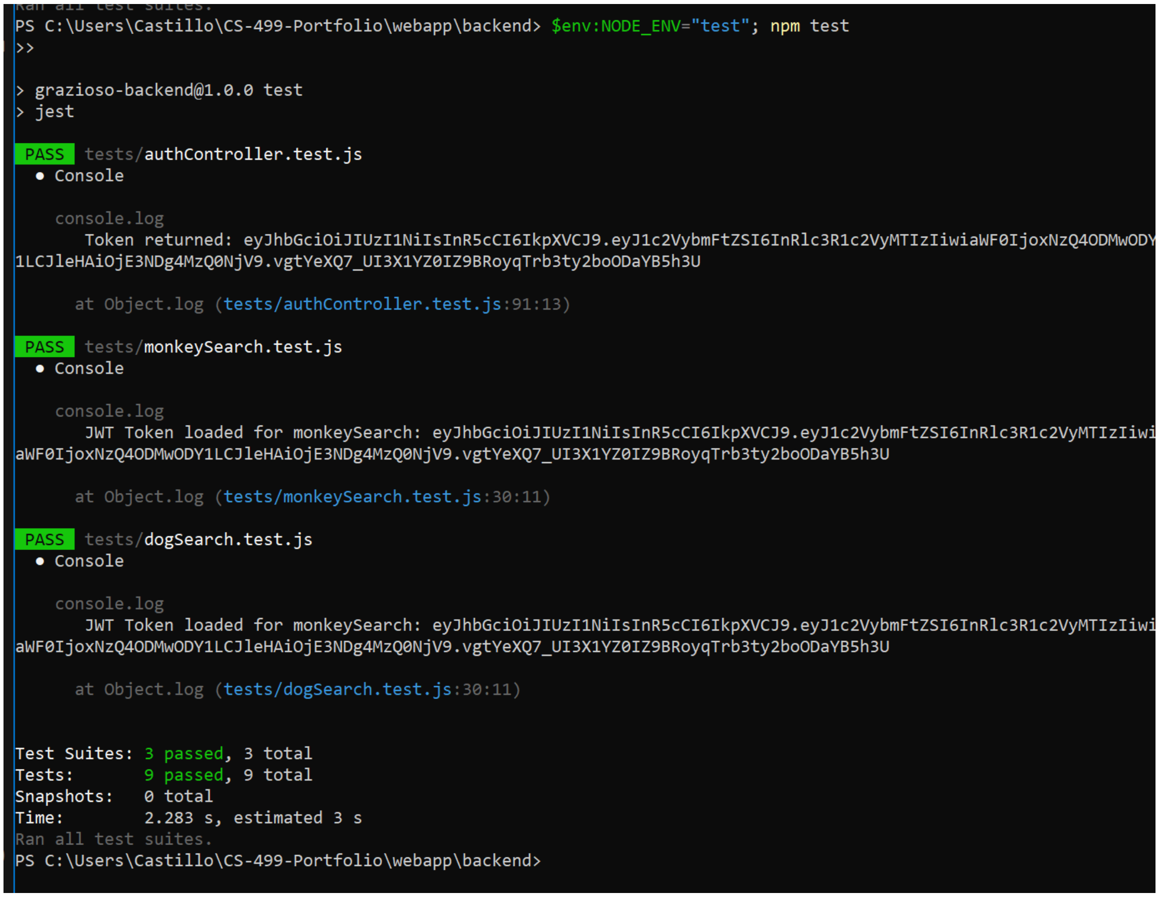


```
CS-499-PORTFOLIO
  webapp
    backend
      controllers
        dogController.js
        monkeyController.js
      data
      middleware
        authMiddleware.js
        validateDog.js
        validateDog.test.js
        validateMonkey.js
        validateMonkey.test.js
        validationmiddleware.js
      models
        Animals.js
        Dogs.js
        Monkeys.js
      routes
        dogRoutes.js
        monkeyRoutes.js
      tests
        authController.test.js
        binarySearch.test.js
        dogPost.test.js
        dogSearch.test.js
        monkeyPost.test.js
        monkeySearch.test.js
        test.js
      utils
        binarySearch.js
        testDbConnect.js
      .gitignore
      authController.js
      dogs.json
TIMELINE
```

- **Dog.js Schema with Validation**
  The model includes required fields, data types, and custom validation logic (e.g., checking species values and unique IDs) using Mongoose.



