API documentation and Unit test cases

for

Solita pre-assignment

V1.0

Author: Likhitha Pentakota

**CATS API**

**API References**

* **Version**: 1.0
* **Host**: localhost:3020
* **Protocols**: https
* **Accepts**: application/json
* **Responds With**: application/json
* **Backend URL:** <http://localhost:3020/cats>

Cats API allows to read, write and delete cats. Collection of API for cats are described as below

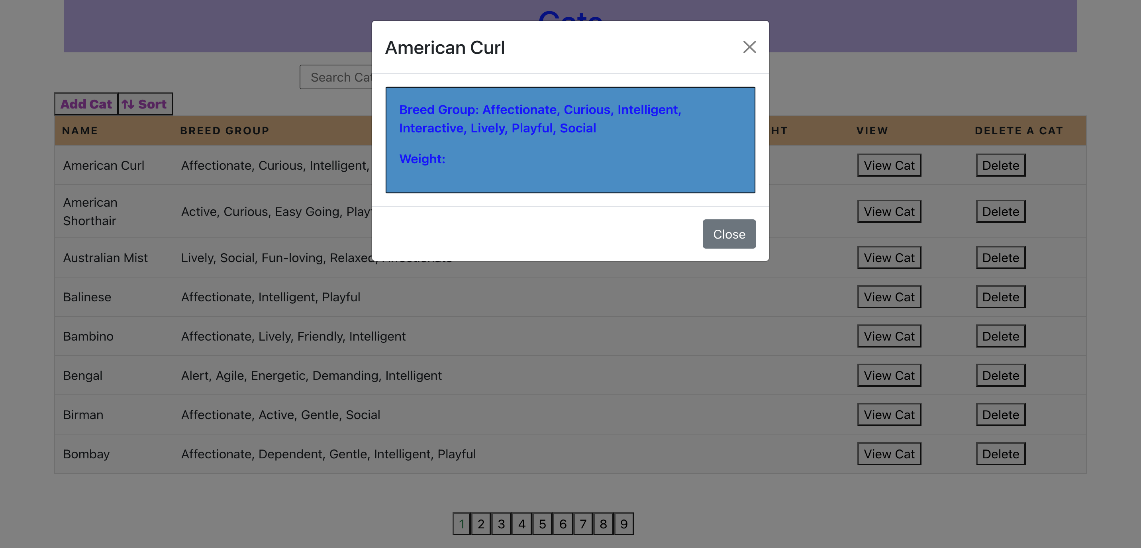
1. **Get all cats**
   1. Fetches all the cats in the list
   2. Endpoint to get all cats
      1. GET **/cats**
   3. Header
      1. Content-type: application/json
   4. Unit test
      1. API test with
         1. Postman

**UI for get all cats**

1. **Get cat by id**
   1. Fetches the cats from the list by cat id
   2. Endpoint to get all cats
      1. GET **/cats/{id}**
   3. Header
      1. Content-type: application/json
   4. Unit test
      1. API test with
         1. Postman

Incase id does not match:

**UI for get cat by id:**



1. **Get cat by name**
   1. Fetches all the cats in the list
   2. Endpoint to get all cats
      1. GET **/cats?name={name}**
   3. Header
      1. Content-type: application/json
   4. Unit test
      1. API test with
         1. Postman

Note: In case of name not found

1. **Delete cat**
   1. Fetches all the cats in the list
   2. Endpoint to get all cats
      1. DELETE **/cats/{id}**
   3. Header
      1. Content-type: application/json
   4. Unit test
      1. API test with
         1. Postman

Before deletion of the record:

200 OK status message for Delete

After deleted the record:

**UI for delete a cat by id:**

1. **Create a cat (POST)**
   1. Fetches all the cats in the list
   2. Endpoint to get all cats
      1. POST **/cats**
   3. Header
      1. Content-type: application/json
   4. Request body
      1. Schema

