CSE 341 Final project Proposal

# General Info

Holly Briggs

NatureNotes

Contents:

Contents

[General Info](#_gjdgxs) **1**

[Application Info](#_oxlmg6u5hyfx) **2**

[What will the API do?](#_1fob9te) 2

[How will your API utilize a login system?](#_tyjcwt) 2

[What database will you use?](#_2et92p0) 2

[How will the data be stored in your database?](#_2s8eyo1) 2

[How would a frontend be able to manage authentication state based on the data you provide?](#_3dy6vkm) 2

[What pieces of data in your app will need to be secured? How will you demonstrate web security principles in the development of this app?](#_1t3h5sf) 2

[What file structure and program architecture will you use for this project (how will you organize your node project)? Why?](#_4d34og8) 2

[What are potential stretch challenges that you could implement to go above and beyond?](#_17dp8vu) 2

[**API Endpoint Planning**](#_mral43ww1pvt) **2**

[Project Scheduling and Delegation](#_1ksv4uv) **3**

[How will you divide up work in your team to ensure the following tasks all get completed?](#_3rdcrjn) 3

[Potential Risks and Risk Mitigation Techniques](#_26in1rg) **4**

[What are the risks involved with you being able to finish this project in a timely manner?](#_lnxbz9) 4

[How will you mitigate or overcome these risks?](#_35nkun2) 4

# Application Info

## What will the API do?

It’s a hobby log for someone who loves to spend time in Nature. It keeps track of wildlife sightings, hiking trails, cycling trails, favorite gear, and users who are campers and hikers.

## How will your API utilize a login system?

Google cloud Oauth 2.0 based authentication and authorization

## What database will you use?

I’ll create my database in MongoDB and use MongoDB Compass

## How will the data be stored in your database?

In four collections

Trails, wildlife sightings, gear, and users (campers/hikers)

## How would a frontend be able to manage authentication state based on the data you provide?

Google cloud would provide a login so that a user could use their relationship and account in the google ecosystem to provide the authentication

## What pieces of data in your app will need to be secured? How will you demonstrate web security principles in the development of this app?

The ability to post an entry like review of gear or a trail, or a wildlife sighting would be reserved for a logged in user. An administrative account would be required to add and delete any users entries. A non logged in user would have read access only. The backend will verify JWTs or OAuth tokens for protected routes.

## What file structure and program architecture will you use for this project (how will you organize your node project)? Why?

I would use MVC and REST architectures. I would have folders for controllers, models, data, helpers, middleware, routes and node\_modules. The controllers files would contain the functions that perform the POST PUT DELETE GET tasks and contain the error handling in try catch statements. The data folder would contain design jsons to help generate the documents contained in the collections in mongoDB. The middleware folder would contain all the validation logic/functions that get called in the routes. The routes folder would contain route files for all the separate collections. The routes files would include all the commenting needed to effectively generate the swagger.json file using the swagger.js file logic. The routes files would also contain all the router paths for the GET, POST, PUT and DELETE tasks. I would include a .gitignore file that would protect the .env file from being published to github as a security precaution. My node and npm installs information would be generated and stored in the package-lock.json and package.json files. I would also include REST client http files for testing the GET, PUT, DELETE, and POST paths. And I would have a server.js and swagger.js and a generated swagger.json all in the root directory.

## What are potential stretch challenges that you could implement to go above and beyond?

extra filtering/search features, analytics, or favorite trail recommendations.

# API Endpoint Planning

For this section, you’ll plan out what API endpoints you’ll need for your project. If you go to [editor.swagger.io](https://editor.swagger.io/) you’ll see the Pet Store application documentation that they have. This can be a good point of reference because they demonstrate how to have multiple database entities (ie: pet, store, user), and CRUD operations for each with various ways of performing them. For this section of the Final Project Proposal, you will make a list of each api endpoint that will be supplied for each database entity. So, if I was going to create the pet store app, I’d put something like this:

* pet
  + POST /pet
  + PUT /pet
  + GET /pet/findByStatus
  + GET /pet/findByTags
  + GET /pet/{petId}
  + POST /pet/{petId}
  + DELETE /pet/{petId}
  + POST /pet/{petId}/uploadImage
* store
  + GET /store/inventory
  + POST /store/order
  + GET /store/order/{orderId}
  + DELETE /store/order/{orderId}
* user
  + POST /user
  + POST /user/createWithArray
  + POST /user/createWithList
  + GET /user/login
  + GET /user/logout
  + GET /user/{username}
  + PUT /user/{username}
  + DELETE /user/{username}

Thinking about this now will be extremely helpful for you because next week when you have to create the swagger documentation for all of this and publish it to heroku so it is ready for the rest of your project.

Users

* POST /users # Create a new user (sign up)
* GET /users # Get a list of all users
* GET /users/{userId} # Get a single user by ID
* PUT /users/{userId} # Update a user’s info
* DELETE /users/{userId} # Delete a user
* POST /users/login # Log in a user (OAuth/bcrypt)
* POST /users/logout # Log out the current user
* GET /users/favorites/{userId} # Get a user's favorite trails

Trails

* POST /trails # Add a new trail
* GET /trails # Get a list of all trails
* GET /trails/{trailId} # Get a single trail by ID
* PUT /trails/{trailId} # Update a trail’s information
* DELETE /trails/{trailId} # Delete a trail
* GET /trails/search?type={type}&difficulty={difficulty} # Filter trails by type/difficulty

Gear

* POST /gear # Add a new piece of gear
* GET /gear # Get a list of all gear
* GET /gear/{gearId} # Get a single gear item by ID
* PUT /gear/{gearId} # Update gear information
* DELETE /gear/{gearId} # Delete a gear item
* GET /gear/user/{userId} # Get all gear owned by a specific user

Wildlife Sightings

* POST /sightings # Log a new wildlife sighting
* GET /sightings # Get all wildlife sightings
* GET /sightings/{sightingId} # Get a single sighting by ID
* PUT /sightings/{sightingId} # Update a sighting
* DELETE /sightings/{sightingId} # Delete a sighting
* GET /sightings/trail/{trailId} # Get all sightings for a specific trail
* GET /sightings/species/{species} # Get all sightings of a specific species

# Project Scheduling and Delegation

Plan out what tasks will get completed with each lesson remaining in the semester (Only edit highlighted text).

|  |  |
| --- | --- |
| Week 04 Tasks | *Project Proposal* |
| Week 05 Tasks | * *Create Git Repo* * *Push to Heroku* * *API DOCUMENTATION is complete and available at route ‘/api-docs’*   Create all the collections in mongoDB and write all the controller files/functions with error handling, |
| Week 06 Tasks | Create the validation and all the routes and REST client http test files |
| Week 07 Tasks | Leave time for testing, lots of testing.  *Film the Video Presentation…* |

## How will you divide up work in your team to ensure the following tasks all get completed?

* HTTP GET, GET (all, single) Holly
* HTTP POST Holly
* HTTP PUT Holly
* HTTP DELETE Holly
* Node.js project creation Holly
* Create git repo and share with group Holly
* MongoDB setup Holly
* API Swagger documentation for all API routes Holly
* Video presentation of node project, all routes functioning, mongoDB data being modified, and API documentation. Holly

# Potential Risks and Risk Mitigation Techniques

## What are the risks involved with you being able to finish this project in a timely manner?

I’ve been behind this whole term so far and I’m likely to do this alone since no one else in my team shows up for the team meetings. I think I can do it, but it won’t be easy.

## How will you mitigate or overcome these risks?

Careful determination. I will commit to working on this project every day, at least three hours until completed.