```
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> npm test

> grazioso-backend@1.0.0 test
> cross-env NODE_ENV=test jest

 PASS  tests/binarySearch.test.js
 PASS  tests/dogSearch.test.js
 PASS  tests/authController.test.js
  ● Console

    console.log
      Token returned: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InRlc3R1c2VyMTIzIiwiaW
eHAiOjE3NDk5NTA1NzR9.DxRI1mexYGbnoy9kSSYVR0gY7dX2vPIamOGzcCM6ivo

      at Object.log (tests/authController.test.js:74:13)

 PASS  tests/monkeySearch.test.js

Test Suites: 4 passed, 4 total
Tests:       23 passed, 23 total
Snapshots:   0 total
Time:        2.206 s, estimated 3 s
Ran all test suites.
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> npm test

> grazioso-backend@1.0.0 test
> cross-env NODE_ENV=test jest

 PASS  tests/binarySearch.test.js
 PASS  tests/monkeySearch.test.js
 PASS  tests/authController.test.js
  ● Console

    console.log
      Token returned: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InRlc3R1c2VyMTIzIiwiaW
eHAiOjE3NDk5NTExODl9.B6GdVoENqcsBm_kduzqJgrH8eSpGkPe86b4T2c5f0sk

      at Object.log (tests/authController.test.js:74:13)

 PASS  tests/monkeyPost.test.js
 PASS  tests/dogSearch.test.js

Test Suites: 5 passed, 5 total
Tests:       25 passed, 25 total
Snapshots:   0 total
Time:        2.503 s
Ran all test suites.
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend>
```



```
Ran all test suites.
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend> $env:NODE_ENV="test"; npm test
>>

> grazioso-backend@1.0.0 test
> jest

 PASS  tests/authController.test.js
  ● Console

    console.log
      Token returned: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InRlc3R1c2VyMTIzIiwiaWF0IjoxNzQ4ODMwODY
1LCJleHAiOjE3NDg4MzQ0NjV9.vgtYeXQ7_UI3X1YZ0IZ9BRoyqTrb3ty2boODaYB5h3U

      at Object.log (tests/authController.test.js:91:13)

 PASS  tests/monkeySearch.test.js
  ● Console

    console.log
      JWT Token loaded for monkeySearch: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InRlc3R1c2VyMTIzIiwi
aWF0IjoxNzQ4ODMwODY1LCJleHAiOjE3NDg4MzQ0NjV9.vgtYeXQ7_UI3X1YZ0IZ9BRoyqTrb3ty2boODaYB5h3U

      at Object.log (tests/monkeySearch.test.js:30:11)

 PASS  tests/dogSearch.test.js
  ● Console

    console.log
      JWT Token loaded for monkeySearch: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InRlc3R1c2VyMTIzIiwi
aWF0IjoxNzQ4ODMwODY1LCJleHAiOjE3NDg4MzQ0NjV9.vgtYeXQ7_UI3X1YZ0IZ9BRoyqTrb3ty2boODaYB5h3U

      at Object.log (tests/dogSearch.test.js:30:11)


Test Suites: 3 passed, 3 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        2.283 s, estimated 3 s
Ran all test suites.
PS C:\Users\Castillo\CS-499-Portfolio\webapp\backend>
```

Links:
GitHub Branch: software-design-enhancement
Live Frontend (Optional if hosted): `https://delfin7emi.github.io/CS-499-Portfolio/`