Object

Id type – uuid4

name type – string

breed\_group type – string

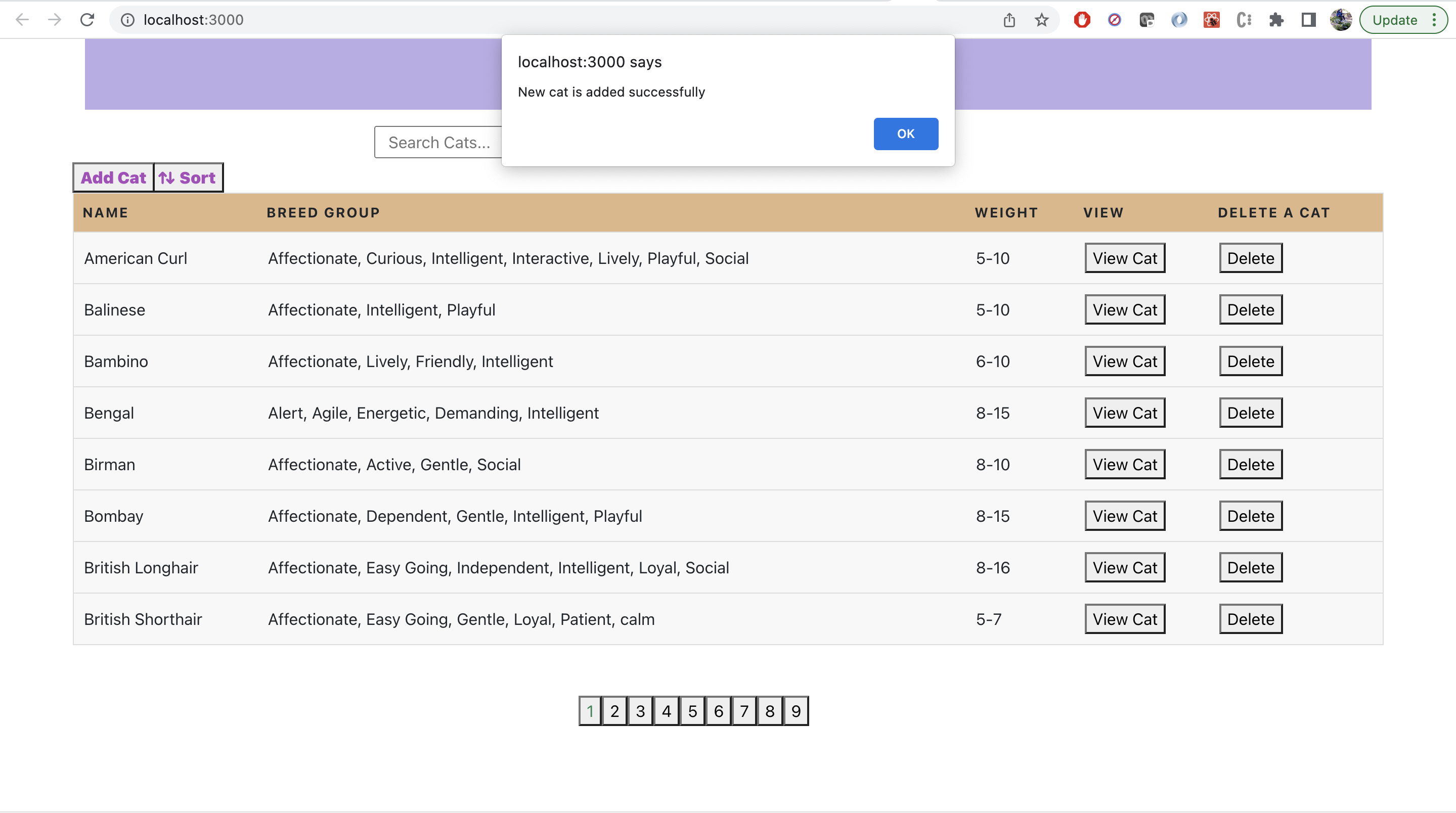
weight type - number

* 1. Unit test
     1. API test with
        1. Postman

Post a new record:

Get request to show the created record by name:

