

# LAB 4 MVC & MONGODB

Due: Week #14 (Friday April 21st, 2023) @ 11:59pm **HARD DEADLINE**

Value: 7.5%

MVC & MONGO DB

Maximum Mark: 100

**Overview:** This lab is made to reinforce the MVC architecture (software pattern) and the use of Mongo DB with Mongoose.

**Evaluation:**

35 Marks: Project Setup

5 Marks: Internal Documentation

70 Marks: Working with Views and Data

## Instructions:

A starter file has been provided for you. Rename it Lab4\_YourName. It only has *a few* of the components you require. You may choose to start from scratch as long as you meet all of the requirements of the lab.

### 1. Project Setup (35 Marks Site Structure)

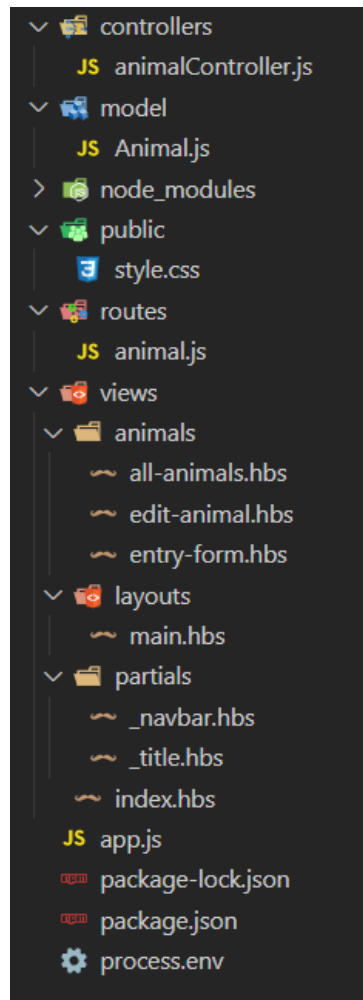
- Create an Express.js MVC project using npm init. Ensure the dependencies pictured below are in your package.js file. Update the name field with your name: (5 marks)

```
"dependencies": {  
  "bcryptjs": "^2.4.3",  
  "bootstrap": "^5.2.3",  
  "dotenv": "^16.0.3",  
  "express": "^4.18.2",  
  "express-handlebars": "^7.0.4",  
  "mongoose": "^7.0.3",  
  "nodemon": "^2.0.22",  
  "passport": "^0.6.0",  
  "path": "^0.12.7"  
}
```

- Make an app.js file for your entry file. Ensure you have a start script like the following: (5 marks)

```
"start": "nodemon app.js"
```

- c. Create a process.env file to hold important secret data. Create a PORT variable that you use to spin up your Express server. (5 marks)
- d. Create a connection to your MongoDB database that can be accessed from any IP address. Ensure you have the password and connection string in your process.env file. (5 marks)
- e. Ensure your starter code is updated to reflect the following MVC architecture: (5 marks)



- f. Create a Mongoose schema in your model/Animal.js file. It should have the following properties:
  - Zoo: String, required
  - Scientific name: String, required
  - Common name: String, required
  - Given name: String, required
  - Gender: String required
  - Date of birth: Date, required
  - Age: Number, required
  - isTransportable: Boolean, required

Export your schema so your controller can use it to query/update the database. (5 marks)

- g. Populate your database with 1000 entries of data (consider using Mockaroo or another mock data generator) with the correct data types. (5 marks)

## 2. Working With Views and Data (65 Marks)

- a. Update the main.hbs to use the \_title.hbs partial for the page title and the \_nav.hbs for the top navigation bar. You will need to make sure each page receives page title data and that you update the nav bar for the Home, Animal List, and Entry Form. Also, make it render the main body templates as we have done in class. Be sure to add styling to your project if you do not use the distributed starter code. (Note: If you change the style links in the header, you might break the style for the page. However, if you know what you're doing and want to improve it, go ahead!) (10 marks)
- b. Update entry-form.hbs to have a form similar to that pictured below. It can look different and use different controls from what is pictured but must capture all the same data points. It must offer a button that adds the animal record to the database and a reset button that works. (10 marks)

