CSE 341 Final Project Proposal

# General Info

Spread Factor: High

Team Members:

Shared Israel Ordaz Santillan

James Lule

Alain Solohery

Application Name: OPEN Archive

Contents:

Contents

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

[Application Info](#_30j0zll) **2**

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

[How will your API utilize a login system?](#_3znysh7) 2

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

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

[**API Endpoint Planning**](#_17dp8vu) **2**

[Project Scheduling and Delegation](#_3rdcrjn) **3**

[How will you divide up work in your team to ensure the following tasks all get completed?](#_26in1rg) 3

[Potential Risks and Risk Mitigation Techniques](#_lnxbz9) **4**

[What are the risks involved with you being able to finish this project in a timely manner?](#_35nkun2) 4

[How will you mitigate or overcome these risks?](#_1ksv4uv) 4

# Application Info

## What will the API do?

…Is a website that stores and delivers the bibliography of books and talks, and quotes.

## How will your API utilize a login system?

…We will be using passport.js for Oauth integration

## What database will you use?

…MongoDB

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

…Using a controller, and MongoDB crud operations. Also we will use Forms

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

…Same, using controllers to manage passport.js

## 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?

…Users database, and the UPDATE and DELETE operations. The contents will be free to use. Passport.js should be enough to manage the security. Giving special permissions only to admin users.

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

…REST, because honestly is the one I know better.

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

…A GUI (Graphical User Interface) on the view. Using ajax to display the files and make the API user-friendly.

# 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:

* root
  + GET / : Displays Home page \*
  + GET /api-docs: Swagger documentation \*
* users
  + GET /users : Displays HTML to manage operations of a user \*
  + GET /users/register: Deliver an HTML register form\*
  + POST /users/register: Register the user from the HTML form\*
  + GET /users/update: Deliver an HTML update form and a DELETE button to delete the user
  + POST /users/update: Update the user from the HTML form
  + POST /users/delete: Delete an user using a form. In the users/update route
  + PUT /users/update: Update a user account with a content-type application/json header (ONLY ADMIN)\*
  + DELETE /users/{userId}: Delete an user (ONLY ADMIN)\*
  + GET /users/login: Deliver an HTML login form
  + GET /users/json: Display all the users in a json (ONLY FOR ADMIN ACCOUNTS)
* books
  + GET /books: Get books list on a json
  + GET /books/{bookId}: Get an specific book info on a json
  + DELETE /books/{bookId}: Delete a book (ONLY ADMIN)
* talks
  + GET /talks: Get talks list on a json
  + GET /talks/{talkId}: Get an specific talk info on a json
  + DELETE /talks/{talkId}: Delete a talk (ONLY ADMIN)
* quotes
  + GET /quotes: Get books list on a json
  + GET /{quoteId}: Get an specific quote info on a json
  + DELETE /quote/{quoteId}: Delete a quote (ONLY ADMIN)
* uploads
  + GET /uploads: Deliver an HTML to upload a book, talk or quote. Select the database on the form and upload it
  + POST /uploads: Upload the talk to

# Project Scheduling and Delegation

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

|  |  |
| --- | --- |
| Lesson 9 Tasks | *Project Proposal* |
| Lesson 10 Tasks | * *Create Git Repo* * *Push to Render* * *API DOCUMENTATION is complete and available at route ‘/api-docs’* * Code the base express code * Make the routers * Make the HTML views (static) |
| Lesson 11 Tasks | …Do the controllers for users/register and users/update |
| Lesson 12 Tasks | …Do the controllers for the rest of user requests and the other routes (books, talks, quotes) |
| Lesson 13 Tasks | …Do the uploads controllers, typescript implementation (optional)  *…Video Presentation…* |

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

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

# Potential Risks and Risk Mitigation Techniques

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

…That my computer fails. Compatibility problems (I'm using Linux, most probably the others use MacOs or Windows), bad commits to the master branch, not being able to meet.

## How will you mitigate or overcome these risks?

…Using only common npm packages, also using only common formats (.js, .ts, .html, .env). Everyone has to have his own git branch and commit to master after the meeting. If it is not possible to meet, read the pull requests and announce the pushes to master branch in teams.