**UI for create a new cat:**

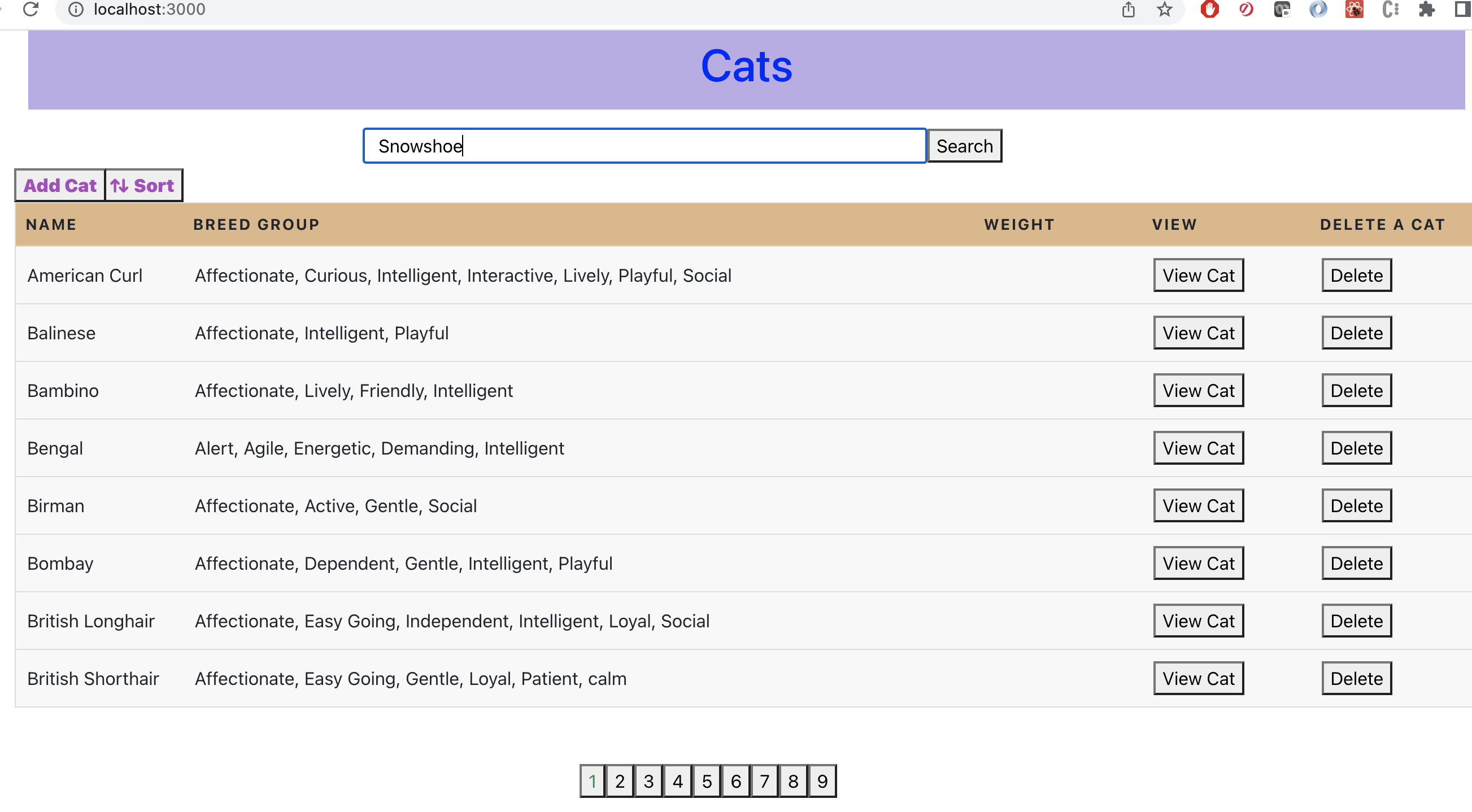


1. Search, Sorting and Pagination

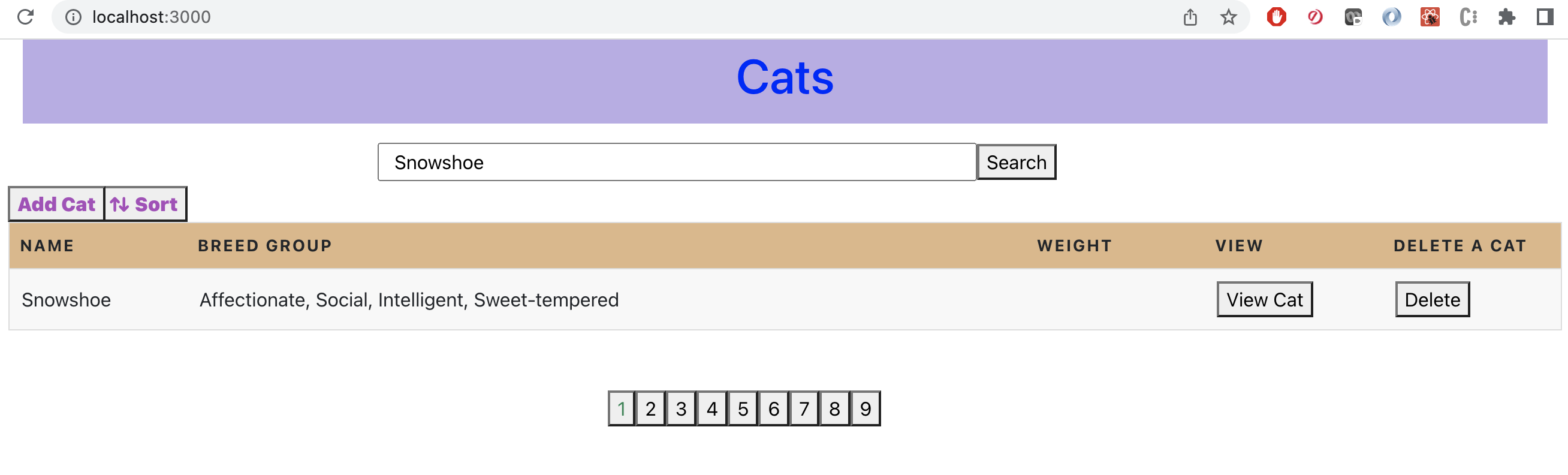
Search, Sorting, and pagination are done in UI and results are displayed as shown