Animal Database

HomeAnimalFormList

### Add an Animal to the Database

Add Animal

Home Zoo

Scientific Name

Common Name

Given Name

Gender

☐ Male ☐ Female

Date of Birth

yyyy-mm-dd

Age

Is Transportable

☐ Available ☐ Unavailable

Add Animal to Database

Reset Form

- c. Update `all-animals.hbs` to display all the animals in your database. You can make your data presentation better than what is below, but all data points should be present. You must have an edit button that links to the `edit-animal.hbs` page. (10 marks)

<b>Zoo</b>	Kuala Lumpur Zoo
<b>Scientific Name</b>	Ratufa indica
<b>Common Name</b>	Squirrel, malabar
<b>Given Name</b>	Jobina2
<b>Gender</b>	Female
<b>Date of Birth</b>	Wed Jun 27 2001 20:00:00 GMT-0400 (Eastern Daylight Saving Time)
<b>Age</b>	21
<b>Transportability?</b>	AVAILABLE
<a href="#">Edit Animal Details</a>	

<b>Zoo</b>	Sekararum Zoo
<b>Scientific Name</b>	Speotyte cuniculata
<b>Common Name</b>	Owl, burrowing
<b>Given Name</b>	Cirilo
<b>Gender</b>	Male
<b>Date of Birth</b>	Sun Jul 22 2007 23:37:47 GMT-0400 (Eastern Daylight Saving Time)
<b>Age</b>	15
<b>Transportability?</b>	UNAVAILABLE
<a href="#">Edit Animal Details</a>	

<b>Zoo</b>	Macapá Zoo
<b>Scientific Name</b>	Taxidea taxus

- d. Update the `edit-animal.hbs` page to allow a user to update an animal's details or delete the animal. Your page should have at a minimum all the form controls as pictured below, but you can make it look better. The Update Animal Record button should update the record in the database. The delete button should delete the record from your database. (10 marks)

Animal Database

HomeAnimalForm  
List

## Edit Animal Entry

### Edit Animal

Zoo  
Kuala Lumpur

Scientific Name  
Ratufa indica

Common Name  
Squirrel, malabar

Given Name  
Jobina2

Gender ☐ Male ☒ Female

Date of Birth  
2001-06-28

Age  
21

Is Transportable ☒ Available ☐ Unavailalbe

Submit

Delete Record

- e. Your routes/animal.js file will have a minimum of 4 get routes and 2 post routes (editing and deleting records) that will point to handlers in the controllers/animalController.js file. It will export a router that will be used by the app in app.js. (10 marks)
- f. All data handling (querying and updating the database) must be handled in appropriate methods in animalController.js. (20 marks)
3. Include Internal Documentation for your program (5 Marks):
- Ensure you include a comment header at the top of your app.js file that indicates: Your Full Name, StudentID and Date Completed (2 Marks).
  - Ensure you include function headers for all of your functions (1 Mark).
  - Ensure your program uses contextual variable names that help make your code human-readable (1 Marks).
  - Ensure you include inline comments as required. As a rule, the code should be descriptive, but sometimes some information is required, especially before any function you include (1 Marks)

## Evaluation Criteria

Feature	Description	Marks
Project Setup	Your Project adheres to the site structure described (including Assets, Content and Scripts folders)	35
Working with Views and Data	The program's deliverables are all met and the program functions as it should. No errors appear as a result of execution. User Input does not crash the program.	70
Internal Documentation	A comment header is present and includes the name of the student, StudentID, and date completed.	5
Total		100

## SUBMITTING YOUR WORK

Your submission should include:

1. A zip archive of your project uploaded to DC Connect
2. A working link to your complete project files on GitHub

This assignment is weighted **15%** of your total mark for this course.

Late submissions:

- NOT PERMITTED AT ALL (DUE LAST DAY OF CLASSES)

EXTERNAL CODE (E.G. FROM THE INTERNET OR OTHER SOURCES) CAN BE USED FOR STUDENT SUBMISSIONS WITHIN THE FOLLOWING PARAMETERS:

1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email/verbal with follow-up email