1. Search – It is implemented based on results fetched from GET all cats

Before search



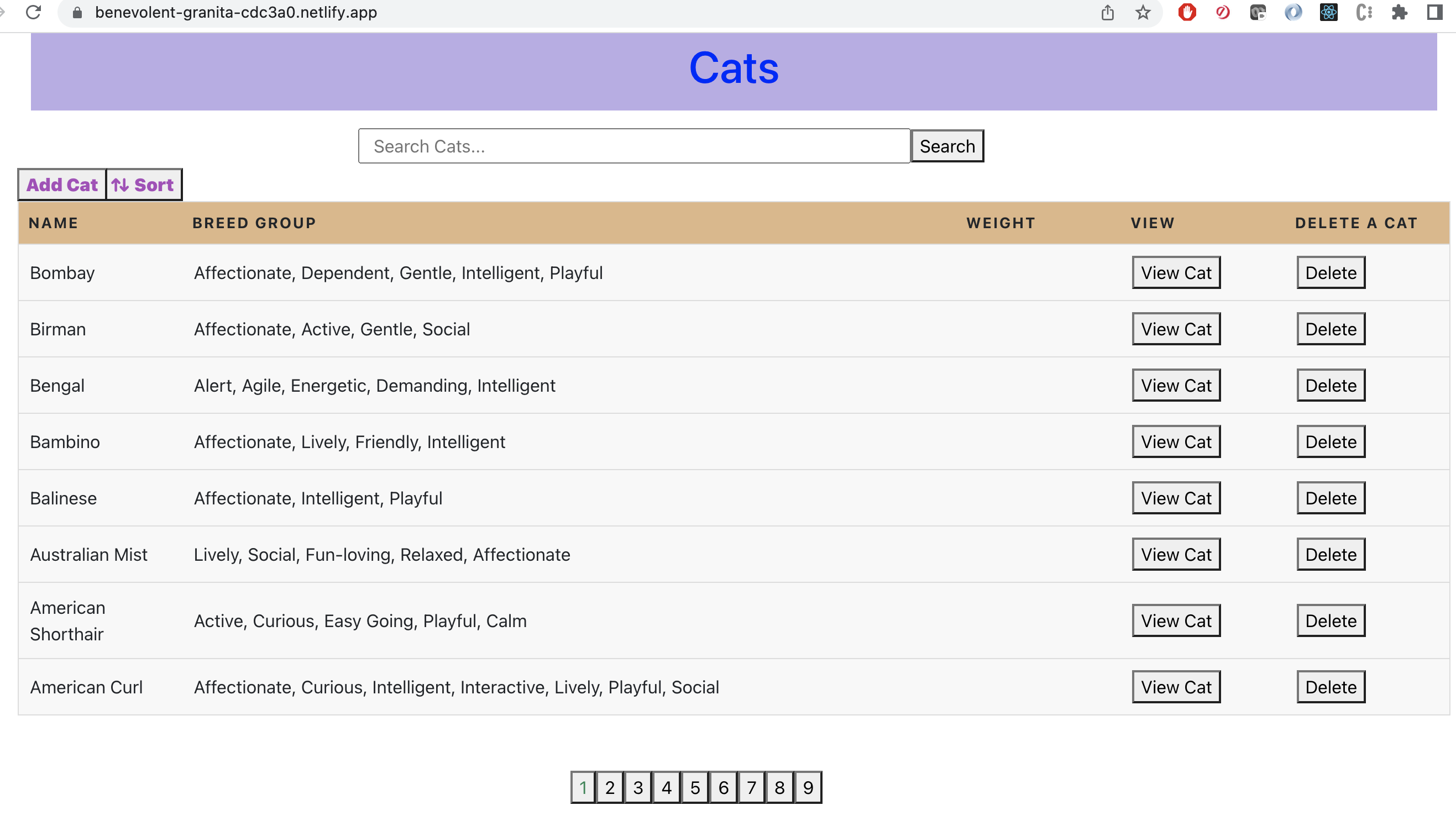
After search



1. Sort – Sort is implemented based on results displayed per each page



After sort is applied to page 1



1. Pagination – slice each data fetched by API with all cats to display data into multiple pages

**Conclusion**: Backend task is implemented based on my understanding on the requirement. It was really a good experience and put all of my knowledge and experience to implement with possible best practices.